Loopy Belief Propagation for Bipartite Maximum Weight B-Matching

Bert Huang, Tony Jebara
Computer Science Department
Columbia University, New York, NY 10027

In addition to its classical applications, recent work has demonstrated weighted b-matching's utility as a preprocessing step for spectral clustering [3]. We formulate the weighted b-matching objective function as an equivalent probability distribution function and show that belief propagation (BP) on its graphical model converges to the optimal b-matching. Standard belief propagation on our graphical model cannot be computed in polynomial time, but we introduce an algebraic method to circumvent the combinatorial message updates. Empirically, the resulting algorithm is on average faster than optimizing using the open source GOBLIN library [4], while scaling at the same asymptotic rate of $O(bn^3)$.

This work generalizes and extends previous work on BP to solve maximum weight matching [1] (which is equivalent to b-matching when $b = 1$). It uses a similar approach to that of [2], where we describe a probability distribution for a combinatorial optimization on which standard BP is intractable, and we compute the belief propagation in closed form, producing an efficient algorithm.

Consider a bipartite graph $G = (U, V, E)$ such that $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$, and $E = (u_i, v_j), \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, n\}$. Let A be the weight matrix of G such that the weight of edge (u_i, v_j) is A_{ij}. Let a b-matching be characterized by a function $M(u_i)$ or $M(v_j)$ that returns the set of neighbor vertices of the input vertex in the b-matching. The b-matching objective function can then be written as

$$\max_M [BM(G, M)] = \max_M \left[\sum_{i=1}^{n} \sum_{v_k \in M(u_i)} A_{ik} + \sum_{j=1}^{n} \sum_{u_l \in M(v_j)} A_{lj} \right] \quad s.t. \quad \begin{array}{l} |M(u_i)| = b, \forall i \in \{1, \ldots, n\} \\ |M(v_j)| = b, \forall j \in \{1, \ldots, n\} \end{array} .$$

If we define variables $x_i \in X$ and $y_j \in Y$ for each vertex such that $x_i = M(u_i)$, and $y_j = M(v_j)$, we can define the following functions:

$$\phi(x_i) = \exp(\sum_{v_j \in x_i} A_{ij}), \qquad \phi(y_j) = \exp(\sum_{u_i \in y_j} A_{ij}), \qquad \psi(x_i, y_j) = \neg(v_j \in x_i \oplus u_i \in y_j).$$

Using the potentials and pairwise clique functions, we can write out the weighted b-matching objective as a probability distribution $p(X, Y) \propto \exp(BM(G, M))$. [1]

$$p(X, Y) = \frac{1}{Z} \prod_{i=1}^{n} \prod_{j=1}^{n} \psi(x_i, y_j) \prod_{k=1}^{n} \phi(x_k)\phi(y_k)$$

We maximize this probability function using the max-product algorithm, which passes messages along the pairwise cliques. The max-product algorithm iteratively passes messages, which are vectors over settings of the variables, between dependent variables and stores beliefs, which are estimates of the max-marginal for each variable. We update messages and beliefs with the following:

$$\begin{aligned} m_{x_i}(y_j) &= \frac{1}{Z} \max_{x_i} \left[\phi(x_i)\psi(x_i, y_j) \prod_{k \neq j} m_{y_k}(x_i) \right] \\ b(x_i) &= \frac{1}{Z} \phi(x_i) \prod_k m_{y_k}(x_i) \end{aligned}$$