libDAI - a FOSS library for Discrete Approximate Inference methods

Joris Mooij[1]

[1]Max Planck Institute for Biological Cybernetics
Tübingen, Germany
joris.mooij@tuebingen.mpg.de

Whistler, December 12, 2008

MAX-PLANCK-GESELLSCHAFT

MPI FOR BIOLOGICAL CYBERNETICS

Radboud University Nijmegen

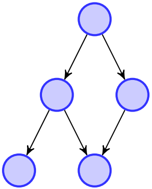
ICIS

Graphical models

- Bayesian Networks

$$\mathbb{P}(\mathbf{x}) = \prod_i \mathbb{P}(x_i \,|\, x_{\mathrm{pa}(i)})$$

where $\mathrm{pa}(i)$ are the *parents* of a node i in a DAG;



- Markov Random Fields

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

where $\mathcal{C}$ are *cliques* of an undirected graph;

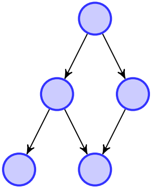
- Generalization of both: factor graphs

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

where $\mathcal{F}$ are *factor* nodes in a *factor graph*.

Graphical models

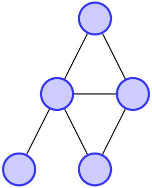
- Bayesian Networks



$$\mathbb{P}(\mathbf{x}) = \prod_i \mathbb{P}(x_i \mid x_{\mathrm{pa}(i)})$$

where $\mathrm{pa}(i)$ are the *parents* of a node i in a DAG;

- Markov Random Fields



$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

where $\mathcal{C}$ are *cliques* of an undirected graph;

- Generalization of both: factor graphs

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

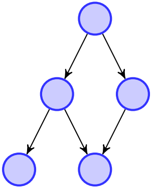
where $\mathcal{F}$ are *factor* nodes in a *factor graph*.

Graphical models

- Bayesian Networks

$$\mathbb{P}(\mathbf{x}) = \prod_i \mathbb{P}(x_i \mid x_{\mathrm{pa}(i)})$$

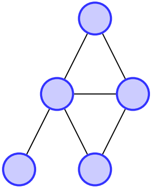
where $\mathrm{pa}(i)$ are the *parents* of a node i in a DAG;



- Markov Random Fields

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

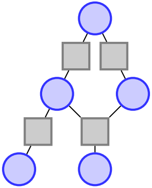
where $\mathcal{C}$ are *cliques* of an undirected graph;



- Generalization of both: factor graphs

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

where $\mathcal{F}$ are *factor* nodes in a *factor graph*.



Approximate inference in graphical models

Given a factor graph

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_I),$$

the following inference tasks are important:

- Calculate the partition sum:

$$Z = \sum_{x_1} \cdots \sum_{x_N} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

- Calculate the marginal distribution of a subset of variables $\{x_i\}_{i \in A}$:

$$\mathbb{P}(\mathbf{x}_A) = \frac{1}{Z} \sum_{\mathbf{x}_{\setminus A}} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

- Calculate the MAP state that has maximal probability mass:

$$\operatorname*{argmax}_{\mathbf{x}} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

Approximate inference in graphical models

Given a factor graph

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_I),$$

the following inference tasks are important:

- Calculate the partition sum:

$$Z = \sum_{x_1} \cdots \sum_{x_N} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

- Calculate the marginal distribution of a subset of variables $\{x_i\}_{i \in A}$:

$$\mathbb{P}(\mathbf{x}_A) = \frac{1}{Z} \sum_{\mathbf{x}_{\setminus A}} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

- Calculate the MAP state that has maximal probability mass:

$$\operatorname*{argmax}_{\mathbf{x}} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

Approximate inference in graphical models

Given a factor graph

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_I),$$

the following inference tasks are important:

- Calculate the partition sum:

$$Z = \sum_{x_1} \cdots \sum_{x_N} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

- Calculate the marginal distribution of a subset of variables $\{x_i\}_{i \in A}$:

$$\mathbb{P}(\mathbf{x}_A) = \frac{1}{Z} \sum_{\mathbf{x}_{\setminus A}} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

- Calculate the MAP state that has maximal probability mass:

$$\operatorname*{argmax}_{\mathbf{x}} \prod_{I \in \mathcal{F}} \psi_I(x_I)$$

Algorithms currently implemented in libDAI

libDAI provides implementations of various algorithms that solve these inference tasks (either exactly or approximately) for factor graphs with discrete variables.

Currently, libDAI contains implementations of the following algorithms:

- Exact inference by brute force enumeration
- Exact inference by junction-tree methods
- Mean Field
- (Loopy) Belief Propagation
- Tree Expectation Propagation (Minka & Qi 2004)
- Generalized Belief Propagation (Yedidia, Freeman & Weiss 2005)
- Double-loop GBP (Heskes, Albers & Kappen 2003)
- Loop Corrected Belief Propagation (Mooij & Kappen 2007; Montanari & Rizzo 2005)
- Gibbs sampling

Algorithms planned for the next release

The following algorithms are planned to be added to the next release:

- Iterative Join-Graph Propagation (Dechter, Kask & Mateescu, 2002)
- Tree Reweighted approximations/bounds (Wainwright, Jaakkola & Willsky, 2005)
- Methods for bounding marginals:
  - Box Propagation (Mooij & Kappen, 2008)
  - Ihler's BP accuracy bounds (Ihler, 2007)
  - Bound Propagation (Leisink & Kappen, 2003)

libDAI features

Key features of libDAI are:

- Free and open source (license: GPL v2+)
- C++ library (MatLab would be orders of magnitude slower)
- Command line interface and a (rudimentary) MatLab interface
- Modular design: easy to add algorithms
- Doxygen documentation (target for next release)
- Compiles out-of-the-box with GCC versions 4.1 and higher under GNU/Linux, and also with MS Visual Studio 2008 under Windows.

Target audience

libDAI is targeted at researchers that have a good understanding of graphical models.

The best way to use libDAI is by writing C++ code that invokes the library.

libDAI can be used to implement new approximate inference algorithms and to easily compare the performance with existing methods.

Non-features of libDAI

- No learning algorithms
- Only supports libDAI factor graph file format
- No GUI provided
- Limited visualization functionalities

Downloads

Releases can also be obtained from MLOSS at

`http://mloss.org/software/view/77/`

The newest code can be obtained from the public git repository

`git://git.tuebingen.mpg.de/libdai.git`

which can be accessed with a web browser as well at

`http://git.tuebingen.mpg.de/libdai`

Last release: 0.2.2 - September 30, 2008

Related software

Other open source software packages supporting both directed and undirected graphical models are:

- Bayes Net Toolbox (BNT) by Kevin Murphy, written in MatLab/C
- Probabilistic Networks Library (PNL) from Intel, written in C++

	libDAI	BNT	PNL
Exact (brute force)	+	+	+
Exact (junction tree)	+	+	+
(Loopy) Belief Propagation	+	+	+
Gibbs sampling	+	+	+
Mean Field	+	-	-
Tree Expectation Propagation	+	-	-
Generalized Belief Propagation	+	-	-
Loop Corrected Belief Propagation	+	-	-
Continuous variables	-	+	+
Dynamic Bayes Nets	-	+	+
Parameter Learning	-	+	+
Structure Learning	-	+	+

Demo

Thank you!

Thank you for your attention!

Thanks to all people who contributed to libDAI! (Martijn Leisink, Giuseppe Passino, Frederik Eaton, Bastian Wemmenhove, Christian Wojek, Claudio Lima, Jiuxiang Hu, Peter Gober)

In particular, thanks to Martijn Leisink for providing the basis from which libDAI has been derived.

Acknowledgments and references

Acknowledgments

This work is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant BSIK03024.

References

[**KFL01**] F. R. Kschischang and B. J. Frey and H.-A. Loeliger (2001): *Factor Graphs and the Sum-Product Algorithm*, IEEE Transactions on Information Theory 47(2):498-519.
[**MiQ04**] T. Minka and Y. Qi (2004): *Tree-structured Approximations by Expectation Propagation*, Advances in Neural Information Processing Systems (NIPS) 16.
[**MoR05**] A. Montanari and T. Rizzo (2005): *How to Compute Loop Corrections to the Bethe Approximation*, Journal of Statistical Mechanics: Theory and Experiment 2005(10)-P10011.
[**YFW05**] J. S. Yedidia and W. T. Freeman and Y. Weiss (2005): *Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms*, IEEE Transactions on Information Theory 51(7):2282-2312.
[**HAK03**] T. Heskes and C. A. Albers and H. J. Kappen (2003): *Approximate Inference and Constrained Optimization*, Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03) pp. 313-320.
[**MoK07**] J. M. Mooij and H. J. Kappen (2007): *Loop Corrections for Approximate Inference on Factor Graphs*, Journal of Machine Learning Research 8:1113-1143.
[**DKM02**] R. Dechter and K. Kask and R. Mateescu (2002): *Iterative Join-Graph Propagation*, Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02) pp. 128-32.
[**WJW05**] M. J. Wainwright and T. Jaakkola and A. S. Willsky (2005) *A New Class of Upper Bounds on the Log Partition Function*, IEEE Transactions on Information Theory 51:2313-2335.
[**MoK08**] J. M. Mooij and H. J. Kappen (2008): *Bounds on marginal probability distributions*, NIPS*2008.
[**Ihl07**] A. Ihler (2007): *Accuracy bounds for belief propagation*, Proceedings of the 23th Annual Conference on Uncertainty in Artificial Intelligence (UAI-07).
[**LeK03**] M. Leisink and B. Kappen (2003): *Bound Propagation*, Journal of Artificial Intelligence Research 19:139-154.