PPAML Challenge Problem 4: The Small Problems Collection

Tom Dietterich, Last Updated March 3, 2015

The goal of the “Small Problems Collection” is to create a set of problems that span important dimensions of the space of probabilistic programs in terms of both program formulation and probabilistic inference.

We hope that this set of problems can help the PPAML teams identify important tradeoffs in the design and implementation of Probabilistic Programming Systems. (Note that this is an important change in direction from the previous focus on benchmarking of PPS systems.

Problem Dimensions

At the Kickoff meeting, we described several problem dimensions. Some of those related to application domains. For the Small Problems Collection, we want to ignore those and focus instead on the dimensions that capture abstract problem structure. These include the following:

1. Data types: Continuous, Discrete, Mixed
2. Data structures: Atoms, Vectors, Relations, Grammars, Graphs
3. Model structure: Directed vs. Undirected, Parametric vs. Non-parametric, Fixed vs. Variable number of variables, Presence of latent variables
4. Query type: MAP, Marginal MAP, Expectations, Posterior Distribution, Posterior Summary (e.g., moments), Anomalies
5. Query timing: One-shot (data and query are presented together), Online (fixed query, but data arrive incrementally, so the query results need to be updated incrementally), Amortized (fixed data, but multiple, related queries arrive incrementally, and inference costs can be amortized across them), Online Amortized (both the data and the queries arrive incrementally).

Our focus is primarily on taking a given model plus some observations and answering queries (i.e., as opposed to having a separate learning phase).

In this setting, we will say that latent variables are present if there are variables that are neither observed nor queried (and hence, must be marginalized out). This means that the “latent variables” dimension becomes a property of the observations + query rather than of the model, per se.

List of Small Problems

There are ten problems in the collection of small problems. Note that two of the problems are expressiveness challenges.

1. Bayesian Linear Regression
2. Medical Diagnosis
3. Discrete time Hidden Markov Model
4. Latent Dirichlet Allocation under Hierarchical Dirichlet Prior
5. Conditional Queries for Probabilistic Context-Free Grammars
6. Network Analysis
7. Friends and Smokers (Undirected Graphical Model)
8. Seismic Event Detection
9. Recursive Reasoning: Scalar Implicature
10. Lifted Inference

Individual Problem Details

Each small problem has a corresponding description document located in the ppaml-cp4 Git repository. https://github.com/GaloisInc/ppaml-cp4

Metrics and Submission Guidelines

For each problem, there are one or more accuracy metrics defined. We would like to measure the values of these metrics as a function of CPU time so that we can see the performance profile of the inference procedures. For sampling-based methods, this curve may have interesting structure. For a direct method, it may be a simple step function.

For each inference problem, query, and metric, you should produce a CSV file with the name "problem-*p*-query-*q*-metric-*m*.csv". The first column is the execution time in milliseconds. The second column is the value of the metric. For some of the problems, additional columns for the metric may be necessary. For example, in problem 3, the Discrete-time Hidden Markov Model, queries 2 and 3 need more columns.

You should choose your reporting interval (elapsed time between rows) to capture the behavior of your method in less than 1000 rows. You may wish to progressively increase the sampling interval (e.g., quadratically or exponentially) to provide a more informative profile.

Example CSV output files are provided in the 'sampleoutput' directory.

Note that there are different instructions for problem 8 and problem 10, found in the problem's documentation.

Expressiveness Challenge Problems

Problems 6 and 9 are expressiveness challenges. The primary requirement is to demonstrate a probabilistic program and show that it runs and computes the right answer. Teams should submit their source code as file “problem-*p*-solution.tar”. Teams may optionally produce performance profiles for a metric of their choice as well. Please define the metric in a file names “problem-*p*-query-*q*-metric.pdf”.

Submission of Source Code

In addition, you should submit the source code for each of your challenge problem solutions as a “tar” archive with the name “problem-*p*-query-*q*-metric-*m*-solution.tar”. If all metrics share the same solution, you can just provide a single file for that query. If all queries share the same solution, you can just provide a single file for the problem. We expect to evaluate the source program with respect to such measures as compactness, elegance, comprehensibility, and extensibility.