Primula Guide: Water Pollution Example

In this example, we demonstrate how a Graph Neural Network (GNN) can be integrated into a Relational Bayesian Network (RBN) to model water pollution in a sub-basin using Primula. More info about how to load other models in Primula can be found in the main documentation.

Primula and PyTorch Integration

Primula supports the use of external GNN models implemented in PyTorch[1]. In this guide, we use **PyTorch Geometric** (PyG)[2], a popular extension library for GNNs in Python. To bridge the communication between Primula and PyTorch, we use the **JEP** library[3]. JEP enables the execution of CPython code directly within the JVM, allowing seamless integration between Java and Python components.

To run this example, ensure you have the following components installed:

- The latest version of **Primula**
- **JEP** (installed via `pip install jep`)
- **PyTorch Geometric** (installed in the same Python environment as JEP)

After installing the necessary components, configure your environment as follows:

1. Identify the path to the Python environment where JEP and PyTorch Geometric are installed.
2. Set the `PYTHONHOME` environment variable to this path. For example:

   ```
   export PYTHONHOME=/path/to/your/python
   ```

3. Ensure that the selected Python environment includes both PyTorch and JEP.

[1] `https://pytorch.org/`
[2] `https://pytorch-geometric.readthedocs.io/en/latest/`
[3] `https://github.com/ninia/jep`

Water Pollution Model: the simple case

Before loading the model and data into Primula, we have to make sure that the path in the `.rbn` file have the right path to the model, and that the model is correctly loaded with the previously learned weights. Open the `config_model.py` file, and change the path of the `load_model()` function where we load the model's weights. The weights can be found in the same folder where this file is (`model_l21e3_bn.pt`). Open the `water_pollution_model.rbn` file and in the folder and change the path in the *Pollution* definition to the path of the file `config_model.py`.

Probabilistic Inference

Now, we can first load the data `simple_subbasin.rdef`, and then the model `water_pollution_model.rbn` into Primula. If we now open *Bavaria* from the main console (Modules:Bavaria), we can see the current instantiaded model, figure 1.

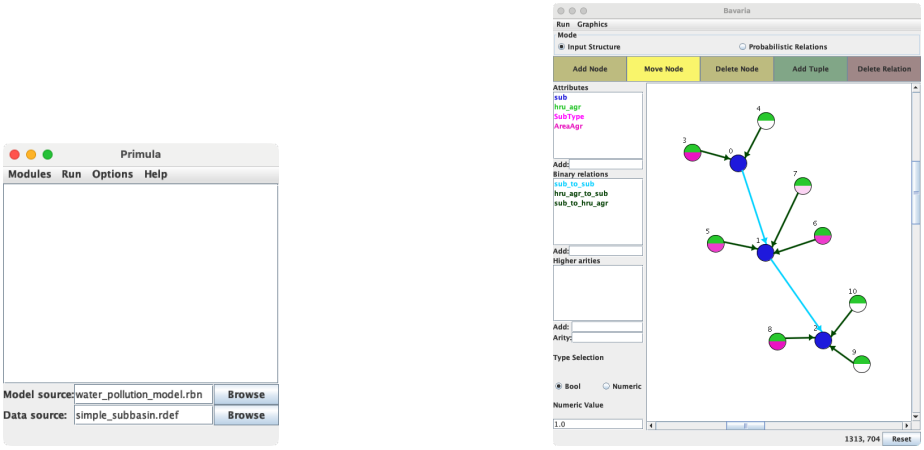

Figure 1: Data and Model loaded, with Bavaria open

In this simple model, we have three *Subbasin* nodes, each conneted with a downstream relation `sub_to_sub`. Each subbasin node, have some *Agricolture* nodes conneted with a specific relation `agr_to_sub`. In the *Agricolture* nodes we can grown some crops (soybean `SOYB`, corn `CORN`, pasture `PAST`, or a mixture of corn and soybean `COSY`). Each crop impact diffrently on the *Pollution* level of each subbasin node.

Probabilistic Inference with SamIam (requires SamIam to be installed): You can construct the corresponding Bayesian Network that represents the model and the specific instance, and then perform probabilistic inference using SamIam. To do this, click on **Run** → **Construct Bayesian Network** from the main Primula console. Once the network is constructed, Primula will

automatically launch SamIam. See Figure 2 for a visual reference.

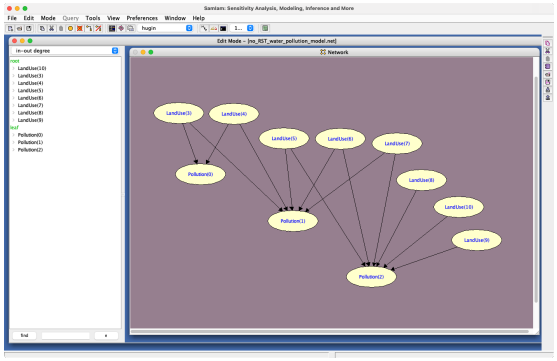

Figure 2: SamIam with constructed Bayesian network

MAP Inference with Primula: To perform MAP inference in Primula, start by selecting **Modules** → **Inference Module** from the Primula console.

Next, go to the *Evidence* tab. From the *Relations* list, select `Pollution`, and from the *Values* list, choose `LOW`. Assign this value to all subbasin nodes by selecting `[sub*]` from the *Element names* list. This will automatically apply `LOW` as evidence to all subbasins.

Then, switch to the *Query* tab. To query all `LandUse` relations, select `LandUse` and then choose `[hru_agr*]` from the *Element names* list. All queried atoms will now be displayed.

Finally, go to the *MAP* tab and click the *Start* button. Primula will attempt to find the most likely assignment that maximizes the posterior probability given the current evidence (i.e., all subbasins set to `LOW`). See Figure 3 for an illustration of the Inference Module.

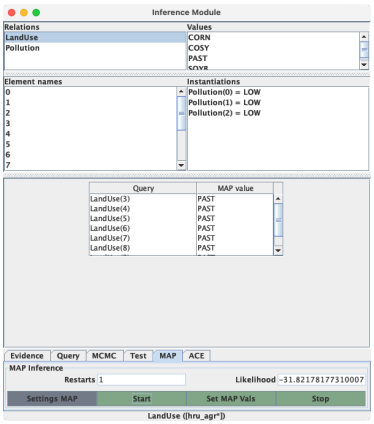

Figure 3: MAP inference in Primula