# Probabilistic Graphical Models, Spring 2012

## Problem Set 4: Approximate MAP inference
**Due: Friday, March 30, 2012 at 5pm (electronically)**

---

For the following question, you may use the programming language of your choice. You are allowed to use basic graph packages (e.g., for representing and working with directed or undirected graphs), but are **not** permitted to use any graphical model or probabilistic inference packages.

**E-mail a PDF of your full assignment, including all code written, to Chris.**

In Lectures 3 and 7 we discussed how to use **dual decomposition** to perform approximate MAP inference in discrete-valued graphical models. This approach allows one to obtain an upper bound on the value of the MAP assignment, and in many real-world problems actually succeeds in finding the exact MAP assignment, with a certificate of optimality.

**Implement the max-product linear programming (MPLP) algorithm** from class (also described in Section 1.5 of [1]). This algorithm is analogous to the max-sum belief propagation algorithm that you implemented in PS 3, and makes use of the log of the potential functions. Use as the stopping criterion a difference in the dual objective of .0002 (i.e., terminate when $L(\delta^{t+1} - L(\delta^t) \leq 0.0002$, where each iteration $t \to t + 1$ corresponds to one pass through all factors). Your code should perform local decoding (i.e., finding the assignment $\mathbf{x}$ given by $x_i \in \arg\max_{\hat{x}_i} \tilde{\theta}_i^\delta(\hat{x}_i)$) at every iteration (not just in the last one), printing both the value of the integer solution found (that is, $\theta(\mathbf{x})$), the value of the dual objective ($L(\delta)$), and the difference of these (known as the *integrality gap*). You should also keep track of the best integer assignment found so far; at termination, this will be your approximate MAP solution.

Although MPLP can be used with any factor graph, in this assignment you are allowed to special-case your code for pairwise MRFs (i.e., the factors will only be on single-node variables and pairs of variables). Thus, the set $F$ from Figure 1.4 of [1] will correspond precisely to the set of edge potentials. Use a schedule (i.e., an ordering of the messages) corresponding to the order of the edge potentials provided in the original UAI file. **Do not** special-case your code for binary variables – problem 2 below involves inference with non-binary variables.

1. MPLP can be shown to always be exact (i.e., have zero integrality gap once the algorithm has converged) for any tree-structured graphical model.

   **Run your algorithm on the two object detection CRFs given in PS 3**, and attach the output (i.e. the per-iteration values of the integer solution found, dual objective, and integrality gap). You should confirm that you found the same MAP assignment as you did using the max-sum belief propagation algorithm in PS 3. How many iterations did MPLP take to find the MAP assignment?

   (Note that the running time of belief propagation is roughly equivalent to one iteration of that of MPLP.)

2. In this question, we apply MPLP to **solve a key task in structural biology**. The side-chain placement problem involves finding the three-dimensional configuration of rotamers given the backbone structure of a protein. This problem can be formulated as finding the MAP configuration of a pairwise model [4, 3].
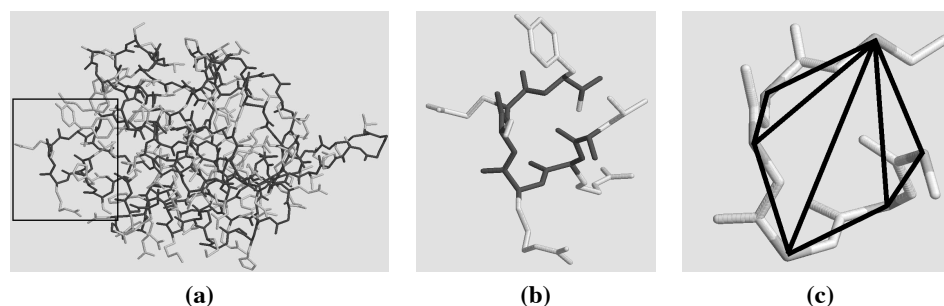
**(a)**          **(b)**          **(c)**

Figure 1: (a) Cow actin binding protein (PDB code "1pne"). (b) A closer view of its 6 C-terminal residues. Given the protein backbone (black) and the amino acid sequence, side-chain prediction is the problem of predicting the native side-chain conformation (gray). (c) Problem representation as a graphical model. Figure and caption from [3].

We have provided you with the CRFs corresponding to the side-chain placement problem for two different proteins, "2dri" and "1exm". These files end in "UAI.LG", denoting that they use the **logarithm** of the potential functions (we do not exponentiate because of potential numerical errors; this is what the algorithm needs anyway).

Run MPLP on both files, and attach the output. For "2dri", you will find that there is no integrality gap. At what iteration do you first find the MAP assignment via local decoding? For "1exm", after running MPLP to convergence, what is the value of the best assignment found, and what is the integrality gap?[1]

# References

[1] David Sontag, Amir Globerson, and Tommi Jaakkola. Introduction to dual decomposition for inference. In Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

[2] David Sontag, Talya Meltzer, Amir Globerson, Yair Weiss, and Tommi Jaakkola. Tightening LP relaxations for MAP using message-passing. In *24th Conference in Uncertainty in Artificial Intelligence*, pages 503–510. AUAI Press, 2008.

[3] Chen Yanover, Talya Meltzer, and Yair Weiss. Linear programming relaxations and belief propagation – an empirical study. *JMLR*, 7:1887–1907, 2006.

[4] Chen Yanover and Yair Weiss. Approximate inference and protein-folding. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1457–1464. MIT Press, Cambridge, MA, 2003.

---

[1]If instead of stopping we had attempted to *tighten* the LP relaxation, the resulting algorithm almost always finds the MAP assignment [2].