Journal of Machine Learning Research () Submitted ; Published

GrMPy: A Python-based software package for graphical models.

Almero Gouws 14366037@sun.ac.za
Department of Electrical and Electronic Engineering
University of Stellenbosch
Matieland 7602, South Africa

Editor:

Abstract

In this article we introduce GrMPy, an open-source Python library of classes and functions for learning and inference in graphical models. GrMPy's features include exact and approximate inference, as well parameter learning via maximum-likelihood estimation and the generalized EM algorithm for both Markov random fields and Bayesian networks. GrMPy is available freely under the GPL open-source license and the source is available at http://dip.sun.ac.za/vision/trac-git/agouws-GrMPy.bzr.

Keywords: Bayesian Networks, Markov Random Fields, Graphical models, Open-source software, Python

1. Introduction

Graphical models (Bishop, 2006) such as Bayesian networks and Markov random fields are widely applied to areas such as speech recognition and computer vision. GrMPy offers an open-source, easy to use toolbox of classes and functions for graphical models, written in the freely available language Python. The library has been designed with several uses in mind, including incorporation into users own applications, or as a guide and example for users writing their own application-specific graphical model code. The toolbox is available to researchers for comparison, and improvement of graphical model algorithms.

2. Features

GrMPy can be applied to a wide range of graphical model problems, and the most prominent features are covered in this section.

2.1 Models

GrMPy allows for the creation and manipulation of both directed Bayesian Networks and undirected Markov random fields. In both cases the user must define the graph structure as an adjacency matrix. When defining Bayesian networks, the user needs to specify the conditional probability distribution for each node. When defining a Markov random field, the user needs to define the cliques and the potentials over those cliques in the graph. The parameters of both the conditional probability distributions and clique potentials can also be determined from observed evidence as described in section 3.3. Currently GrMPy only supports *discrete* conditional probability distributions and clique potentials.

2.2 Inference

Once the graphical model has been defined, queries can be performed on the model, known as probabilistic inference. GrMPy offers exact inference, in the form of the junction Tree Algorithm (Jensen and Jensen, 1994), and approximate inference, in the form of the Loopy Belief Propagation algorithm (Weiss, 1997), for both Bayesian Networks and Markov random fields. There are two main inference problems relating to graphical models. Firstly, users may wish to find the marginal probability distribution for one node, or a set of nodes in a graph. Secondly, users may want to find the maximum-a-priori configuration of all the variables in the model that corresponds to the maximum probability of the models joint probability distribution. In both cases users may also want to specify observed evidence for certain nodes before performing inference. All of these features are implemented in the library, and can be performed using either exact or approximate inference algorithms.

2.3 Learning

GrMPy offers the following two types of parameter learning for both Markov random fields and Bayesian networks. Maximum-likelihood-estimation (MLE) (Heckerman, 1995) parameter learning can be performed using a set of fully-observed sample data. The generalized Estimation-maximization (EM) algorithm (Dempster et al., 1977) can be used to learn parameters from partially observed sample data.

3. Development and related software

GrMPy has been developed as a cross-platform, open-source project using the freely available language Python. It also makes use of the NumPy (Jones et al., 2001-2008a) and SciPy (Jones et al., 2001-2008b) libraries that are both freely available from http://www.scipy.org. GrMPy includes a set of unit-tests to ensure future updates do not damage critical functionality. GrMPy also includes a full set of tutorial-like examples outlining the use of MLE and EM learning, as well as Exact and Approximate inference, on both Bayesian networks and Markov random fields. A more detailed example illustrating the use of GrMPy to perform image denoising is also included. GrMPy's algorithms and implementation are detailed in (Gouws, 2009).

Although there are several existing software packages for implementing graphical models, few of them are both freely available and can support both directed and undirected graphs. The most prominent software packages that share these traits with GrMPy are BNT, MIM, WinMine. BNT was originally written by Dr. K. P. Murphy, and later released as an open-source project. Though the development of BNT was abandoned in October 2007, it is still available at http://www.cs.ubc.ca/ murphyk/Software/BNT/bnt.html. MIM is an application aimed at modeling data with graphical models, and is freely available at http://www.hypergraph.dk/. WinMine is an application aimed at determining statistical models from data, and is freely available at http://research.microsoft.com/en-us/um/people/dmax/winmine/tooldoc.htm. WinMine has not been updated since 2006, and development on the project has been abandoned. MIM was last updated in 2008, and is scheduled for one last update before halting development. A tabular comparison of GrMPy and these packages is shown in Table 1.

	GrMPy	**BNT**	**MIM**	**WinMine**
Source available	Yes	Yes	No	No
Language	Python	Matlab/C	N/A	N/A
Supported Variables	D	D/C	D/C	D/C
Platform	Platform-independent	Windows	Windows	Windows
Inference	E/A	E/A	E/A	None
Parameter Learning	Yes	Yes	Yes	Yes
Structure Learning	No	Yes	Yes	Yes
Development	Ongoing	Halted	Halted	Temp. ongoing

Table 1: A comparison of GrMPy with other freely available software packages which also support both directed and undirected graphical models. In the Supported Variables field: C - Continuous and D - Discrete, and in the Inference field: E - Exact and A - Approximate.

As shown in Table 1, GrMPy is the only library which is both open-source and is written in a freely available language. It is also the only package that is available on an open-source platform, such as Linux. Although GrMPy does not currently support structure learning from data, or continuous variables, it is the only package which is still actively under development, and will at some point support these features.

4. Future work

Future work includes adding extensive documentation, adding support for continuous conditional probability distributions and clique potentials, as well as adding support for creation, learning and inference of dynamic Bayesian networks. A further goal is to incorporate a graph-structure learning algorithm. Since GrMPy has been implemented in Python, which is an interpreter based language, its execution is much slower than that of code written in a compiler based language such as C, and therefore an ongoing goal is to improve the codes speed of execution as well decreasing its memory usage.

5. Conclusion

We have developed a library of classes and functions to easily perform learning and probabilistic inference on graphical models. As an open-source project we hope that GrMPy will continue to increase in functionality, and become widely used for graphical model based research and applications.

Acknowledgments

We would like to acknowledge support for this project from the National Research Foundation.

References

C. M. Bishop. *Pattern recognition and Machine Learning*. Springer, 2006.

A. P. Dempster, N. M. Laird, and N. M. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

A Gouws. A python implementation of graphical models. Master's thesis, University of Stellenbosch, 2009.

D. Heckerman. A tutorial on learning bayesian networks. Technical report, Communications of the ACM, 1995.

Finn V. Jensen and Frank Jensen. Optimal junction trees. In *In UAI*, pages 360–366. Morgan Kaufmann, 1994.

E. Jones, T Oliphant, P Peterson, et al. NumPy: Open source numerical tools for Python, 2001-2008a. URL `http://www.numpy.org/`.

E. Jones, T Oliphant, P Peterson, et al. SciPy: Open source scientific tools for Python, 2001-2008b. URL `http://www.scipy.org/`.

Y Weiss. Belief propagation and revision in networks with loops. Technical report, MIT, 1997.