ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

School of Computer and Communication Sciences

Handout 8 Statistical Physics for Communication and Computer Science
Homework 8 April 14, 2011, INR 113 - 9:15-11:00

You will implement Belief Propagation (BP) for K-SAT (say $K = 3$ and $K = 4$) in a form presented in *Information, Computation and Physics, by A. Montanari and M. Mézard, pp. 468-473*. There are three aspects to the problem. The first one is to find a convenient parametrization of the BP messages. The second is to investigate numerically the convergence of BP as a function of α (the clause density). The third is to implement a decimation algorithm that finds satisfying assignments for α not too large.

Problem 1 (Belief Propagation Equations for K-SAT). We recall the BP equations in the form

$$\mu_{i\to a}(x_i) \cong \prod_{b\in\partial i\backslash a} \hat{\mu}_{b\to i}(x_i), \qquad \hat{\mu}_{a\to i}(x_i) \cong \sum_{\sim x_i} f_a(\{x_k, k\in\partial a\}) \prod_{k\in\partial a\backslash i} \mu_{k\to a}(x_k)$$

Here the messages are normalized probability distributions and $\cong$ means that the right hand side has to be divided by a normalization factor such that $\mu_{i\to a}(0) + \mu_{i\to a}(1) = 1$ and $\hat{\mu}_{a\to i}(0) + \hat{\mu}_{a\to i}(1) = 1$. Moreover ∂a is the set of neighbors of a and ∂i is the set of neighbors of i. The function f_a is the indicator function on assignments that satisfy clause a.

The first step is to transform these equations in a more convenient form. For this we need the following notation. Let $J_{ia} = 1$ when variable i appears negated (as $\bar{x}_i$) in clause a and $J_{ia} = 0$ when variable i appears unnegated (as x_i) in clause a. For a fixed edge ia the value of J_{ia} is fixed. We decompose $\partial i\backslash a$ in two disjoint sets

$$\partial i\backslash a = S_{ia} \cup U_{ia}$$

where $b \in S_{ia}$ iff $J_{ib} = J_{ia}$ and $b \in U_{ia}$ iff $J_{ib} \neq J_{ia}$. Finally we parametrize the messages as $\mu_{i\to a}(J_{ia}) = \zeta_{i\to a}$ and $\hat{\mu}_{a\to i}(J_{ia}) = \hat{\zeta}_{a\to i}$.

The BP equations are equivalent to

$$\zeta_{i\to a} = \frac{[\prod_{b\in S_{ia}} \hat{\zeta}_{b\to i}][\prod_{b\in U_{ia}}(1-\hat{\zeta}_{b\to i})}{[\prod_{b\in S_{ia}} \hat{\zeta}_{b\to i}][\prod_{b\in U_{ia}}(1-\hat{\zeta}_{b\to i})] + [\prod_{b\in U_{ia}} \hat{\zeta}_{b\to i}][\prod_{b\in S_{ia}}(1-\hat{\zeta}_{b\to i})]}$$

$$\hat{\zeta}_{a\to i} = \frac{1-\prod_{j\in\partial a\backslash i}\zeta_{j\to a}}{2-\prod_{j\in\partial a\backslash i}\zeta_{j\to a}}$$

Question : go through the derivation, especially if this was not done in detail during class.

Problem 2 (Implementation of BP). You will implement BP according to the flooding (or parallel) schedule. initialize the messages uniformly randomly in $[0, 1]$. One iteration means that you send messages from nodes to clauses and back from clauses to variables. Define the following "convergence criterion": declare that the messages have "converged" if there is an iteration number (time) $t_{\rm conv}(\delta)$ such that no messages changes by more than δ at $t_{\rm conv}(\delta)$ (take the smallest such time).

Perform the following experiment. Take 100 K-SAT instances of length say $N = 5000$ and 10000 variables and for each instance implement BP as explained above with $\delta = 10^{-2}$. If the iterations do not converge stop them at a large time say $t_{\max} \approx 1000$. When they converge, they should do so in a shorter time $t_{\mathrm{conv}}(\delta) < t_{\max}$ that does not change much with N.

Plot as a function of α the empirical probability that the iterations converge. You should see that this probability is large for $\alpha < \alpha_{\mathrm{BP}}$ and drops abruptly around some threshold α_{BP}. For $K = 3$, $\alpha_{\mathrm{BP}} \approx 3.85$ and $K = 4$, $\alpha_{\mathrm{BP}} \approx 10.3$.

Problem 3 (BP guided decimation). Now you will implement the following algorithm for finding SAT assignments. It uses the above BP procedure as a guide to take decisions on how to fix values for the variables. Once a variable has been fixed the K-SAT formula is suitably reduced - this step is called "decimation" - and BP is run again.

- Initialize with a K-SAT formula $\mathcal{F}$ of length N.
- For n= 1, ..., N do:
  - Run BP on an instance, as in the previous exercise (with the same convergence criterion).
  - If BP does not converge, return "assignment not found" and exit.
  - If BP converges, for each variable j compute its bias (express it in terms of $\hat{z}eta$ variables!)

$$\pi_j = \mu_j(1) - \mu_j(0) = \frac{\prod_{a\in\partial j} \mu_{a\to j}(1) - \prod_{a\in\partial j} \mu_{a\to j}(0)}{\prod_{a\in\partial j} \mu_{a\to j}(1) + \prod_{a\in\partial j} \mu_{a\to j}(0)}$$

  - Pick a variable $j(n)$ that has the largest absolute bias $|\pi_{j(n)}|$.
  - If $\pi_{j(n)} \geq 0$ fix $x_{j(n)} = 1$. Otherwise fix $x_{j(n)} = -1$.
  - Replace $\mathcal{F}$ by the K-SAT formula obtained by decimating variable $j(n)$.
- End-For
- Return all fixed variables.

Give for several values of α, the empirical success probability of this algorithm when tested over 100 instances. Compare this empirical success probability with the empirical convergence probability of the previous exercise. You should observe that $K = 3$ and $K = 4$ do not behave on the same way. Try to find an approximate threshold α_t beyond which the algorithm does not find SAT assignments.