Exercise: comparing sampling methods with Belief Propagation

Joris Mooij (j.mooij@science.ru.nl)

June 7, 2007

0.1 Pairwise, binary graphical models

Consider binary random variables x_i for $i = 1, \dots, N$. Each variable x_i has two possible values: -1 and $+1$. We consider the following probability distribution over $x = (x_1, \dots, x_N)$:

$$P(x) = \frac{1}{Z} \exp\left(\sum_{i=1}^{N} \theta_i x_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} J_{ij} x_i x_j\right)$$

where the partition sum Z ensures that the probability is normalized:

$$Z = \sum_{x \in \{-1,+1\}^N} \exp\left(\sum_{i=1}^{N} \theta_i x_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} J_{ij} x_i x_j\right).$$

The probability distribution is parameterized by the N-dimensional (real-valued) vector $\theta = (\theta_1, \dots, \theta_N)$ and the real-valued symmetric matrix $J = J^T$, for which we assume $J_{ii} = 0$ for all $i = 1, \dots, N$.

Because each variable x_i is binary, its marginal can be conveniently parameterized by a single real value, the *magnetization* (or expected value)

$$M_i := P(x_i = +1) - P(x_i = -1)$$

for $i = 1, \dots, N$.

0.2 Ising grid

Take $n = 10$ and $N = n^2$. Arrange the variables on a rectangular grid and let $J_{ij} = 0$ if x_i and x_j are not nearest neighbours on the grid. This means that there are at most four non-zero entries in each row and column of J (less than four for variables at the borders of the grid).

Choose these non-zero entries of J to be i.i.d. according to a $\mathcal{N}(0, \beta)$ distribution for some value of β (respecting the symmetry constraint $J_{ij} = J_{ji}$). Choose the entries of θ to be i.i.d. according to a $\mathcal{N}(0, \Theta)$ distribution for some value of Θ.

0.3 Tasks

1. Implement a Metropolis-Hastings sampler for a binary pairwise graphical model, specified by a pair (J, θ). For the transitions, you could take "single spin flips", i.e., changing the state of a single variable x_i to $-x_i$.

2. Implement a Gibbs sampler for a binary pairwise graphical model specified by a pair (J, θ). Note that

$$P(x_i \mid x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_N) = P(x_i \mid x_{\partial i})$$

   where $\partial i := \{j \in \{1, \dots, N\} : J_{ij} \neq 0\}$.

3. For an Ising grid of size $n = 10$, use both samplers to obtain approximate magnetizations $m_i \approx M_i$ for all $i = 1, \dots, n^2$. Compare the results with the exact magnetizations and with the Belief Propagation (BP) approximations. You can use the library libDAI to calculate the exact and BP magnetizations (see below for detailed instructions). Do this for several values of β and Θ (e.g., ranging from small values such as 0.1 to large values such as 10) and plot the results. How much samples do you need to obtain similar accuracy as BP? What can you say about the relative performance of the samplers? Do your conclusions depend on the value of β and/or Θ?

0.4 Instructions for libDAI

The free/open source library libDAI of the author of this exercise contains an implementation of Belief Propagation and of an exact JunctionTree method, amongst other approximate methods. Needed: a recent C++ compiler (e.g., gcc version 3.4 or higher). Instructions for use (assuming UNIX):

1. Download from `http://www.mbfys.ru.nl/~jorism/libDAI`:
   `cn01> wget http://www.mbfys.ru.nl/~jorism/libDAI/libDAI-0.2.tar.gz`

2. Extract the tarball:
   `cn01> tar zxvf libDAI-0.2.tar.gz`

3. Enter the directory:
   `cn01> cd libDAI-0.2`

4. Replace the existing file example.cpp by the one at `http://www.mbfys.ru.nl/~jorism/libDAI/example.cpp`: and download the MatLab script `http://www.mbfys.ru.nl/~jorism/libDAI/jth2fg.m`:
   ```
   cn01> rm example.cpp
   cn01> wget http://www.mbfys.ru.nl/~jorism/libDAI/example.cpp
   cn01> wget http://www.mbfys.ru.nl/~jorism/libDAI/jth2fg.m
   ```

5. To speed up linking, edit the `Makefile` and remove the `-g` flag from the `CCFLAGS` macro. Then, build the example:
   `cn01> make example`

6. Test whether the build was successful:
   `cn01> ./example tests/testfast.fg`
   This should output 16 lines of three columns each, the first column being the variable labels i, the second being the exact magnetizations M_i and the third being the BP approximations m_i of the magnetizations. For your convenience, the header line is written to stderr instead of stdout, so you can write the data to a file:
   `cn01> ./example tests/testfast.fg > testfast.dat`

To convert a pair (J, θ) to a .fg file that can be read by libDAI, you can use the MatLab script `jth2fg.m`.

For questions, you can email me (`j.mooij@science.ru.nl`).