

# Lecture 18

## Generalized Belief Propagation and Free Energy Approximations

In this lecture we talked about graphical models and belief propagation algorithms. As an application, we discussed a generalized belief propagation algorithm for approximating the free energy of a protein structure.

### 1 Free Energy

A general definition for the *free energy* of a system is “the amount of energy which can be converted into work”. Although there are several types of free energy, the most widely used is probably the *Gibbs free energy*, which can be defined as the amount of thermodynamic energy which can be converted into work at constant temperature and pressure. Formally, we write:

$$G = H - T \cdot S = (E + P \cdot V) - T \cdot S$$

where  $G$ =Gibbs free energy,  $H$ =enthalpy (the heat content of the system),  $S$ =entropy (a measure of the degree of randomness of the system),  $E$ =internal energy,  $T$ =temperature,  $P$ =pressure and  $V$ =volume.

The change in the Gibbs free energy of a system is  $\Delta G = (\Delta E + P \cdot \Delta V) - T \cdot \Delta S$ , and since the change in volume is small for nearly all biochemical reactions we can write  $\Delta G = \Delta E - T \cdot \Delta S$ .

Free energy functions have been successfully used in protein structure prediction, fold recognition, homology modeling, protein design [1]. However, most free energy functions only model the internal energy  $E$  using inter- and intramolecular interactions terms (van der Waals, electrostatic, solvent, etc.). The entropy  $S$  is usually ignored because it is harder to compute, since it involves summing over an exponential number of terms. Another approach is to compute the free energy using statistical potentials derived from known protein structures (*e.g.* from PDB). Such methods have the advantage that the derived potentials encode both the entropy  $S$  and the internal energy  $E$ . But there are several disadvantages, the most important being the fact that the observed interactions are usually not independent [2]. In [3], Kamisetty *et al.* use a generalized belief propagation algorithm to compute an approximation of the free energy function which includes the entropy term.

### 2 Graphical Models

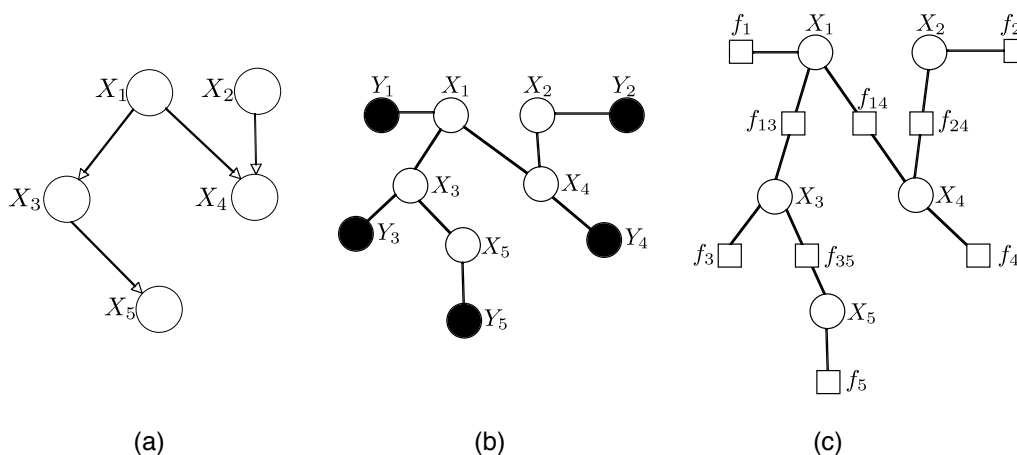
Graphical models are graphs that represent the dependencies among discrete random variables. The variables are usually represented as nodes in the graph and the dependencies as

edges. Examples of graphical models: bayesian networks (Figure 1a), Markov random fields (Figure 1b), factor graphs (Figure 1c), etc. We use capital letters for the random variables ( $X_1, X_2, \dots, X_n$ ) and small letters for the discrete values that the random variables can take ( $x_1, x_2, \dots, x_n$ ).

Let  $p(x_1, x_2, \dots, x_n)$  be the joint probability of all the variables in the model. The *marginal probability* of each variable can be written as:

$$p(x_i) = \sum_{x \setminus x_i} p(x) = \sum_{\{x_1, \dots, x_n\} \setminus \{x_i\}} p(x_1, x_2, \dots, x_n)$$

Computing these marginal probabilities is usually computationally expensive because it involves a sum over an exponential number of terms. Belief propagation algorithms try to overcome this problem by approximating the marginals  $p(x_i)$  with the *beliefs*  $b(x_i)$ .



**Figure 1:** (a) Bayesian network. (b) Pair-wise Markov random field. (c) Factor graph.

## 2.1 Bayesian Networks

In a Bayesian network (BN) the dependencies (conditional probabilities) are represented by arrows. Each variable is independent of all the non-descendants, given its parent. For example, for the BN in Figure 1a we have  $p(x_4|x_1, x_2, x_3, x_5) = p(x_4|x_1, x_2)$ . The joint probability can be written as:

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i | \text{Parents}(x_i))$$

## 2.2 Pair-wise Markov Random Fields

A Markov random field (MRF) contains two types of random variables: *hidden* ( $X_1, X_2, \dots$ ) and *observed* ( $Y_1, Y_2, \dots$ ). The dependencies among variables are represented using *compatibility functions* (aka *potentials*):  $\phi_i(x_i, y_i) =$  the local evidence for  $x_i$  (sometimes written

simply  $\phi_i(x_i)$ ), and  $\psi_{ij}(x_i, x_j)$  = the compatibility between  $x_i$  and  $x_j$ . In this case we call the MRF “pair-wise” because the potential functions are defined pair-wise.

The joint probability for a pair-wise MRF can be written as:

$$p(x_1, \dots, x_n, y_1, \dots, y_m) = \frac{1}{Z} \prod_{ij} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i)$$

where  $Z$  is a normalization constant, also called *the partition function*.

## 2.3 Factor Graphs

In a factor graph (FG) the interactions between variables are represented by the *factor* nodes (the square nodes in Figure 1c), and the joint probability can be written as a product of all the factors:  $p(x) = 1/Z \prod_a f_a(x_a)$ . For example, for the FG in Figure 1c we have:

$$p(x_1, \dots, x_n) = \frac{1}{Z} f_1(x_1) f_2(x_2) f_{13}(x_1, x_3), \dots$$

It is important to notice that all three graphical models described here (BNs, pair-wise MRFs and FGs) are equivalent in terms of the systems they can model [4]. In particular, the equivalence of pair-wise MRFs and FGs can be illustrated by the equivalence between the compatibility functions and the factors:

$$\begin{aligned} \phi_i(x_i) \text{ in MRF} &\Leftrightarrow f_i(x_i) \text{ in FG} \\ \psi_{ij}(x_i, x_j) \text{ in MRF} &\Leftrightarrow f_{ij}(x_i, x_j) \text{ in FG} \end{aligned}$$

## 3 Belief Propagation (BP)

Belief propagation is a method for approximating the marginal probabilities in a graphical model, in a time linear in the number of variables (nodes). BP is precisely mathematically equivalent for pair-wise MRFs, BNs and FGs. However, it is easier to explain on a pair-wise MRF.

Let  $X_i$  be the hidden variables. We define a *message*  $m_{ij}(x_j)$  from a node  $i$  to a node  $j$  as an array containing information about what state node  $j$  should be in. For example, if  $x_j$  can take the values 1, 2, or 3, then the message  $m_{ij}(x_j) = (0.7, 0.1, 0.2)$  can be interpreted as follows: node  $i$  is telling node  $j$  that it should be in state 1 with probability 0.7, in state 2 with probability 0.1 and in state 3 with probability 0.2.

BP is an iterative algorithm. It starts with randomly initialized messages and then it applies the message update rule below until the messages converge.

$$m_{ij}(x_j) \leftarrow \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i)$$

where  $\phi_i(x_i)$  and  $\psi_{ij}(x_i, x_j)$  are the potential functions and  $N(i)$  is the set of all neighbors of node  $i$ . After convergence, the beliefs (the approximate marginals) can be computed as:

$$b_i(x_i) = k\phi_i(x_i) \prod_{j \in N(i)} m_{ij}(x_i)$$

When the MRF has no cycles, the computed beliefs are exact! Even when the MRF has cycles, the BP algorithm is still well defined and empirically it often gives good approximate answers.

## 4 The Connection between Belief Propagation and Free Energy

We can write the Gibbs free energy of a system as  $G = E - T \cdot S$ , where the entropy is  $S = -\sum p(x) \log(x)$  ( $x$  iterates over all possible states of the system).

For a graphical model, if  $x$  is an instantiation of all the random variables (or equivalently, the state of the model), we can write the joint probability  $p(x)$  as using Boltzmann's law as  $p(x) = \frac{1}{Z} e^{-E(x)}$ . Notice that we considered  $T = 1$  and we ignored Boltzmann's constant (which affects only the scale of the units).

Using BP, we approximated the joint  $p(x)$  with the belief  $b(x)$ . To see how close these two distributions are we can use the Kullback-Leibler distance:

$$D(b(x)||p(x)) = \sum_x b(x) \ln \frac{b(x)}{p(x)} = \sum_x b(x) \ln(x) + \sum_x b(x) E(x) + \ln Z$$

Notice that the second equality was obtained by substituting  $p(x)$  as defined using Boltzmann's law.

Two important properties of the KL distance are that it is always  $\geq 0$  and it becomes zero if and only if the two probability distributions are exactly the same. In this case we have:

$$G = \underbrace{\sum_x b(x) E(x)}_E + \underbrace{\sum_x b(x) \ln b(x)}_{-S} = -\ln Z$$

In other words, when the beliefs are exact, the approximate Gibbs free energy achieves its minimal value  $-\ln Z$ , also called the "Helmholtz free energy" [5].

## 5 Generalized Belief Propagation (GBP)

Computing the Gibbs free energy  $G = E - TS = \sum_x b(x) E(x) + T \sum_x b(x) \ln b(x)$  involves summing over an exponential number of terms, which is usually computationally intractable. Instead, several approximations of the free energy have been developed:

- **mean-field** approximations use only one-node beliefs,
- **Bethe** approximations [6] uses one-node and two-node beliefs,
- **region-based** approximations are based on the following idea: break up the graph into a set of regions, compute the free energy over each region and then approximate the total free energy by the sum of the free energies over the regions.

Region-based approximations can be computed using generalized belief propagation (GBP), a message-passing algorithm similar to BP. GBP uses messages between regions of nodes instead of messages between nodes. The regions that exchange messages can be visualized as *region graphs* [5].

Different choices of region graphs give different GBP algorithms. It is usually a tradeoff between complexity and accuracy. However, how to optimally choose the the regions is still an open research question. A good advice is to try to include at least the shortest cycles inside regions.

Similar to BP, GBP is guaranteed to give the optimal solution when the region graph has no cycles, but it also works very well on many graphs that contain cycles. In [5], Yedidia *et al.* try to understand why GBP works on graphs with cycles and they offer some guidelines for constructing region graphs for which some theoretical guarantees can be offered.

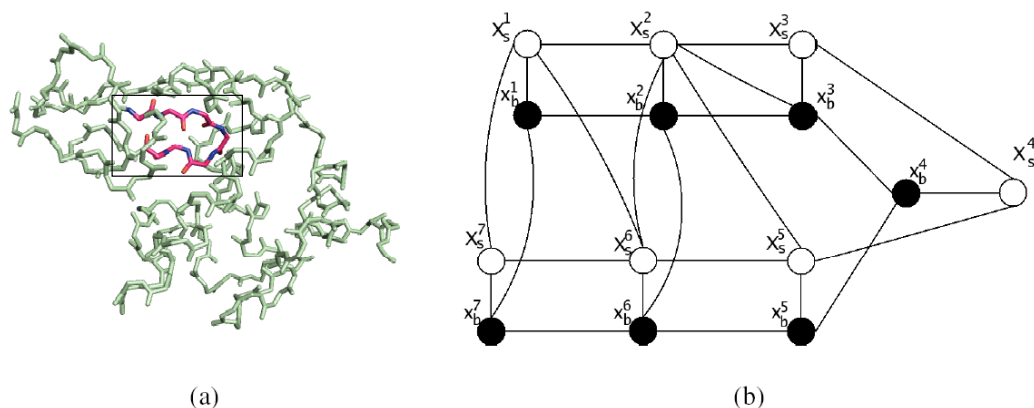
## 6 An application of GBP: Estimating the Free Energy of Protein Structures

In [3], Kamisetty *et al.* use GBP to estimate the free energy of all-atom protein structures. They model the structure of a protein as a complex probability distribution using a pair-wise MRF:

- the *observed* variables are the backbone atom positions  $X_b^i$  (continuous),
- the *hidden* variables are the side chain atom positions  $X_s^i$  represented using rotamers (discrete),
- two variables (atoms) share an edge (interact) if they are closer than a threshold distance, set to  $8\text{\AA}$ ,
- the *potential functions* are defined as:

$$\begin{aligned}\psi(X_s^{i_p}, X_s^{j_q}) &= \exp(-E(x_s^{i_p}, x_s^{j_q})/k_bT) \\ \psi(X_s^{i_p}, X_b^j) &= \exp(-E(x_s^{i_p}, x_b^j)/k_bT)\end{aligned}$$

where  $E(x_s^{i_p}, x_s^{j_q})$  is the energy of interaction between rotamer state  $p$  of residue  $X_i^s$  and rotamer state  $q$  of residue  $X_j^s$ .



**Figure 2:** (a) Structure of lysozyme (pdb id: 2lyz) with a few residues highlighted (b) Part of the random field induced by the highlighted residues:  $X_s^i$ 's are the hidden variables representing the rotameric state, the visible variables are the backbone atoms in conformations  $x_b^i$ .

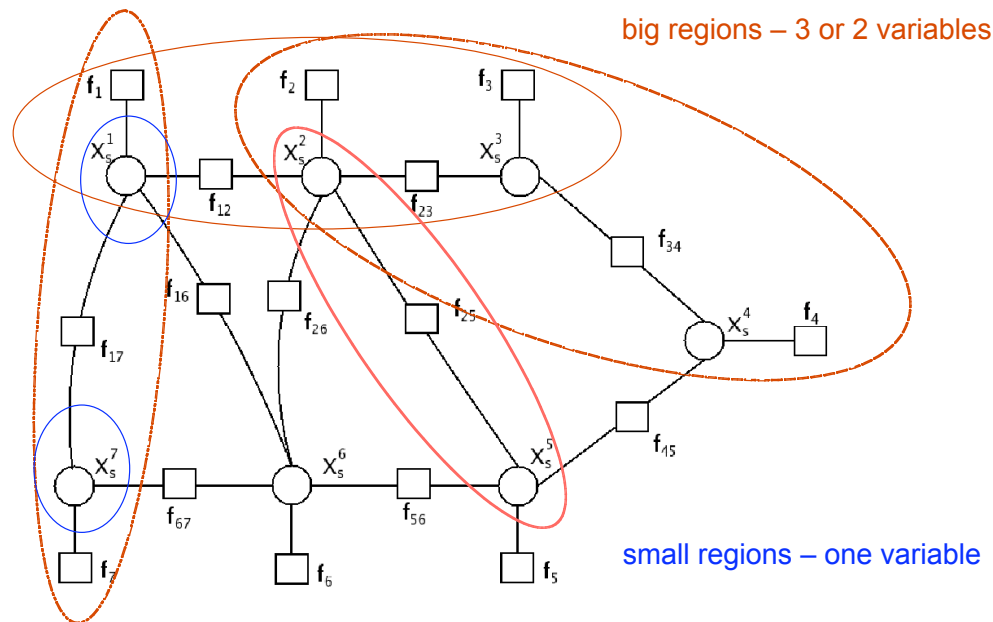
After building the pair-wise MRF, Kamisetty *et al.* transform the MRF into the equivalent factor graph. Then, they split the factor graph into regions: *big regions* containing two or three variables and *small regions* containing just one variable (see the examples in Figure 3). The basic idea behind the choice of regions: place residues that are closely coupled together in the same big region. This idea was borrowed from Aji and McEliece [7], and it provides a balance between the accuracy of the approximation and the complexity of the model.

The region-graph is then built by drawing an edge from each big region to all the small regions that contain a strict subset of the big region's nodes. Finally, a GBP algorithm called "the two-way algorithm" [5] is applied on the resulting region graph.

## 6.1 Results

Kamisetty *et al.* use the entropy component of their free energy estimate to distinguish between native protein structures and decoys (structures with similar internal energy to that of the native structure, but otherwise incorrect). They find that when the structures are ranked in decreasing order of their entropy, the native structure is ranked the highest in 87.5% of the datasets that contain decoys with backbones similar to the backbone of the native structure, and 84% of the datasets that include dissimilar backbones.

The authors also compute estimates of the  $\Delta\Delta G$  upon mutation and their results correlate well with the experimental values (with correlation coefficients varying from 0.63 to 0.7, and p-values between  $1.5e-5$  and 0.0063).



**Figure 3:** The factor graph.

## References

- [1] Lazaridis, T., Karplus, M. (2000) Effective energy functions for protein structure prediction, *Current Opinion in Structural Biology* 2000, **10**:138–145.
- [2] Thomas, P.D., Dill, K.A. (1996) Statistical Potentials Extracted From Protein Structures: How Accurate Are They?, *Journal of Molecular Biology*, 257, 457–469
- [3] Kamisetty, H., Xing, E.P., Langmead, C.J. (2006) Free Energy Estimates of All-atom Protein Structures Using Generalized Belief Propagation, CMU-CS-06-160, November 2006.
- [4] Yedidia, J.S., Freeman, W.T., Weiss, Y. (2002) Understanding Belief Propagation and its Generalizations, TR-2001-22, Mitsubishi Electric Research Laboratories, January 2002.
- [5] Yedidia, J.S., Freeman, W.T., Weiss, Y. (2005) Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms, *IEEE Transactions on Information Theory*, July 2005, 51(7): 2282-312.
- [6] Yedidia, J.S., Freeman, W.T., Weiss, Y. (2001) Bethe free energy, Kikuchi approximations, and belief propagation algorithms, TR-2001-10, Mitsubishi Electric Research Laboratories, May 2001.

- [7] Aji, S.M., McEliece, R.J. (2003) The generalized distributive law and free energy minimization. *Proceedings of the 39th Allerton Conference on Communication, Control and Computing*, 459–467.