ParaGnosis

ParaGnosis is a C++ weighted model counting toolset for linux. Its implementation is based on [1,2,3,4]. We have also added a significant number of Bayesian networks to play with (under ./data/net)

The toolset consists of the following command-line tools:

- bn-to-cnf: a c++ tool to create Conjunctive Normal Form (CNF) encodings from a Bayesian network.
- bnc: a c/c++ Bayesian Network Compiler for multiple target representations.
- bnmc : a c++ Bayesian Network Model Counter.
- pg: a ParaGnosis user friendly interface to the tools above, written in Python.

The currently supported target languages are:

- Weighted Positive Binary Decision Diagrams (WPBDD)
- Weighted Positive Multi-Valued Decision Diagrams (WPMDD)
- Tree-driven Weighted Positive Multi-valued Decision Diagrams (TD-WPMDD)

Installation (Ubuntu 18.04+)

Install requirements with apt:

```
> sudo apt-get install -y libboost-all-dev python3 \
   python-setuptools make cmake gcc g++ libgmp-dev \
   libgsl-dev libreadline-dev make cmake evince
```

Install latest pip (the python package installer):

```
> sudo python3 -m pip install --upgrade pip
```

Install 'sympy' with pip

```
> sudo pip3 install sympy
```

To build all tools in the toolset, type:

```
> make
```

Binaries will be installed in the <path/to/source>/bin directory, and the pg script will be available system wide.

(Re-)configure pg

The make process automatically configures pg, so this step is optional or if the configuration has failed. In order to let the pg script know where the toolset is located, we can run pg commands with pg --source-dir=<path/to/source> ..., or adjust the following in pg 's configuration file pg--pg

```
location = <path/to/source>
```

To test if the installation is successful, we can give the following a try. Open a terminal at any location and type:

```
> pg encode asia
or:
> pg --source-dir=<path/to/source> encode asia
```

This should produce encoding statistics for the asia network.

Usage

All available commands can be found through pg --help, pg compile --help, pg encode --help and pg inference --help. For comprehensive examples, please see the demo.

References

- [1] G.H. Dal, A.W. Laarman, A. Hommerso and P.J.F. Lucas, "A Compositional Approach to Probabilistic Knowledge Compilation", in International Journal of Approximate Reasoning, vol 138:38-66, 2021.
- [2] G.H. Dal, A.W. Laarman and P.J.F. Lucas, "Parallel Probabilistic Inference by Weighted Model Counting", in Proceeding of the International Conference on Probabilistic Graphical Models, PMLR, vol 72:97-108, 2018.
- [3] G.H. Dal, S. Michels and P.J.F. Lucas, "Reducing the Cost of Probabilistic Knowledge Compilation", in Proceedings of Machine Learning Research, volume 73, pages 41-152, 2017.
- [4] G.H. Dal and P.J.F. Lucas, "Weighted Positive Binary Decision Diagrams for Exact Probabilistic Inference", in Journal of Approximate Reasoning, volume 90, pages 411-432, 2017.