Discrete Hyper-graph Matching

Junchi Yan[12], Chao Zhang[2], Hongyuan Zha[3], Wei Liu[2], Xiaokang Yang[1], Stephen M. Chu[2]
[1]Shanghai Jiao Tong University. [2]IBM Research. [3]Georgia Institute of Technology.

This paper focuses on the problem of hyper-graph matching, by accounting for both unary and higher-order affinity terms. Our method is in line with the linear approximate framework while the problem is iteratively solved in discrete space. It is empirically found more efficient than many extant continuous methods. Moreover, it avoids unknown accuracy loss by heuristic rounding step from the continuous approaches. Under weak assumptions, we prove the iterative discrete gradient assignment in general will trap into a degenerating case – an m-circle solution path where m is the order of the problem. A tailored adaptive relaxation mechanism is devised to detect the degenerating case and makes the algorithm converge to a fixed point in discrete space. Evaluations on both synthetic and real-world data corroborate the efficiency of our method.

The hyper-graph matching problem can be formulated as the constrained combinatorial optimization problem adopted by extant works [1, 2]:

$$\mathbf{p}^* = \arg\max(\mathbf{H} \otimes_1 \mathbf{p} \otimes_2 \mathbf{p} \ldots \otimes_m \mathbf{p}) \quad (1)$$
$$s.t. \quad \mathbf{P}\mathbf{1}_{n_2} = \mathbf{1}_{n_1}, \mathbf{P}^T\mathbf{1}_{n_1} \leq \mathbf{1}_{n_2}, \mathbf{p} \in \{0,1\}^{n_1 n_2 \times 1}$$

We first propose our baseline method *Hyper Discrete Gradient Assignment* (HDGA) which uses a series of iterative linear assignments to approximate the original objective by fixing m-1 subsequent variables $\{\mathbf{p}_j\}_{j=k+2-m}^{k}$ in the objective (1). Then it adopts the Hungarian method (denoted as $H_d(\cdot)$) to obtain the *global-optimal discrete* solution regarding the linearized objective per iteration: $\mathbf{p}_{k+1}$=$H_d(\mathbf{H} \otimes_1 \mathbf{p}_{k+2-m} \ldots \otimes_{m-1} \mathbf{p}_k)$ as stated in Alg.1.

Theorem (1) shows under the moderate condition *i.e.* '*equal-score* equals *unique-solution*' as stated in Assumption (1), HDGA will converge to an m-point cycling solution path.

Theorem (2) shows by modifying the affinity tensor, the solution path will converge to a fixed point. This theory stimulates the *adaptively regularized* gradient assignment algorithm as described in Alg.2. We use the term *adaptively regularized* because it gradually modifies the objective function that implicitly penalizes the deviation between two iterative solutions.

We summarize the features of HADGA: i) solve the problem in discrete domain by theoretical convergence guarantee; ii) the convergence is fulfilled by adaptively modifying the affinity tensor which avoids the m-circle case, meanwhile induces a relaxed objective that keeps the global optimum unchanged for the original objective; iii) the approach is generalizable to any-order problem and no matter the affinity tensor is full or sparse, as its formulation is independent from the property of the affinity tensor.

Assumption, definitions, lemmas and theorems are listed as follows.

Assumption 1 *Given* $\boldsymbol{p}_{i_1}, \boldsymbol{p}_{i_2}, \ldots, \boldsymbol{p}_{i_{m-1}}$; $\forall j,k, j \neq k$, *assume* $\boldsymbol{H} \otimes_1 (\boldsymbol{p}_j - \boldsymbol{p}_k) \otimes_2 \boldsymbol{p}_{i_1} \ldots \otimes_m \boldsymbol{p}_{i_{m-1}} \neq 0$; $\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}_k$ *are vectorized permutation matrix in Alg.1's solution chain.*

Theorem 1 *Alg.1 will converge to an m-point circle:* $(\boldsymbol{p}_{k-m}, \boldsymbol{p}_{k+1-m}, \ldots, \boldsymbol{p}_{k-1}) \rightarrow \cdots \rightarrow (\boldsymbol{p}_{k-m}, \boldsymbol{p}_{k+1-m}, \ldots, \boldsymbol{p}_{k-1}) \rightarrow \cdots$ *if Assumption (1) holds.*

Definition 1 *For an m-tuple* $(j_1, j_2, \ldots, j_m)$, *where* $\{j_k\}_{k=1}^{m}$ *is a positive integer, its multiplicity is defined as the largest number of duplicate elements in the tuple.*

Definition 2 *A q-multiplicity unit tensor is defined as:*

$$\boldsymbol{H}^{(q)} = \begin{cases} \boldsymbol{H}^{(q)}_{i_1,i_2,\ldots,i_m} = 1 & (i_1, i_2, \ldots, i_m)\text{'s multiplicity is } q \\ \boldsymbol{H}^{(q)}_{i_1,i_2,\ldots,i_m} = 0 & \text{otherwise} \end{cases}$$

Theorem 2 *If Assumption (1) holds, one can introduce* $\alpha_2, \alpha_3, \ldots, \alpha_m$ *to modify the affinity tensor by* $\boldsymbol{H} = \boldsymbol{H} + \alpha_2\boldsymbol{H}^{(2)} + \alpha_3\boldsymbol{H}^{(3)} + \ldots + \alpha_m\boldsymbol{H}^{(m)}$ s.t. *HDGA will converge to a fixed point in the permutation matrix space.*

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.

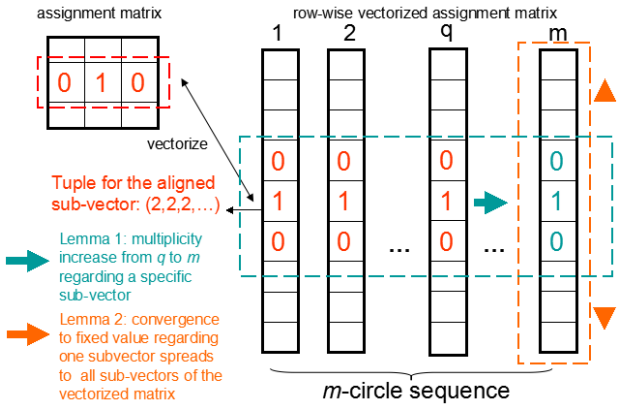

Figure 1: The *dimension-spread* idea illustration regarding Lemma 1 and Lemma 2: HADGA will converge to a fixed point.

Lemma 1 *For the n m-tuple* $\{(j_1^d, j_2^d, \ldots, j_m^d)\}_{d=1}^{n}$ *derived from the m-circle sequence* $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ *generated by HDGA, by distorting the affinity tensor by HADGA, the largest multiplicity* q_k *can increase to m. In other words, the sub-part* $(\boldsymbol{p}_1^k, \boldsymbol{p}_2^k, \ldots, \boldsymbol{p}_m^k)$ *converges to a fixed point.*

Lemma 2 *For the m-circle* $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ *generated by HDGA, if it converges to a fixed point regarding the sub-part* $(\boldsymbol{p}_1^k, \boldsymbol{p}_2^k, \ldots, \boldsymbol{p}_m^k)$, *then by distorting the affinity tensor* $\boldsymbol{H}$ *by HADGA,* $(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$ *will converge to a fixed point in the permutation matrix space.*

Lemma 1 and Lemma 2 directly establish Theorem 2 as illustrated in Fig.1. Fig.2 illustrates HDAGA performs better as iteration continues compared with the baseline HDGA and another peer method HIPFP.

Algorithm 1 Hyper Discrete Gradient Assignment

Input: $\mathbf{p}_0 = [\frac{1}{n^2}, \cdots, \frac{1}{n^2}]^T$, $\mathbf{H}$, K
Output: vectorized assignment matrix solution $\mathbf{p}^*$
for k=1:K **do**
  $\mathbf{p}_{k+1} = H_d(\mathbf{H} \otimes_1 \mathbf{p}_{k+2-m} \otimes_2 \ldots \otimes_{m-1} \mathbf{p}_k)$
  if converge **then**
    return $\mathbf{p}^* = \mathbf{p}_{k+1}$;
  end if
end for

Algorithm 2 Hyper Adaptive Discrete Gradient Assignment

Input: $\mathbf{p}_0 = [\frac{1}{n^2}, \cdots, \frac{1}{n^2}]^T$, $\mathbf{p}^* = \mathbf{p}_0$, $\mathbf{H}$, α, K
Output: vectorized assignment matrix solution $\mathbf{p}^*$
for k=1:K **do**
  $\mathbf{p}_{k+1} = H_d(\mathbf{H} \otimes_1 \mathbf{p}_{k+2-m} \otimes_2 \cdots \otimes_{m-1} \mathbf{p}_k)$
  if converge **then**
    return: $\mathbf{p}^* = \mathbf{p}_{k+1}$
  else if Fall into the m-cycling loop sequence **then**
    $\mathbf{H} = \mathbf{H} + \alpha_2\mathbf{H}^{(2)} + \alpha_3\mathbf{H}^{(3)} + \ldots + \alpha_m\mathbf{H}^{(m)}$
  end if
end for

[1] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *CVPR*, 2009.
[2] J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted randomwalks. In *CVPR*, 2011.

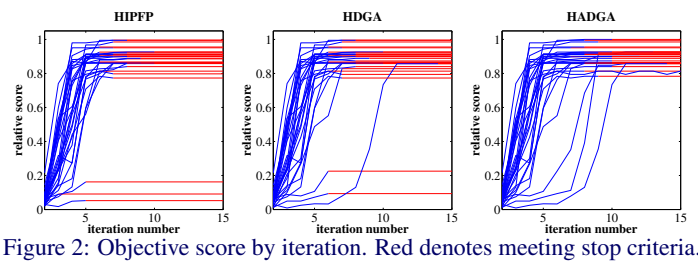

Figure 2: Objective score by iteration. Red denotes meeting stop criteria.