BoltzMM: an R package for maximum pseudolikelihood estimation of fully-visible Boltzmann machines

Andrew T. Jones[1], **Jessica J. Bagnall**[2], **and Hien D. Nguyen**[2]

1 School of Mathematics and Physics, University of Queensland, St. Lucia 4072, Queensland Australia **2** Department of Mathematics and Statistics, La Trobe University, Bundoora 3086, Victoria Australia

Summary

The `BoltzMM` package (Version 0.1.3; https://CRAN.R-project.org/package=BoltzMM) for the `R` statistical programming environment (R Core Team, 2018) provides a complete suite of functions for application and estimation of *fully-visible Boltzmann Machine* (FVBM) models. The FVBM was first considered in Hyvarinen (2006), and can be described as follows.

Let $\boldsymbol{X} \in \mathbb{X}$ (where $\mathbb{X} = \{-1, +1\}^d$ is the set of *spin-binary strings* of dimension d) be a random variable with *probability mass function* (PMF) of the form

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \exp\left(\frac{1}{2}\boldsymbol{x}^\top \mathbf{M}\boldsymbol{x} + \mathbf{b}^\top \boldsymbol{x}\right)/z(\boldsymbol{\theta}),$$

where

$$z(\boldsymbol{\theta}) = \sum_{\boldsymbol{\xi} \in \mathbb{X}} \exp\left(\frac{1}{2}\boldsymbol{\xi}^\top \mathbf{M}\boldsymbol{\xi} + \mathbf{b}^\top \boldsymbol{\xi}\right),$$

is a normalizing constant, $\mathbf{b} \in \mathbb{R}^d$, and $\mathbf{M} \in \mathbb{R}^{d \times d}$ is a symmetric matrix with zero diagonal. We put the unique elements of the *bias vector* $\mathbf{b}$ and the *interaction matrix* $\mathbf{M}$ into the *parameter vector* $\boldsymbol{\theta}$. Interpretations of the parameter vector elements of an FVBM can be found in Section 2 of Bagnall, Jones, Karavarsamis, & Nguyen (2018).

The FVBM model can be viewed as a multivariate extension of the classical Bernoulli distribution, on spin-binary random variable strings (i.e, random variables $\boldsymbol{X} \in \mathbb{X}$, rather $\boldsymbol{X} \in \{0, 1\}^d$), and can be demonstrated to be equivalent to the logistic multivariate binary model that was proposed by Cox (1972). It can also be considered as a fully-connected binary graphical model (see Bagnall et al., 2018 for more details). Simply put, the FVBM is a model that allows for the characterization of interactions between multiple binary random variables, simultaneously.

Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in \mathbb{X}$ be a sample of n *independent and identically distributed* (IID) observations from an FVBM with unknown parameter vector $\boldsymbol{\theta}_0$. In Hyvarinen (2006), it was proved that $\boldsymbol{\theta}_0$ could be consistently estimated from $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in \mathbb{X}$ via the so-called *maximum pseudolikelihood estimator* (MPLE) of the type described in Lindsay (1988). Furthermore, in H. D. Nguyen & Wood (2016a), a *minorization-maximization* (MM; see Hunter & Lange, 2004; H. D. Nguyen, 2017) was proposed for the computation of the MPLE. The asymptotic normality of the MPLE was proved in H. D. Nguyen & Wood (2016b), which allows for the use of the MPLE of the FVBM as both a point estimator and a method for constructing useful hypothesis tests regarding relationships between binary random variables.

The `BoltzMM` package makes the technical results above accessible. The most fundamental of the functions of `BoltzMM` are `pvfbm` and `rfvbm`, which allow for the computation of the probability of any binary string, given valid input bias vector and interaction matrix, and the random generation of binary strings, given input parameter elements, respectively. Since the support $\mathbb{X}$ is finite, it is often desirable to compute the probabilities of all possible outcomes, which can be achieved using the function `allpfvbm`. It is also often desirable to compute the marginal probability of any particular string index

$$f_i\left(x_i; \boldsymbol{\theta}\right) = \mathbb{P}\left(X_i = x_i\right),$$

where $i = 1, \ldots, d$, $X_i \in \{-1, 1\}$, and $\boldsymbol{X}^\top = (X_1, \ldots X_d)$, for any parameter vector. This can be achieved using the function `marginpfvbm`.

The function `fitfvbm` implements the MM algorithm of H. D. Nguyen & Wood (2016a) in order to compute the MPLE of an unknown FVBM parameter vector $\boldsymbol{\theta}_0$, from an n observation realization of the sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ (i.e., $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$). That is, `fitfvbm` takes in the data as a $n \times d$ dimensional matrix, where each row is a spin-binary string of length d. Using the `fitfvbm` function output, and the asymptotic normality results of H. D. Nguyen & Wood (2016b), standard errors can be computed using the function `fvbmstderr`, with the functions `fvbmcov`, `fvbmHess`, and `fvbmpartiald` providing intermediate calculations required by `fvbmstderr`.

As described in H. D. Nguyen & Wood (2016b) and demonstrated in Bagnall et al. (2018), hypothesis testing can be conducted using Wald-type hypothesis tests. These Wald-type tests can be constructed for testing of hypotheses of the form

$$\text{H}_0\colon \theta_{0k} = \theta_k, \text{ versus } \text{H}_1\colon \theta_{0k} \neq \theta_k,$$

where θ_{0k} is the kth element of the unknown parameter vector $\boldsymbol{\theta}_0$, and θ_k is some hypothesized value that it may take. The p-values and z-scores for these hypothesis tests can be computed using the outputs from `fitfvbm` and `fvbmstderr`, via the function `fvbmtests`.

All of the functions have been documented, and the documentation and examples for each function can be accessed via the `help` function in R. The analysis of data arising from the voting patterns of the Senate of the 45th Australian Parliament in 2016, using the `BoltzMM` package, can be found in Bagnall et al. (2018). The report describes the manner in which the FVBM can be used to conduct inference regarding multivariate binary data. The data that were analysed in the report are included in the package and can be accessed via the command `data(senate)`.

The `BoltzMM` package is programmed in native R, with computationally intensive subroutines programmed in C and integrated via the `Rcpp` and `RcppArmadillo` packages of Eddelbuettel (2013). The package is therefore suitable for analysis of large data sets. On https://cran.r-project.org, the only other package that provides facilities for estimation of *Boltzmann machine* (BM) models, of any kind, is the `deepnet` package of Rong (2014). Here, facilities for estimation of the so-called *restricted Boltzmann machine* (RBM) are included, via the functions `rbm.down`, `rbm.train`, and `rbm.up`.

The RBM is an alternative simplification, to the FVBM, of the BM models first considered by Ackley, Hinton, & Sejnowski (1985). The BM is a parametric probabilistic model that can closely approximate any PMF on $\mathbb{X}$ (see, e.g., Le Roux & Bengio, 2008). Unfortunately, due to the intractability of its requirement for hidden (or unobserved) variables, it is often difficult to efficiently and meaningfully conduct parameter estimation when both dimension d and sample size n are large. Like the BM models, estimation of RBM models is also intractable, due to the requirement for hidden random variables. These hidden variables also prevent the establishment of asymptotics for the RBM parameter estimators, which makes the statistical inference of such models limited. Thus we view the functions from `deepnet` as being complementary but not overlapping with `BoltzMM`.

Users can obtain the latest build of `BoltzMM` on GitHub. The latest stable build can be obtained from CRAN (https://CRAN.R-project.org/package=BoltzMM), and an archival build can be obtained from Zenodo. Algorithm derivations and theoretical results regarding the implemented functions can be found in H. D. Nguyen & Wood (2016b) and H. D. Nguyen & Wood (2016a). An example application of the `BoltzMM` package for the analysis of real data is described in Bagnall et al. (2018). Thorough descriptions of the package functions appear in the manual, which can be accessed at https://cran.r-project.org/web/packages/BoltzMM/BoltzMM.pdf. Bug reports and other feedback can be directed to the GitHub issues page.

Acknowledgements

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*, 147–169. doi:10.1016/S0364-0213(85)80012-4

Bagnall, J., Jones, A., Karavarsamis, N., & Nguyen, H. (2018). *The fully-visible Boltzmann machine and the Senate of the 45th Australian Parliament in 2016*. SSRN. doi:10.2139/ssrn.3287903

Cox, D. R. (1972). The analysis of multivariate binary data. *Journal of the Royal Statistical Society C*, *21*(113-120). doi:10.2307/2346482

Eddelbuettel, D. (2013). *Seamless R and C++ Integration with Rcpp*. New York: Springer. doi:10.1007/978-1-4614-6868-4

Hunter, D. R., & Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, *58*, 30–37. doi:10.1198/0003130042836

Hyvarinen, A. (2006). Consistency of pseudolikelihood estimation of fully visible Boltzmann machines. *Neural Computation*, *18*, 2283–2292. doi:10.1162/neco.2006.18.10.2283

Le Roux, N., & Bengio, Y. (2008). Representation power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, *20*, 1631–1649. doi:10.1162/neco.2008.04-07-510

Lindsay, B. (1988). Composite likelihood methods. *Contemporary Mathematics*, *8*, 221–239. doi:10.1090/conm/080/999014

Nguyen, H. D. (2017). An introduction to MM algorithms for machine learning and statistical estimation. *WIREs Data Mining and Knowledge Discovery*, *7*, e1198. doi:10.1002/widm.1198

Nguyen, H. D., & Wood, I. A. (2016a). A block successive lower-bound maximization algorithm for the maximum pseudolikelihood estimation of fully visible Boltzmann machines. *Neural Computation*, *28*, 485–492. doi:10.1162/NECO_a_00813

Nguyen, H. D., & Wood, I. A. (2016b). Asymptotic normality of the maximum pseudolikelihood estimator for fully visible Boltzmann machines. *IEEE Transactions on Neural Networks and Learning Systems*, *27*, 897–902. doi:10.1109/TNNLS.2015.2425898

R Core Team. (2018). *R: a language and environment for statistical computing.* R Foundation for Statistical Computing.

Rong, X. (2014). *deepnet: deep learning toolkit in R.* Retrieved from https://CRAN.R-project.org/package=deepnet