Graph Cuts and Protein Design

Adi Mukund Jennifer Zou

1 The Protein Design Problem

The protein design problem can be described as follows. Given a set of backbone coordinates $\mathbf{c} = (\vec{c_1}, \vec{c_2}, \ldots, \vec{c_n})$ and rotamer library R of length r, identify the optimal rotamer assignment sequence $\vec{r} = (r_1, r_2, \ldots, r_n)$, $r_i \in R$, $1 \leq i \leq n$ according to an energy function $E(\vec{r})$. This assignment sequence is known as the global minimum energy conformation (GMEC).

2 Graph Labeling

The graph-label formulation of the protein design problem represents the set of all residues as a sparse interaction graph, a modification of the residue interaction graph $G = (V, \mathcal{E})$ in which each vertex represents a protein residue and each edge represents an interaction between two such residues. The sparse graph is created by omitting a set of interactions $\mathcal{E}^*$ to create the graph $G^* = (V, \mathcal{E} - \mathcal{E}^*)$.

The label graph G_l is generated by adding a set of vertices V_r representing all of the rotamers in the rotamer library R to the vertex set V^* of G_* and connecting each vertex in V^* to each vertex in V_r to generate the vertex set V_l and corresponding edge set $\mathcal{E}_l$. The set of all edges connecting vertices in V_r and V^* is $\mathcal{E}_r$. Edges from $\mathcal{E}_l$ are then pruned until every vertex in V^* is connected to exactly 1 vertex in V_r (that is, until every member of V^* is represented in $\mathcal{E}_r$ exactly once) in order to generate a labeling for the G where each vertex in V^* is connected to a vertex in V_r representing its optimal rotameric assignment.

Given a function $\phi(v, r_v)$ that returns the cost of assigning label r_v to vertex v and a function $\psi(v, r_v, w, r_w)$ that returns the pairwise interaction between residues v and w when assigned labels r_v and r_w respectively, the energy of the graph (and associated rotamer assignment sequence) can be computed as follows:

The energy of the graph can then be computed by finding the sum over the rotamers $v, w \in V^*$ where $w > v$ and their associated labels r_v and r_w as follows:

$$E_G = \sum_v \phi(v, r_v) + \sum_v \sum_{w>v} \psi(v, r_v, w, r_w) \tag{1}$$

Thus, finding the GMEC is equivalent to minimizing equation 1 over the set of all possible labelings of G_l.

3 Approximate Labeling based on Linear Programming

Note: This section is based on the paper "Approximate Labeling via Graph Cuts Based on Linear Programming" by Komodakis and Tziritas.

3.1 Metric Labeling

The Metric Labeling (ML) problem can be stated as follows: classify a set $\mathcal{V}$ of n objects by assigning to each object a label from a given set $\mathcal{L}$ of labels, given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$. For each $p \in \mathcal{V}$ there is a label cost $\mathbf{c}_p(a) \geq 0$ for assigning the label $a = f_p$ to p, and for every edge pq there is a pairwise cost $\mathbf{c}_{pq}(a, b) = w_{pq} d_{pq}(ab)$ where $d_{pq}(ab)$ is the distance between (or cost of) label a on vertex p and b on vertex q. Thus, the cost of a labeling f is as follows:

$$\text{COST}(f) = \sum_{p \in \mathcal{V}} \mathbf{c}_p(f_p) + \sum_{(p,q) \in \mathcal{E}} w_{pq} d_{pq}(f_p, f_q) \tag{2}$$

3.2 The Primal-Dual Schema

Given the primal program $\{\min \mathbf{c}^T \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ it is possible to formulate the dual program $\{\max \mathbf{b}^T \mathbf{y} \mid \mathbf{A}^T \mathbf{y} \leq \mathbf{c}\}$. The primal-dual principle states that if $\mathbf{c}^T \mathbf{x} \leq f \cdot \mathbf{b}^T \mathbf{y}$, then $\mathbf{x}$ is an f -approximation to the optimal integral solution $\mathbf{x}^*$, that is, $\mathbf{c}^T \mathbf{x}^* \leq \mathbf{c}^T \mathbf{x} \leq f \cdot \mathbf{c}^T \mathbf{x}^*$.

The Primal-Dual Schema is to keep generating pairs of primal and dual solutions $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k=1}^t$ until the pair $(\mathbf{x}^t, \mathbf{y}^t)$ are both feasible and satisfy the relaxed primal complementary slackness conditions.

Some shenanigans then occur, and I have no idea how it all works, but it seems that a (roughly) 1.4-approximation can be found. It might be useful to implement this and see if we get better results empirically.

4 A* all the things

Just A* over the cut space - maintain a sorted list of partial cuts, expanding until we get the best cut. This also guarantees ensemble enumeration! Though really this is pretty pointless.