

The Libra Toolkit for Probabilistic Models

Daniel Lowd

Amirmohammad Rooshenas

Department of Computer and Information Science

University of Oregon

Eugene, OR 97403, USA

LOWD@CS.UOREGON.EDU

PEDRAM@CS.UOREGON.EDU

Editor: Antti Honkela

Abstract

The Libra Toolkit is a collection of algorithms for learning and inference with discrete probabilistic models, including Bayesian networks, Markov networks, dependency networks, and sum-product networks. Compared to other toolkits, Libra places a greater emphasis on learning the structure of tractable models in which exact inference is efficient. It also includes a variety of algorithms for learning graphical models in which inference is potentially intractable, and for performing exact and approximate inference. Libra is released under a 2-clause BSD license to encourage broad use in academia and industry.

Keywords: probabilistic graphical models, structure learning, inference

1. Introduction

The Libra Toolkit is a collection of algorithms for learning and inference with probabilistic models in discrete domains. What distinguishes Libra from other toolkits is the types of methods and models it supports. Libra includes a number of algorithms for *structure learning for tractable probabilistic models* in which exact inference can be done efficiently. Such models include sum-product networks (SPN), mixtures of trees (MT), and Bayesian and Markov networks with compact arithmetic circuits (AC). These learning algorithms are not available in any other open-source toolkit. Libra also supports *structure learning for graphical models*, such as Bayesian networks (BN), Markov networks (MN), and dependency networks (DN), in which inference is not necessarily tractable. Some of these methods are unique to Libra as well, such as using dependency networks to learn Markov networks. Libra provides a variety of exact and approximate inference algorithms for answering probabilistic queries in learned or manually specified models. Many of these are designed to exploit local structure, such as conjunctive feature functions or tree-structured conditional probability distributions.

The overall goal of Libra is to make these methods available to researchers, practitioners, and students for use in experiments, applications, and education. Each algorithm in Libra is implemented in a command-line program suitable for interactive use or scripting, with consistent options and file formats throughout the toolkit. Libra also supports the development of new algorithms through modular code organization, including shared libraries for different representations and file formats.

Learning General Models	
○	BN structure with tree CPDs (Chickering et al., 1997)
	DN structure with tree/boosted tree/LR CPDs (Heckerman et al., 2000)
●	MN structure from DNs (Lowd, 2012)
	MN parameters (pseudo-likelihood)
Learning Tractable Models	
●	Tractable BN/AC structure (Lowd and Domingos, 2008)
●	Tractable MN/AC structure (Lowd and Rooshenas, 2013)
●	Mixture of trees (MT) (Meila and Jordan, 2000)
●	SPN structure (ID-SPN algorithm) (Rooshenas and Lowd, 2014)
	Chow-Liu algorithm (Chow and Liu, 1968)
●	AC parameters (maximum likelihood)
Approximate Inference	
	Gibbs sampling (BN,MN,●DN) (Heckerman et al., 2000) (DN)
	Mean field (BN,MN,●DN) (Lowd and Shamaei, 2011) (DN)
	Loopy belief propagation (BN,MN)
	Max-product (BN,MN)
	Iterated conditional modes (BN,MN,●DN)
●	Variational optimization of ACs (Lowd and Domingos, 2010)
Exact Inference	
○	AC variable elimination (BN,MN) (Chavira and Darwiche, 2007)
○	Marginal and MAP inference (AC,SPN,MT) (Darwiche, 2003)

Table 1: Learning and inference algorithms implemented in Libra. Filled circles (●) indicate algorithms that are unique to Libra, and hollow circles (○) indicate algorithms with no other open-source implementation.

Libra is available under a modified (2-clause) BSD license, which allows modification and reuse in both academia and industry. Libra’s source code and documentation can be found at <http://libra.cs.uoregon.edu>.

2. Functionality

Libra includes a variety of learning and inference algorithms, many of which are not available in any other open-source toolkit. See Table 1 for a brief overview.

Libra’s command-line syntax is designed to be simple. For example, to learn a tractable BN, run the command: “`libra acbn -i train.data -mo model.bn -o model.ac`” where `train.data` is the input data, `model.bn` is the filename for saving the learned BN, and `model.ac` is the filename for the corresponding AC representation, which allows for efficient, exact inference. To compute exact conditional marginals in the learned model: “`libra acquery -m model.ac -ev test.ev -marg`”. To compute approximate marginals in the BN with loopy belief propagation: “`libra bp -m model.bn -ev test.ev`”. Additional command-line parameters can be used to specify other options, such as the priors and heuristics used by `acbn` or the maximum number of iterations for `bp`. These are just three of more than twenty commands included in Libra.

Libra supports a variety of file formats. For data instances, Libra uses comma separated values, where each value is a zero-based index indicating the discrete value of the corresponding variable. For evidence and query files, unknown or missing values are represented with the special value “*”. For model files, Libra supports the XMOD representation from

Toolkit	Representation		Inference		Learning	
	Model Types	Factors	Exact	Approx.	Param.	Structure
Libra	BN,MN,DN,SPN,AC	Tree,Feature	ACVE	G,BP,MF	ML,PL	BN,...,AC
FastInf	BN,MN	Table	JT	Many	ML,EM	-
libDAI	BN,MN	Table	JT,E	Many	ML,EM	-
OpenGM2	BN,MN	Sparse	-	Many	-	-
Banjo	BN,DBN	Table	-	-	-	BN
BNT	BN,DBN,ID	LR,OR,NN	JT,VE,E	G,LW,BP	ML,EM	BN
Deal	BN	Table	-	-	-	BN
OpenMarkov	BN,MN,ID	Tree,ADD,OR	JT	LW	ML	BN,MN
SMILE	BN,DBN,ID	Table	JT	Sampling	ML,EM	BN
UnBBayes	BN,ID	Table	JT	G,LW	-	BN

Table 2: Comparison of Libra to several other probabilistic inference and learning toolkits.

the WinMine Toolkit, the Bayesian interchange format (BIF), and the simple representation from the UAI inference competition. Libra converts among these different formats using the provided `mconvert` utility, as well as to its own internal formats for BNs, MNs, and DNs (`.bn`, `.mn`, `.dn`). Libra has additional representations for ACs and SPNs (`.ac`, `.spn`). These formats are designed to be easy for humans to read and programs to parse.

Libra is implemented in OCaml. OCaml is a statically typed language that supports functional and imperative programming styles, compiles to native machine code on multiple platforms, and uses type inference and garbage collection to reduce programmer errors and effort. OCaml has a good foreign function interface, which Libra uses for linking to C libraries and a few memory-intensive subroutines. The code to Libra includes nine support libraries, which provide modules for input, output, and representation of different types of models, as well as commonly used algorithms and utility methods.

3. Comparison to Other Toolkits

In Table 2, we compare Libra to other toolkits in terms of representation, learning, and inference.

In terms of representation, Libra is the only open-source software package that supports ACs and one of a very small number that support DNs or SPNs. Libra does not currently support dynamic Bayesian networks (DBN) or influence diagrams (ID). For factors, Libra supports tables, trees, and arbitrary conjunctive feature functions. BNT (Murphy, 2001) and OpenMarkov (CISIAD, 2013) also support additional types of CPDs, such as logistic regression, noisy-OR, neural networks, and algebraic decision diagrams, but they only support tabular CPDs for structure learning. OpenGM2 (Andres et al., 2012) supports sparse factors, but iterates through all factor states during inference. Libra is unique in its ability to learn models with local structure and exploit that structure in inference.

For exact inference, the most common algorithms are junction tree (JT), enumeration (E), and variable elimination (VE). Libra provides ACVE (Chavira and Darwiche, 2007), which is similar to building a junction tree, but it can exploit structured factors to run inference in many high-treewidth models. For approximate inference, Libra provides Gibbs sampling (G), loopy belief propagation (BP), and mean field (MF), all of which are optimized for structured factors. A few learning toolkits offer likelihood weighting (LW) or

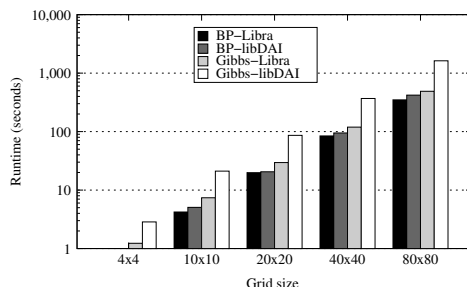


Figure 1: Running time of belief propagation and Gibbs sampling in Libra and libDAI, evaluated on grid-structured MNs of various sizes.

additional sampling algorithms for BNs. FastInf (Jaimovich et al., 2010), libDAI (Mooij, 2010), and OpenGM2 offer the most algorithms but only support tables.

For learning, Libra supports maximum likelihood (ML) parameter learning for BNs, ACs, and SPNs, and pseudo-likelihood (PL) optimization for MNs and DNs. Libra does not yet support expectation maximization (EM) for learning with missing values. Structure learning is one of Libra’s greatest strengths. Most toolkits only provide algorithms for learning BNs with tabular CPDs or MNs using the PC algorithm (Spirtes et al., 1993). Libra includes methods for learning BNs, MNs, DNs, SPNs, and ACs, and all of its algorithms support learning with local structure.

In experiments on grid-structured MNs, Libra’s implementations of BP and Gibbs sampling were at least as fast as libDAI, a popular C++ implementation of many inference algorithms. The accuracy of both toolkits was equivalent. Parameter settings, such as the number of iterations, were identical. See Figure 1 for more details.

4. Conclusion

The Libra Toolkit provides algorithms for learning and inference in a variety of probabilistic models, including BNs, MNs, DNs, SPNs, and ACs. Many of these algorithms are not available in any other open-source software. Libra’s greatest strength is its support for tractable probabilistic models, for which very little other software exists. Libra makes it easy to use these state-of-the-art methods in experiments and applications, which we hope will accelerate the development and deployment of probabilistic methods.

Acknowledgments

The development of Libra was partially supported by ARO grant W911NF-08-1-0242, NSF grant IIS-1118050, NIH grant R01GM103309, and a Google Faculty Research Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, NIH, or the United States Government.

References

- B. Andres, T. Beier, and J. H. Kappes. OpenGM: A C++ library for discrete graphical models. *ArXiv e-prints*, 2012. URL <http://arxiv.org/abs/1206.0111>.
- M. Chavira and A. Darwiche. Compiling Bayesian networks using variable elimination. In *IJCAI*, pages 2443–2449, 2007.
- D. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *UAI*, pages 80–89, 1997.
- C. K. Chow and C. N Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.
- Research Center on Intelligent Decision-Support Systems (CISIAD). OpenMarkov 0.1.3. 2013. <http://www.openmarkov.org>.
- A. Darwiche. A differential approach to inference in Bayesian networks. *JACM*, 50(3): 280–305, 2003.
- D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *JMLR*, 1:49–75, 2000.
- A. Jaimovich, O. Meshi, I. McGraw, and G. Elidan. FastInf: An efficient approximate inference library. *JMLR*, 11:1733–1736, 2010.
- D. Lowd. Closed-form learning of Markov networks from dependency networks. In *UAI*, 2012.
- D. Lowd and P. Domingos. Learning arithmetic circuits. In *UAI*, 2008.
- D. Lowd and P. Domingos. Approximate inference by compilation to arithmetic circuits. In *NIPS*, 2010.
- D. Lowd and A. Rooshenas. Learning Markov networks with arithmetic circuits. In *AISTATS*, 2013.
- D. Lowd and A. Shamaei. Mean field inference in dependency networks: An empirical study. In *AAAI*, 2011.
- M. Meila and M. Jordan. Learning with mixtures of trees. *JMLR*, 1:1–48, 2000.
- Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *JMLR*, 11:2169–2173, 2010.
- K. Murphy. The Bayes net toolbox for MATLAB. *Computing Sci. and Statistics*, 33:2001.
- A. Rooshenas and D. Lowd. Learning sum-product networks with direct and indirect interactions. In *ICML*, 2014.
- P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer, New York, NY, 1993.