`graph_energy` Overview

You Chen, Torin Stetina, Andrew Wildman

May 3, 2018

Summary

The goal of this project is twofold:

1. Develop a software framework that predicts the energy distribution of a group of microstates from their topological parameters
2. Validate this framework on two sets of example data, specifically an ideal ice model and a lattice gas model.

Approach

The central prediction capabilities of `graph_energy` are provided through a machine learning algorithm. Because the relationship between the graph connections and the energy is expected to be non-linear, (e.g. for a hydrogen bond graph, four edges on a node will likely be lower in energy than three or five) a simple, feed forward neural net has been chosen as the prediction algorithm. Neural networks have a large meta-parameter space, so an optimization of these meta-parameters will be built into the software itself.

Use Cases

The `graph_energy` package has four main use cases:

1. Train a neural net to reproduce the energy distribution of a set of microstates, given graph topological parameters that define the microstate, temperature, and population of the microstate at that temperature.
2. Evaluate the accuracy of nets trained on different sets of topological parameters at capturing the energy distribution.
3. Use previously trained neural nets to predict the energy distribution of microstates whose energy distribution was previously unknown.
4. Visualize the predicted energy distribution and, if available, contrast it with the known energy distribution.

Data Description

The raw data comes from the Clark Group's Shannon entropy program. It consists of a set of `parameters.dat` files, each of which is generated at a given temperature and contain the topological parameter values of each microstate, as well as the energy of each subgraph. The population of the microstates are given by the `output.dat` files.

An example of an parameters.dat file is given in figure 1.

Software Design

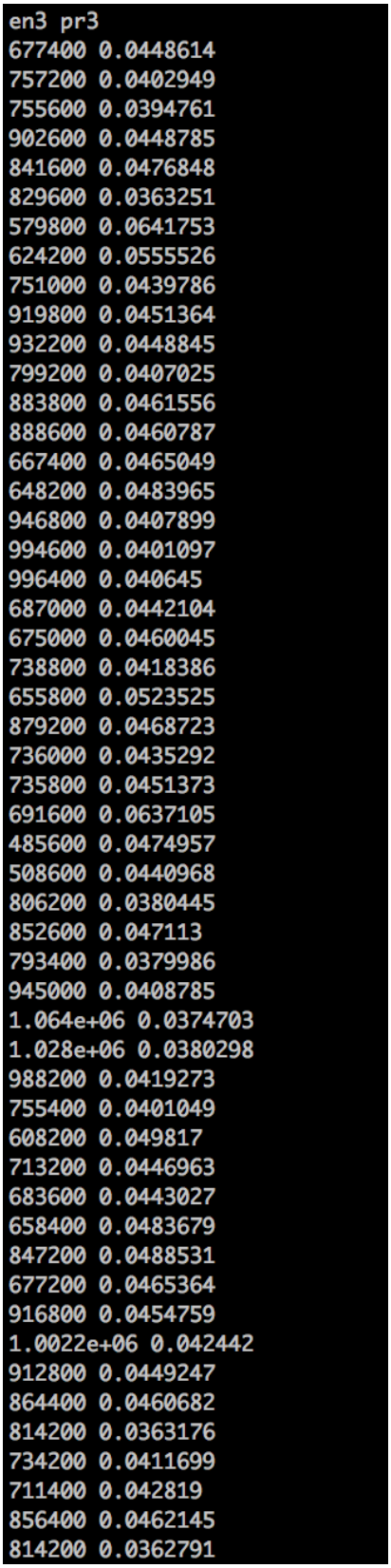
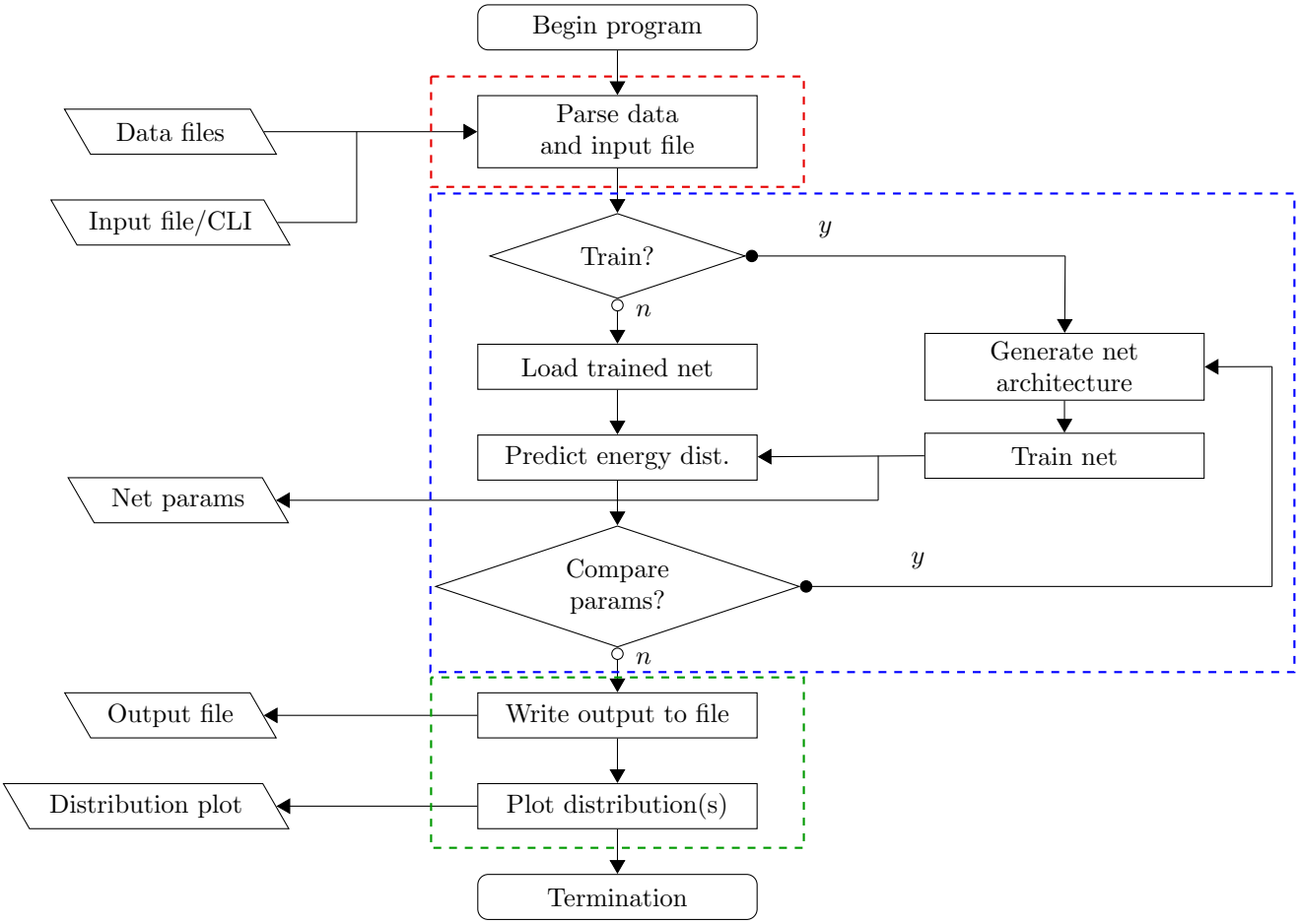


Figure 1: An example output file from the Clark Group's Shannon entropy program. `en3` is the 3rd order subgraph energy, and `pr3` is the 3rd order page rank.

Software flow



The rhombuses represent inputs or outputs to the program, rectangles represent processes, diamonds represent tests, and rounded boxes delimit the bounds of the program. The dashed boxes represent software components/classes, which will be described in the next section.

Component specification

Parser (red)

The parser takes the `parameters.dat` files as well as the input file, then it converts them into the expected inputs for the parser.

Input:

- `/path/to/parameters.dat`
- Job type: `train` or `predict`
- Meta parameter options for neural net

Outputs:

- Dataframe object from columns of `parameters.dat`
- Dictionary of meta parameters and job type for neural net

Exceptions:

- Handle wrong `.dat` file format

- Handle input file format errors

Predictor (blue)

The predictor takes the metaparameter options and data from the parser, then does the following:

1. Checks type of job (predict vs train, which type of training)
2. Loads appropriate architecture for type of job
3. If training, trains the net
4. Predicts based on the trained net
5. Repeats steps 2-4 as many times as job requires

Inputs:

- Dataframe whose rows are the subgraphs and whose columns are the topological parameters (and true energy)
- Dictionary of neural net meta-parameters

Outputs:

- Dataframe whose rows are the subgraphs and whose columns are the topological parameters as well as the predicted (and true) energy
- (Optional) External file written with saved net weights and architecture

Exceptions:

- Empty Dataframe
- Unsupported topological parameter
- (If train) No true energies
- (If predict) No matching pre-trained net for set of params

Visualizer (green)

Inputs:

- Dataframe with predicted energy (and true energy)
- Boolean for showing true energy

Outputs:

- Plot of energy distribution
- Text file containing dataframe

Exceptions

- Empty dataframe
- No predicted energy
- (If compare) No true energy