bnpy : Reliable and scalable variational inference for Bayesian nonparametric models

Michael C. Hughes and Erik B. Sudderth
Department of Computer Science, Brown University, Providence, RI 02912
mhughes@cs.brown.edu, sudderth@cs.brown.edu

Abstract

We introduce `bnpy`, a new inference engine implemented in Python for unsupervised learning from millions of examples. Our framework applies to a large class of parametric and Bayesian nonparametric (BNP) clustering models that capture sequential, hierarchical, or spatial structure. For BNP models, we develop memoized variational algorithms that explore adding or removing clusters to discover compact, interpretable models.
Python code: `http://bitbucket.org/michaelchughes/bnpy-dev/`

1 Goals

A primary goal in machine learning is to infer interpretable clusters or segmentations from complex datasets. While the simple mixture model is quite popular, many models go beyond universal exchangeability to capture spatial, temporal, hierarchical, or relational structure. Further work has combined these structured models with Bayesian nonparametric (BNP) priors [1] like the Dirichlet process (DP) to enable *learning* the number of clusters from data, rather than fixing this number in advance as parametric models do.

Unfortunately, most implementations are not effective for real-world applications. Both MCMC sampling-based methods [2] and optimization-based methods [3] show sensitivity to poor initializations and can remain trapped in local optima even after days of computation. Practitioners often keep the best of dozens of independent restarts, though this approach is too expensive for large-scale datasets and often still does not yield satisfactory results.

Our research goal is to provide a scalable and reliable inference framework for a large (but not universal) family of clustering models, including both BNP and parametric models. In pursuit of this goal, we are building a Python toolbox called `bnpy` that can apply diverse models and algorithms to real-world datasets via compositional modules. To achieve scalability, we focus our efforts on two modern optimization-based approaches which can process data one subset (or "batch") at a time: *stochastic* variational inference [4] and *memoized* variational inference [5]. To achieve reliability, we design birth, merge, and delete moves that can add or remove clusters for BNP models during a single run of inference, enabling discovery of a parsimonious set of clusters with good predictive power. As a secondary goal, we plan in the future to develop Gibbs sampling methods for our framework.

2 Models

Our framework supports a broad class of models unified by the compositional structure shown in Fig. 1. Every model generates each data token x_n by assigning it to a single cluster indicated by discrete variable $z_n \in \{1, 2, \dots K, \dots\}$. If $z_n = k$, we draw x_n from an exponential family (EF) density with natural parameter ϕ_k: $p(x_n|\phi_k) \propto \exp[s(x_n)^T \phi_k]$,

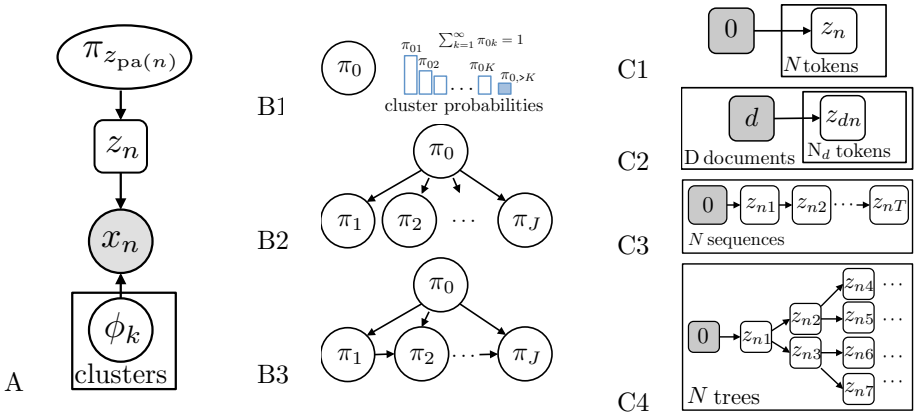

Figure 1: Our compositional view of clustering models. *Col. A:* Generative model for one data token x_n. *Col. B:* Possible dependency graphs for cluster probability vectors π: DP (top), HDP (middle), and dependent DPs (bottom). *Col. C:* Possible graphs for cluster indicators z.

where $s(x_n)$ is a sufficient statistic. This general EF form allows many data types (real, binary, discrete) but ensures tractable inference. Without the EF assumption, we cannot summarize large datasets via compact sufficient statistics, which complicates scalability.

The allocation of cluster assignments across a dataset requires two sets of variables: probability vectors π and indicators z. We have one z_n indicator for each data token, and one or more π_j nodes, each a positive vector that sums to unity with an entry for each cluster. By choosing a fixed graph structure for π and z, we encode structural assumptions into the model, as shown in Fig. 1. Always, indicator z_n is drawn from the distribution over clusters defined at one π_j node. The chosen π_j node is assigned by token n's parent in the z graph: $j = z_{\mathrm{pa}(n)}$. This restricts z to multiple trees, but still allows general π.

Our framework defines a single *allocation* model by combining fixed graph structures for π, z from columns B and C. Each model can be either parameteric or nonparametric, based on the prior distribution of the top-level π_0. The pair (B1,C1) yields mixture models [6], while (B2, C2) gives topic models [7, 8], and (B2, C3) gives hidden Markov models [9]. The pair (B2, C4) yields hidden Markov trees used for multi-scale image modeling [10, 11] and text parsing [12, 13]. B3 and C2 could yield a topic model where frequencies vary over time, as in [14]. This framework also extends to relational block models [15, 16], hierarchical or sticky sequential models [17, 18], and spatial models for image segmentation [19].

3 Variational Inference

Variational methods frame posterior inference as an optimization problem [3]. The expectation-maximization (EM) algorithm is a simple example, but BNP models require sophisticated methods that better manage uncertainty. These methods seek an approximate density q over hidden variables that is as close as possible in KL divergence to the true, intractable posterior. Factorization assumptions make q tractable by restricting each variable in z, π, ϕ to an EF density controlled by free parameters [3]. The goal of optimization is to update these free parameters to maximize an objective $\mathcal{L}$ that lower bounds the evidence: $\log p(x) \geq \mathcal{L}(\ldots)$. Closed-form iterative update algorithms exist for many cases.

Scalability. Recent stochastic algorithms [4] extend variational inference to large datasets by processing one small data subset (or "batch") at a time. These methods optimize a noisy function whose expectation is the whole-dataset objective $\mathcal{L}(\cdot)$. They are sensitive to the choice of learning rate, which must be carefully tuned for ideal performance. Further, the noisy objective can complicate decisions about adding or removing clusters in BNP models.

A promising alternative is *memoized* variational inference [5]. This method extends the incremental EM algorithm of [20] to BNP models, aggregating EF sufficent statistics across

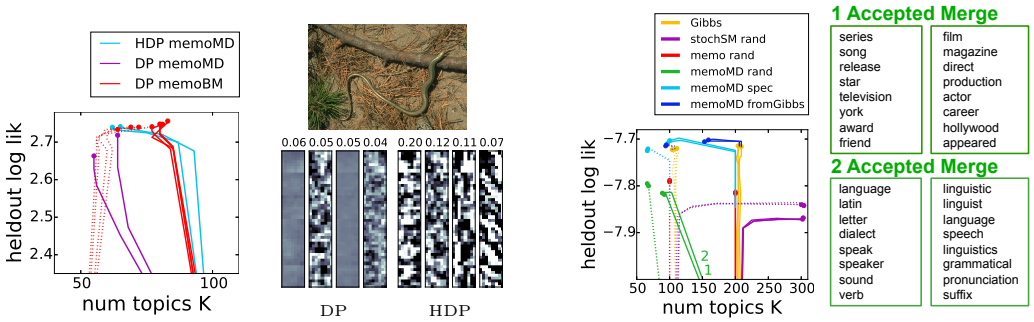

Figure 2: *Left: Image patches* - Heldout performance vs. K for runs with 50 (dashed) and 100 (solid) initial clusters, plus sample patches from final DP and HDP top-ranked clusters when applied to snake test image. *Right: Wikipedia articles* - Heldout performance vs. K for HDP algorithms with 100 (dashed) and 200 (solid) initial topics, plus two topic pairs accepted by our merge moves.

batch-by-batch updates to exactly optimize the whole-dataset objective. Memoized inference has the same run-time complexity as stochastic, but avoids learning rates entirely.

Reliability via merge, delete, and birth moves. Discovering a compact set of clusters benefits interpretability and improves algorithm speed. We use two non-local proposals that remove clusters: pair-wise *merges* eliminate redundancy and *deletes* remove unnecessary clusters. Given a candidate proposal, we evaluate $\mathcal{L}(\cdot)$ and accept if it improves. Memoization allows rapid construction and verification even for large datasets.

Escaping poor initialization requires adding useful clusters missing from the current model. We developed data-informed *birth* moves that can add many clusters at once, even if no single batch alone contains enough evidence for the cluster. When deployed for DP mixtures [5], these moves enable models started with one cluster to quickly reach hundreds if necessary.

Some previous efforts [21] employ similar non-local moves, but ours apply to a wider class of models and can scale to large datasets. Our memoized approach is crucial for this goal. Fig. 2 shows the poor performance of a stochastic split-and-merge method [22], likely caused by making decisions based on a noisy single-batch (not whole-dataset) objective.

4 Implementation and Results

We have prototyped our framework in an open-source Python package called `bnpy`. To run inference on a new dataset, we specify via pre-defined keywords an allocation model, data-generation method, and algorithm (with optional moves). We plan to move beyond keywords like "DPMix" for π, z pairs toward a general-purpose specification language. Command **A** below trains a DP-mixtures-of-Gaussians model on 3.5 million image patches, while **B** trains an HDP admixture on the same data. **C** trains an HDP topic model on Wikipedia articles.

```
A: run('ImgPatch', 'DPMix', 'Gauss', 'memo', moves='birth,merge')
B: run('ImgPatch', 'HDPAdmix', 'Gauss', 'memo', moves='merge,delete')
C: run('Wiki', 'HDPAdmix', 'Multinomial', 'stochastic')
```

Fig. 2 compares `bnpy` and competitor methods on these two applications. Trace plots show births (B), merges (M), and deletes (D) making big changes in the number of clusters while improving predictions on heldout data. In our HDP topic model comparison, neither Gibbs samplers [8] nor stochastic split-merge methods [22] make such productive changes. We have also verified our topic modeling algorithms on 1.8 million NY Times articles.

5 Conclusion

Via a modular approach to models and algorithms, `bnpy` empowers practioners to explore huge datasets without writing custom inference code. Our memoized variational approach scales like stochastic methods but offers more principled verification for moves that escape poor initializations to discover interpretable, compact structure.

References

[1] Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. Springer, 2010.

[2] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

[3] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[4] Matt Hoffman, David Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1), 2012.

[5] Michael C. Hughes and Erik B. Sudderth. Memoized online variational inference for dirichlet process mixture models. In *Neural Information Processing Systems*, 2013.

[6] David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

[7] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

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

[9] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden markov model. In *Neural Information Processing Systems*, 2001.

[10] Matthew S Crouse, Robert D Nowak, and Richard G Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902, 1998.

[11] Jyri J Kivinen, Erik B Sudderth, and Michael I Jordan. Learning multiscale representations of natural scenes using dirichlet processes. In *International Conference on Computer Vision*, 2007.

[12] J. R. Finkel, T. Grenager, and C. D. Manning. The infinite tree. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2007.

[13] Percy Liang, Slav Petrov, Michael I Jordan, and Dan Klein. The infinite pcfg using hierarchical dirichlet processes. In *Empirical Methods in Natural Language Processing*, 2007.

[14] David M Blei and John D Lafferty. Dynamic topic models. In *International Conference on Machine Learning*, pages 113–120, 2006.

[15] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. In *Neural Information Processing Systems*, 2009.

[16] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI Conference on Artificial Intelligence*, 2006.

[17] Emily B Fox, Erik B Sudderth, Michael I Jordan, Alan S Willsky, et al. A sticky hdp-hmm with application to speaker diarization. *Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[18] Katherine A Heller, Yee W Teh, and Dilan Görür. Infinite hierarchical hidden markov models. In *Artificial Intelligence and Statistics*, 2009.

[19] Erik B Sudderth and Michael I Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. In *Neural Information Processing Systems*, 2009.

[20] Radford M Neal and Geoffrey E Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[21] Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, and Geoffrey E Hinton. Smem algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.

[22] Michael Bryant and Erik B. Sudderth. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In *Neural Information Processing Systems*, 2012.