Advanced Probabilistic Machine Learning and Applications

Alessandro Lonardi and Caterina De Bacco

June 8, 2021

1 Tutorial 7: Bethe approximation and BP

1.1 Exercise 1: graph coloring and BP

Coloring is a classical problem of graph theory. Given a (unweighted, undirected) graph $G(V, E)$ a coloring $M \subseteq V$ is an assignment of labels, called colors, to the vertices of a graph such that no two adjacent vertices share the same color.

The problem can be tackled using a set of spins $\{s_i\}$ taking values in the set $s_i \in \{1, ..., q\}$ for each $i = 1, ..., N = |V|$. In particular, we can represent the spins configurations considering the relaxed probability distribution (derived in tutorial 6):

$$P(\mathbf{s}) = \frac{1}{Z} \prod_{(ij) \in \tilde{F}} \psi_{ij}\left(s_i, s_j\right) = \frac{1}{Z} \prod_{(ij) \in E} \mathrm{e}^{-\beta \mathbb{I}(s_i = s_j)} \quad ,$$

with $\tilde{F}$ set of factor nodes representing the interaction between spins in the coloring problem.

In this tutorial, we want to complete the following tasks:

Part I

1. Using BP to model marginals of the coloring assignment, denote as:
   - $\nu_{s_i}^{(ij)\to i}$ the messages from function node (ij) to variable node i.
   - $\chi_{s_i}^{i\to(ij)}$ the message from variable node i to function node (ij).

   Note that they are both functions of the state s_i of the variable node i.
   Write BP equations for this model.
2. Find a fixed point of these equations. (PS: recall that there might be more than one fixed point)

   Hint: what would a random guess do?
3. Write the equation for the one-point marginal $P(s_i)$ and the two-point marginal $P(s_i, s_j)$ obtained from BP.

Part II

Now, we want to code belief propagation for Erdös-Rényi graphs coloring for $\beta \to \infty$.

1. Initialize BP close to be uniform fixed point, i.e. $1/q + \varepsilon_s^{j\to i}$ and iterate the equations until convergence. Define converge as the time when the

   $$\frac{1}{2qM} \sum_{(ij)\in E} \sum_{s} \left| \left(\chi_s^{i\to(ij)}(t+1) - \chi_s^{i\to(ij)}(t) \right) \right| < \tilde{\varepsilon}$$

   with suitably chosen small $\tilde{\varepsilon}$.

2. Check how the behavior depends on the order of update, i.e. compare what happens if you update all messages at once or randomly one by one.
3. or parameters where the update converges, plot the convergence time as a function of the average degree c. Do this on as large graphs as is feasible with your code.
4. Assign one color to each node at convergence, based on the argmax of the one-point marginals. Count how many violations you get over $N_{\mathrm{real}} = 100$ random initializations of the graph and plot them as a function of c.
5. Check how the behavior depends on the initialization. What if initial messages are random? What if they are all points towards the first color?