Fully Bayesian Unsupervised Disease Progression Modeling

Arya Pourzanjani[*]
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA
arya@umail.ucsb.edu

David Stück
Evidation Health
Santa Barbara, CA
dstuck@evidation.com

David Sontag
Department of Computer Science
New York University
New York, NY
dsontag@cs.nyu.edu

Luca Foschini
Evidation Health
Santa Barbara, CA
luca@evidation.com

Abstract

We present a practical implementation of a fully unsupervised disease progression model [10]. The implementation utilizes all new components we developed for generic use in Bayesian disease progression modeling. It improves upon [10] by providing a more informative fully Bayesian approach and a faster inference algorithm. The implementation is completely built on the pyMC3 open-source library making it easy to extend the model and apply to new settings.

1 Disease Progression Models

Traditionally, disease severity and progression have been assessed manually by physicians using guidelines such as the GOLD criteria for COPD [6]. These guidelines are typically based on rules applied to the patient's biomarkers, demographics, and other data easily extracted from health records. The sub-area of machine learning called disease progression modeling (DPM) focuses on automating this process [5]. Automation leads to more accurate diagnoses and optimal treatment paths which can literally be the difference between life and death as in the case of coagulopathy patients [9]. More broadly, we expect that algorithms that learn disease progression models from electronic health records will lead to new insights on the progression of rare and difficult to stage chronic diseases, guiding both clinical practice and medical research.

2 Bayesian Models and pyMC3

Bayesian networks provide a natural framework for modeling disease progression. They allow for the flexible modeling of "hidden states" which often arise in medical scenarios where measurements are simply proxies for variables of interest. Furthermore, Bayesian posteriors provide a full description of parameters of interest as oppose to point estimates and simple confidence intervals. Several examples of Bayesian network models for disease progression exist in the literature [1, 2, 4, 7, 10].

pyMC3 is a Python module that provides a unified and comprehensive framework for fitting Bayesian models using MCMC [8]. pyMC3's key strength is its modularity and extensibility: ran-

[*] Research performed while interning at Evidation Health

dom variables in a Bayesian network can be easily added or replaced to construct a model and multiple general purpose samplers are available.

3 New pyMC3 Tools for DPM

We implemented several tools in Python that could be used with `pyMC3` to aid with generic disease progression modeling:

- A Markov Jump Process for continuous modeling of state using discrete observations that come at irregular times
- A multi-dimensional binary Markov process for modeling the onset of comorbidities
- A noisy-or network for modeling health measurements as symptoms of comorbidities

We are releasing these tools as free and open-source software, which we hope will help accelerate progress on machine learning research of disease progression modeling.

4 Using Our Tools to Implement an Unsupervised DPM

We demonstrate our tools' capabilities by implementing the unsupervised DPM by Wang et al. described in [10]. The model utilizes all three newly implemented components to infer comorbidites and disease stage from electronic medical records. The top layer is a Markov Jump Process that reveals disease stage at discrete time points. The middle layer is a vector of comorbidities that are either on or off at each time step with some probability according to disease stage. The last layer consists of a noisy-or network where observations or measurements are triggered by comorbidities or a leak term. Figure 1 provides an overview.

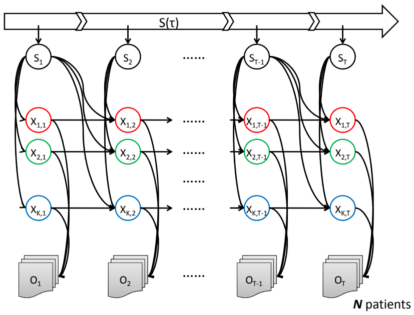

Figure 1: Overview of DPM from Wang et al. [10]

Our implementation of the model from [10] not only provides full posterior estimates (as opposed to point estimates from the EM algorithm), but takes advantage of `pyMC3`'s automatic differentiation to enable the NUTS sampler[3]. NUTS is a self-tuning Hamiltonian Monte Carlo sampler which takes advantage of the gradient of the likelihood to propose larger, more intelligent steps.

After implementing the model in `pyMC3`, we evaluated it using a synthetic dataset consisting of $N = 100$ patients with 2 to 34 time points each, $M = 4$ disease states, $K = 4$ comorbidities, and $D = 16$ types of observations, resulting in a total of 1609 clinical observations. We evaluated the robustness of the model to parameter initialization by comparing the model behavior when parameters are initialized at random vs. initialized to their ground truth value. We found that a random initialization rapidly finds the highest likelihood region achieving the same value of a near-equilibrium initialization. Additionally, the inferred distribution of leak terms for both initialization scenarios looks almost identical. The results are reported in Figure 2.

References

[1] E. Choi, N. Du, R. Chen, L. Song, and J. Sun. Constructing disease network and temporal progression model via context-sensitive hawkes process.

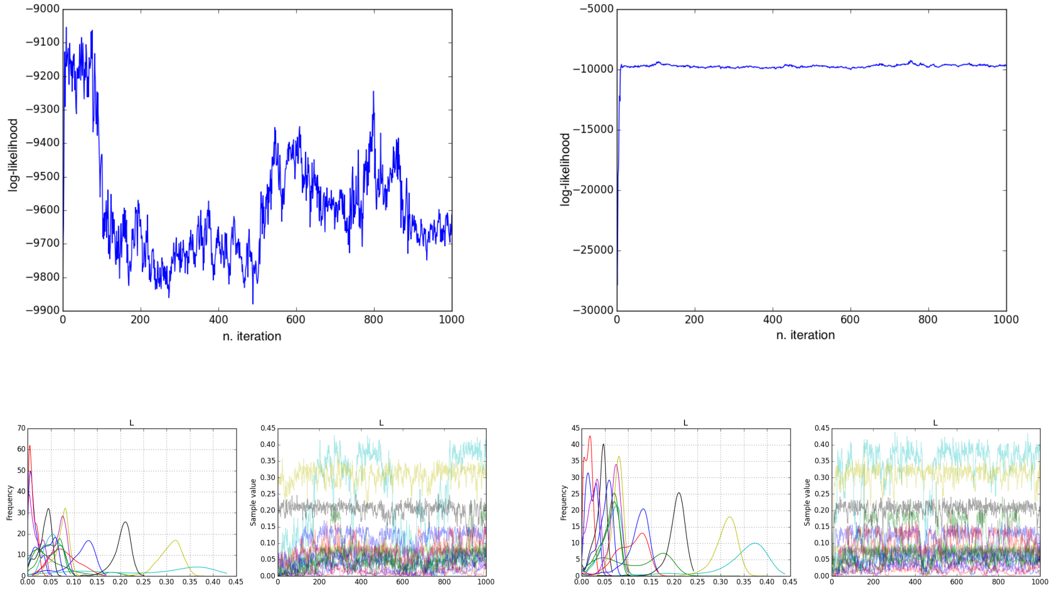

Figure 2: Top row: Total model log-likelihood when using NUTS. When parameters are initialized at near-equilibrium the log-likelihood hovers around a stable value (left). The same log-likelihood value is rapidly achieved when parameters are initialized at random (right). Bottom row: distributions of the leak terms in the noisy-or network for near-equilibrium (left) and random (right) initializations. The distributions look almost identical.

[2] K. P. Exarchos, T. P. Exarchos, C. V. Bourantas, M. Papafaklis, K. K. Naka, L. K. Michalis, O. Parodi, D. Fotiadis, et al. Prediction of coronary atherosclerosis progression using dynamic bayesian networks. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 3889–3892. IEEE, 2013.

[3] M. D. Homan and A. Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *The Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[4] C. H. Jackson, L. D. Sharples, S. G. Thompson, S. W. Duffy, and E. Couto. Multistate markov models for disease progression with classification error. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(2):193–209, 2003.

[5] D. Mould. Models for disease progression: new approaches and uses. *Clinical Pharmacology & Therapeutics*, 92(1):125–131, 2012.

[6] R. A. Pauwels, A. S. Buist, P. M. Calverley, C. R. Jenkins, and S. S. Hurd. Global strategy for the diagnosis, management, and prevention of chronic obstructive pulmonary disease. *American journal of respiratory and critical care medicine*, 2012.

[7] R. Pivovarov, A. J. Perotte, E. Grave, J. Angiolillo, C. H. Wiggins, and N. Elhadad. Learning probabilistic phenotypes from heterogeneous ehr data. *Journal of biomedical informatics*, 2015.

[8] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in python using pymc.

[9] D. R. Spahn, B. Bouillon, V. Cerny, T. J. Coats, J. Duranteau, E. Fernández-Mondéjar, D. Filipescu, B. J. Hunt, R. Komadina, G. Nardi, et al. Management of bleeding and coagulopathy following major trauma: an updated european guideline. *Crit Care*, 17(2):R76, 2013.

[10] X. Wang, D. Sontag, and F. Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.