

hca: Tool for Multicore Non-parametric Topic Models

Wray Buntine, wray.buntine@monash.edu
Faculty of Information Technology
Monash University
Clayton, VIC, 3800, Australia

April 1, 2016

Abstract

This paper describes **hca**, an open source command-line tool to train and test topic models. The tool implements a variety of Gibbs samplers for non-parametric models using an efficient multicore implementation of hierarchical Pitman-Yor processes. These are used for both the document-topic component and the topic-word component, and to model burstiness of words in topics. Various diagnostics, document completion testing and coherence measurements with PMI are also supported. The package consists of a main command-line tool and a set of support utilities. The documentation includes a user's guide with a mini tutorial.

Keywords: topic model, hierarchical Pitman-Yor process, non-parametric Bayesian model, Gibbs sampling

1 Introduction

Topic models are a form of non-negative matrix factorisation and some versions also correspond to normalised independent (multinomial) components. They were originally developed as a Bayesian variant of probabilistic latent semantic analysis. The early model, Latent Dirichlet Allocation (LDA), used a simple symmetric Dirichlet prior for columns of the document-topic component and rows of the topic-word component, but researchers soon realised non-symmetric priors could improve modelling performance (Wallach et al., 2009). Using a Dirichlet process (DP) as a prior for columns

of the document-topic component was proposed by (Teh et al., 2006), and Teh distributed a relatively robust Gibbs sampler for the task (Teh, 2004). While this purportedly allows the model to “pick” the right number of topics, more importantly, it allows some topics to be rarer than others. In the simple symmetric model of LDA, all topics are assumed equally likely *a priori*.

Over the subsequent decade many research efforts have attempted to improve on Teh’s algorithm, using methods including collapsed Gibbs samplers, approximate fitting and variational methods. Others have developed alternative extensions to LDA. (Sato and Nakagawa, 2010) have also placed a Pitman-Yor process (PYP) as a prior for the rows of the topic-word component. The common prior acts like a “background” topic: it alleviates the need for non-topical words (like stop words) to be modelled by a single topic because they are shared across all topics. (Doyle and Elkan, 2009) have modelled rows of the topic-word component using a Dirichlet Compound multinomial, rather than a multinomial, in order to introduce “burstiness,” whereby some words in a document are encouraged to appear multiple times (in a “burst”). This tends to produce models with less topics, and improves perplexity. This is implemented with a PYP, yielding a small factor overhead in space and time over the standard collapsed Gibbs sampler.

While there are numerous other improvements made to vanilla topic modelling, for instance using an Indian buffet process (IBP) on both component matrices to sparsify them, from (Archambeau et al., 2015), these above are the ones reproduced in the tool `hca`:

- DPs and PYPs on both the document-topic component and the topic-word component,
- burstiness via an additional document specific PYP on topic-word component, and
- hyperparameter sampling by default using adaptive rejection sampling of (Gilks and Wild, 1992).

Results using the tool were presented at KDD 2014 by (Buntine and Mishra, 2014). The implementation uses the table indicator sampling method of (Chen et al., 2011) which is a collapsed version of Teh’s sampler but requires just a small factor in space and time overhead over the traditional collapsed sampler for LDA of (Griffiths and Steyvers, 2004). This collapses the sizes of the tables in the classic hierarchical Chinese restaurant samplers, leaving just the count of the number of tables. Most importantly, it uses no dynamic memory like the Chinese restaurant samplers.

2 The hca Commandline Tool

The `hca` tool is distributed on the MPL 2.0 license and is implemented as a Unix style command line tool with multiple options for tasks including specifying a variety of different non-parametric topic models, training and testing models, reporting diagnostics, initialising and controlling fitting of hyperparameters, restarts and checkpointing. These options are all documented in a standard Unix style man page with duplicate PDF version. Moreover, further examples of using the tool are given at the end of the man page.

The tool itself is implemented in C. A multicore version can be compiled which uses threads and atomic operations available in C11 to keep data consistent. This gets, for instance, about 5 times speed-up with 8 cores, and uses little memory duplication so can work with quite large data sets (gigabytes of text). In our experience, however, it does not work well with more cores like 32. The code is self-contained, with no library dependencies. Source code for tasks such as adaptive rejection sampling, and Gamma variable sampling, have been included from existing open source libraries to eliminate dependencies. Compilation and testing has been successful on Mac OSX and a variety of Unices, and single-core compilation has been done under Cygwin.

While the development repository is in Github¹, the best way to obtain the code is via MLOSS² because this combines the two components.

3 Using the System

While the tool accepts many different data formats, the most common is a format sometimes referred to as the LDA-C format. This is a sparse matrix format where each line represents a document, and the line contains a list of “index:count” pairs giving the non-zero entries of the document by word matrix. An example dataset (“ch.ldac” and its companion “ch.tokens” listing the words corresponding to the indices) is included in the release. This is composed of 400 news articles about church, “bagged up” as a data product from (Lewis et al., 2004). A typical sequence of commands is:

```
hca -v -C200 -K40 -q6 data/ch /tmp/C1
hca -v -v -V -r0 -C0 -orat,10 -e data/ch /tmp/C1
```

¹<https://github.com/wbuntine/topic-models> and <https://github.com/wbuntine/libstb>

²<http://www.mloss.org/software/view/527/>

The first line builds a 40-topic model with 300 Gibbs cycles using the default configuration for `hca`, run on 6 cores in 12 seconds. The default configuration has a DP on the document-word component, a PYP on the topic-word component, both with a GEM at their root, and does no burstiness modelling. The second lines reports on the learnt model and parameters giving the top ten words per topic. For instance, the PYP on the topic-word component settled on a discount of 0.31 and concentration of 209, and the DP on the document-word component settled on a concentration of 0.8. The three most populous topics have significant words that show the era of the articles (1986-1987):

```
teresa,missionaries,nirmala,calcutta,nursing,respirator
diana,parker,bowles,camilla,charles,divorce
appendix,crucitti,tumour,parkinson,trembling,pope
```

More examples of the system, for instance document completion testing (Buntine, 2009), MCMC estimates of the component matrices, and diagnostic reports of the generated topics are given in the document.

4 Related Software

A variety of software for HDP-LDA, which places a hierarchical DP on the document-topic component, were compared with `hca` and reported in (Buntine and Mishra, 2014). Not all software is distributed, so some comparisons were done by duplicating experiments. Performance can be evaluated in terms of test perplexity, and in terms of computational speed. However, different authors make subtle distinctions in their evaluation of perplexity so comparisons are treacherous. For instance, for document completion one may hold out every 4-th word of a document or the final quarter of a document. Since documents vary in topic across paragraphs, this difference can be significant. Moreover, some authors do a poor choice of hyperparameters, sometimes not fitting them, so reported results are not always significant.

`Mallet` (McCallum, 2002) has a asymmetric-symmetric LDA mode that behaves like a truncated HDP-LDA approximation. This is significantly faster than all others, and also supports multicore. However, it uses memory duplication with multicore so is not able to run on larger data sets. Similarly, `hca` is a factor faster than all others, except `Mallet`, and it supports multicore but without significant memory overhead. It is much more efficient in memory, partly due to its implementation in C: a factor of 3 compared to `onlinehdp` (Wang et al., 2011), and a factor of 5 or so compared

to HDP (Teh, 2004). The variational approximation of (Sato and Nakagawa, 2010) yields good performance in perplexity, comparable to HDP and Mallet. Of the HDP-LDA implementations, however, `hca` was significantly the best in terms of perplexity.

Most distributed implementations of topic modeling software perform poorly in nonparametric modelling of the topic-word component. An approximate implementation in Mallet reported negative experiments (Wallach et al., 2009). For `hca`, nonparametric modelling of the topic-word component usually significantly improves perplexity, and the root topic has a beautiful interpretation as a background topic that picks up non-topical words.

Many large scale implementations of LDA have been developed, scaling to billions of documents. A well known example is YahooLDA³, and indeed LDA is considered a test case for large scale distributed machine learning (Li et al., 2014b). Techniques have been developed to scale simpler non-parametric methods (Li et al., 2014a), such as HDP-LDA, but these are not yet generally available. Moreover, with billions of documents, it is not clear of high-fidelity topic models, produced by Bayesian non-parametric method, would be useful. Almost certainly one would build models where the numbers of topics are small relative to what statistically could be supported. This places learning in the “large sample” situation where one should best ignore Bayesian methods and focus on the optimisation problem.

Note, (Archambeau et al., 2015) model sparsity with an IBP. Their FTM models sparsity on the document-topic component, and their LIDA models sparsity in both document-topic and topic-word components. They report good results compared with Teh’s HDP, although they used a fixed setting for the concentration of the DP so it is not clear if the results would hold with proper fitting. Attempting to reproducing their results using `hca` with the same data sets yields the following results for log of the test-set perplexity.

Data	KOS	NIPS
FTM (3-par)	7.266±0.009	6.883±0.008
LIDA	7.257±0.010	6.795±0.007
HPD-LDA (<code>hca</code>)	7.253±0.003	6.792±0.002
NP-LDA (<code>hca</code>)	7.156±0.003	6.722±0.003

In these results, NP-LDA has a DP on the document-topic component and a PYP on the topic-word component.

In summary, in terms of sheer speed and software support for desktop use, Mallett is a clear winner. When high performance in terms of perplex-

³See https://github.com/sudar/Yahoo_LDA

ity is required, and with a fully non-parametric implementation, `hca` offers an alternative.

Acknowledgements

Some parts of the software were co-authored by, and benefited greatly from discussions with Dr. Lan Du and Swapnil Mishra. The diagnostics were developed with consultation from Christopher Ball of Metaheuristica. The software was developed at NICTA Canberra and Monash University, based on earlier infrastructure developed at Helsinki Institute for Information Technology.

References

- Archambeau, C., Lakshminarayanan, B., and Bouchard, G. (2015). Latent IBP compound Dirichlet allocation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):321–333.
- Buntine, W. (2009). Estimating likelihoods for topic models. In *Proceedings of the 1st Asian Conference on Machine Learning*, Nanjing, China.
- Buntine, W. and Mishra, S. (2014). Experiments with non-parametric topic models. In *20th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, page 881890.
- Chen, C., Du, L., and Buntine, W. (2011). Sampling table configurations for the hierarchical Poisson-Dirichlet process. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD*, pages 296–311. Springer.
- Doyle, G. and Elkan, C. (2009). Accounting for burstiness in topic models. In *Proc. of the 26th Annual Int. Conf. on Machine Learning, ICML '09*, pages 281–288.
- Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, pages 337–348.
- Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.
- Lewis, D., Yang, Y., Rose, T., and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5.

- Li, A., Ahmed, A., Ravi, S., and Smola, A. (2014a). Reducing the sampling complexity of topic models. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*.
- Li, M., Anderson, D., Park, J., Smola, A., Ahmed, A., Josifovski, V., Long, J., Shekita, E., and Su, B.-Y. (2014b). Scaling distributed machine learning with the parameter server. In *Operating Systems Design and Implementation (OSDI)*, pages 583–598.
- McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Sato, I. and Nakagawa, H. (2010). Topic models with power-law using Pitman-Yor process. In *16th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, pages 673–682. ACM.
- Teh, Y. W. (2004). Nonparametric Bayesian mixture models - release 2.1. <http://www.stats.ox.ac.uk/~teh/software.html>. MATLAB and C code [Online; accessed 18 March 2016].
- Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the ASA*, 101(476):1566–1581.
- Wallach, H., Mimno, D., and McCallum, A. (2009). Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems 19*.
- Wang, C., Paisley, J., and Blei, D. (2011). Online variational inference for the hierarchical Dirichlet process. In *AISTATS '11*.