Additional Comments

Running our code

To run our code, please copy all .py files into a directory. Then inside that directory, run:

```
python main.py batch_id=1 name=sparse_sampling move_limit=100 root_path="./"
```

Dependencies: Python 3, scikit-learn 0.19.1

More about the script parameters:

- batch_id and name are unique identifiers used for batch jobs on flux
- move_limit sets the training time for the algorithm
- root_path is the directory where the final models are saved
- the parameters above will run the sparse sampling algorithm
- to run sparse sampling with Thompson sampling, add the parameters:
  - prune=T
  - ts_hyper_param=25
  - where ts_hyper_param determines how *quickly* the additional exploration condition on sparse sampling is removed (we suggest ts_hyper_param = (move_limit * 0.25)
- to run sparse sampling with episodic reset and bootstrapping, add the parameters:
  - bootstrap=T
  - ep_len=1
  - where ep_len determines how many games make one training episode

The sparse sampling algorithm (Kearns et al., 2001) is implementing in bayesSparse.py. The file gpPosterior.py fits the internal belief-based models (for belief-based positions of terminal states). The mdpSimulator.py allows the agent to switch between belief-based models of the MDP and the real MDP. The Beta/Dirichlet posteriors using for Thompson Sampling are defined in thompsonSampling.py.

Contributions

All coding and reporting work was done evenly by Ray Lee and Aman Taxali.