IEICE Trans. Inf. & Syst. Vol. E92-D, No.8, 2009, pp1580-1583

LETTER

A New Approach to Weighted Graph Matching

Kai-Jie ZHENG[†,††a)], ***Student Member***, **Ji-Gen PENG**[†b)], ***and*** **Shi-hui YING**[†††c)], ***Nonmembers***

SUMMARY Weighted graph matching is computationally challenging due to the combinatorial nature of the set of permutations. In this paper, a new relaxation approach to weighted graph matching is proposed, by which a new matching algorithm, named alternate iteration algorithm, is designed. It is proved that the algorithm proposed is locally convergent. Experiments are presented to show the effectiveness of the proposed algorithm.

key words: *Weighted graph matching; Permutation matrix; Alternate iteration algorithm.*

1. Introduction

Given two graphs with weights on edges, the aim of weighted graph matching is to find an optimal permutation of nodes of one graph such that the permuted graph is closest to the other graph.

Graph matching arises in various fields, for example, computer vision, distributed control, and facility allocation problems [2]. In computer vision, matching the structure one object to another can be formulated as graph matching[1]. In distributed control, a graph is considered as a natural mathematical description for capturing interconnection topology[3], [4]. Mathematically, facility allocation problem is a typical Quadratic Assignment Problem[5], which is similar to graph matching.

However, it has been proved that the weighted graph matching is NP-hard [5]. Indeed, the quadratic assignment problems with more 30 nodes are practically intractable[5]. There has been no polynomial time algorithm that can directly solve it with 0-1 integer solutions. The existing matching methods can be divided into two categories. The first category consists of the methods based on some forms of tree search with backtracking[6], and while the second consists of those methods based on relaxation[7]–[9]. In this paper we will consider methods in the latter category. After some preliminaries are introduced in Section 2, we propose a new relaxation approach to weighted graph matching and then derive a new matching algorithm, named alternate iteration algorithm, in Section 3. The experimental results given in Section 4 show the effectiveness of the proposed algorithm. The paper is concluded in Section 5.

[†]This work was supported by NCET and partially the National Basic Research Program of China under Grant No. 2007CB311002

[†]The author is with the Institute for Information and System Science, Xi'an Jiaotong University, Xi'an, 710049, P.R.China

[††]The author is with the School of Mathematics and Computer Science, Fujian Normal University, FuZhou, 350007, P.R.China

[†††]The author is with the School of Science, ShangHai University, ShangHai, 200444, P.R.China

a) E-mail: kaijie_zheng@163.com
b) E-mail: jgpeng@xjtu.edu.cn
c) E-mail: shying@shu.edu.cn

2. Preliminaries

A weighted graph G is an ordered pair (V, W), where V is a set of n vertices in the graph, and W is a weighting function, which gives a real nonnegative value $W(v_i, v_j)$ to each pair of vertices (v_i, v_j). An undirected graph is a graph in which the weights $W(v_i, v_j) = W(v_j, v_i)$, for $i, j = 1, 2, \cdots, n$. For a weighted directed graph, $W(v_i, v_j) \neq W(v_j, v_i)$, for some $i, j = 1, 2, \cdots, n$. The adjacency matrix of a weighted graph $G = (V, W)$ is given by $A = (a_{ij})_{nn}$, where $a_{ij} = W(v_i, v_j)$.

The problem of matching two weighted graphs $\bar{G} = (\bar{V}, \bar{W})$ and $\tilde{G} = (\tilde{V}, \tilde{W})$ consists of finding a permutation π from $\{1, 2, \cdots, \bar{n}\}$ to $\{1, 2, \cdots, \tilde{n}\}$ (generally, $\bar{n} \leq \tilde{n}$), that makes $\bar{G}$ and $\tilde{G}$ as close as possible with respect to a certain norm.

Generally, the weighted graph matching problem is formulated as [10]

$$\pi^* = arg \min_{\pi \in S_{\bar{n},\tilde{n}}} \left\{ \sum_{i=1}^{\bar{n}} \sum_{j=1}^{\bar{n}} (\bar{a}_{ij} - \tilde{a}_{\pi(i)\pi(j)})^2 \right\}, \quad (1)$$

where $S_{\bar{n},\tilde{n}}$ is the set of permutations from $\{1, 2, \cdots, \bar{n}\}$ to $\{1, 2, \cdots, \tilde{n}\}$.

Definition 1 (Permutation Matrix): A $n \times n$ matrix $P = (p_{ij})_{nn}$ is said to be a permutation matrix if $p_{ij} \in \{0, 1\}$ and

1. $\sum_{j=1}^{n} p_{ij} = 1$, for all $i = 1, 2, \cdots, n$;
2. $\sum_{i=1}^{n} p_{ij} = 1$, for all $j = 1, 2, \cdots, n$.

Definition 2 (Corresponding Matrix): An $\bar{n} \times \tilde{n} (\bar{n} \leq \tilde{n})$ matrix $P = (p_{ij})$ is said to be a corresponding matrix if there exists $\mathcal{J} = \{j_1, j_2, \cdots, j_{\bar{n}}\}$ such that the matrix $P(:, \mathcal{J}) = (p_{ij_k})_{\bar{n}\times\bar{n}}$ is a permutation matrix.

Let $\mathcal{P}(n)$ be the set of permutation matrices, and $\mathcal{P}(\bar{n}, \tilde{n})$ be the set of corresponding matrices. Then, the weighted graph matching problem can be equivalently formulated in matrix form as follows

$$P^* = arg \min\{\|\bar{A} - P\tilde{A}P^T\|^2 | P \in \mathcal{P}(\bar{n}, \tilde{n})\}, \quad (2)$$

where $\|.\|$ denotes Frobenius norm $\|X\| = tr(XX^T)^{0.5}$. The relation between π^* in (1) and P^* in (2) is

$$P^*(1, 2, \cdots, \tilde{n}) = (\pi^*_1, \pi^*_2, \cdots, \pi^*_{\tilde{n}}).$$

In the case when $\bar{n} = \tilde{n}$, the problem (2) is equivalent to the following one:

$$P^* = arg \min\{\|\bar{A}P - P\tilde{A}\|^2 | P \in \mathcal{P}(\bar{n}, \tilde{n})\}, \quad (3)$$

which is popularly considered in weighted graph matching. In this paper, we only consider the case when $\bar{n} = \tilde{n}$.

3. Alternate Iteration Algorithm

In solving the graph matching problem (3) or (2), the combinatorial nature of the set of permutations is the biggest challenge. Generally, methods based on tree search are very time-consuming. Compared with tree search methods, methods based on relaxation are very fast. Noticing the fact that a nonnegative orthogonal matrix is a permutation matrix[2], in the problem (3) (equivalently, (2)). We relax the admissible set of permutation matrices and meanwhile incorporate a barrier function into the objective functions. The problem resulted is a non-combinatorial problem.

Firstly, we formulate and then solve the following optimization problem:

$$(X^*, Y^*) = arg \min\{F_\alpha(X, Y) | X \in M^+(n), Y \in \mathcal{O}(n)\}. \quad (4)$$

where $F_\alpha(X, Y) = \|\bar{A}X - X\tilde{A}\|^2 + \alpha\|X - Y\|^2$, $\alpha > 0$ is a barrier parameter, and $\mathcal{O}(n)$ and $\mathcal{M}^+(n)$ are the set of orthogonal matrices and the set of non-negative matrices of order n, respectively.

Secondly, we address the following optimization problem

$$P^* = arg \min\{\|P - X^*\|^2 | P \in \mathcal{P}(n)\} \quad (5)$$

in order to obtain a permutation matrix closest to X^*.

Obviously, the problem (5) is equivalent to the following linear assignment problem:

$$P^* = arg \min\{tr(-X^{*T}P) | P \in \mathcal{P}(n)\} \quad (6)$$

It should be noted that in most of linear assignment algorithms the weight matrix is always assumed to be nonnegative. However, such assumption is not restrictive. In fact, for the weight matrix $-X^*$ we can make it nonnegative by plus a sufficient large constant matrix, not affecting the solution. A typical linear assignment algorithm, such as the Kuhn-Munkres algorithm, has a complexity $O(n^3)$[11].

Theorem 1: [2] $\mathcal{P}(n) = \mathcal{O}(n) \bigcap \mathcal{M}^+(n)$.

Theorem 2: Let

$$F(\alpha) = \min\{\|\bar{A}X - X\tilde{A}\|^2 + \alpha\|X - Y\|^2 | X \in M^+(n), Y \in \mathcal{O}(n)\}.$$

Then, for all $\alpha > 0$,

(1) $F(\alpha) \le \min\{\|\bar{A}P - P\tilde{A}\|^2 | P \in \mathcal{P}(n)\}$ and

(2) $F(\alpha)$ is non-decreasing on α and $\lim_{\alpha + \to \infty} F(\alpha) = \min\{\|\bar{A}P - P\tilde{A}\|^2 | P \in \mathcal{P}(n)\}$

Proof. The proof is immediate. □

It is clear to see that solving (4) is dominative in our method. In the following, we design the algorithm for problem (4).

Alternate Iteration Algorithm:
Step 0. Let $k = 0$. Given a tolerance $\epsilon > 0$, a barrier parameter $\alpha > 0$ and a initial point $X^0 \in M^+(n)$.
*Step1.*Solve the following subproblem :

$$Y^k \in arg \min\{\|X^k - Y\|^2 | Y \in \mathcal{O}(n)\}. \quad (7)$$

If $k > 0$ and $F_\alpha(X^k, Y^k) - F_\alpha(X^{k-1}, Y^{k-1}) > \epsilon$, then stop. Otherwise, let $k = k + 1$ and go to Step 2.
*Step2.*Solve the following subproblem :

$$X^k \in arg \min\{\|\bar{A}X - X\tilde{A}\|^2 + \|X - Y^{k-1}\|^2 | X \in M^+(n)\}. \quad (8)$$

Goto Step 1.□

Below is the convergence theorem for the alternate iteration algorithm.

Theorem 3: The algorithm always converges monotonically to a local minimum. That is,

$$F_\alpha(X^k, Y^k) \ge F_\alpha(X^{k+1}, Y^{k+1})$$

where $\{X^k, Y^k\}_{k=1}^{+\infty}$ are the iterative sequences.

Proof. According to $Y^k \in arg \min\{\|X^k - Y\|^2 | Y \in \mathcal{O}(n)\}$ and $X^{k+1} \in arg \min\{\|\bar{A}X - X\tilde{A}\|^2 + \|X - Y^k\|^2 | X \in M^+(n)\}$, we have

$0 \le F_\alpha(X^{k+1}, Y^{k+1}) \le F_\alpha(X^{k+1}, Y^k) \le F_\alpha(X^k, Y^k)$. So, the sequence $\{F(X^k, Y^k)\}$ is non-increasing and bounded below. It means that alternate iteration algorithm converge monotonically to a local minimum. □

It, however, should be pointed out that the alterative iteration algorithm is local. Experimentally, it decreases fast during the early few iterations and then slows down as it approaches the local minimum. This is demonstrated by the experiments done in the next section.

The process of alternate iteration algorithm consists of the following three subproblems:

(1) *The choice of initial point*. Since the algorithm is local, the choice of the initial point is very important. By Birkhoff theorem [12], any doubly stochastic matrix is a convex combination of finitely many permutation matrices. Accordingly, we here suggest choosing the solution of the quadratic convex programming:

$$X^1 = arg \min \|\bar{A}X - X\tilde{A}\|^2 + \beta(n - \mathbf{1}_n^T P \mathbf{1}_n)$$
$$s.t. \quad X\mathbf{1}_n \le \mathbf{1}_n, \mathbf{1}_n^T X \le \mathbf{1}_n^T, X \ge 0$$

as the initial point, where $\beta > 0$ is a barrier coefficient, $\mathbf{1_n}$ is a vector of length n with entry 1 . Experiments presented in the next section show that this choice strategy works well.

(2) *Quadratic convex programming (8)*. Such a standard quadratic convex programming is easy to be solved ([13], [14]).

(3) *Optimization with orthogonal constraints (7)*. It is

easy to show that the optimization is equivalent to

$$Y^k \in arg\max\{tr(YX^{kT})|Y \in O(n)\} \quad (9)$$

In the following we prove that the optimization (9) has close-form solution.

Lemma 1: If A is a semipositive matrix, then $\max\{tr(YA)|Y \in O(n))\} = tr(A)$.

Proof. Let CC^T be Cholesky decomposition of A (that is, $A = CC^T$). Then, for any orthogonal matrix Y, we have $tr(YA) = tr(YCC^T) = tr(C^TYC) = \sum_{i=1}^{n} c_i^T Y c_i$. And, by Cauchy-Schwartz inequality we have that $c_i^T Y c_i \leq [(c_i^T c_i)(c_i^T Y^T Y c_i)]^{0.5} = c_i^T c_i$. Therefore, there holds that $tr(YA) \leq tr(A)$ for any orthogonal matrix Y. □

Theorem 4: Let $A = U\Lambda V^T$ be the SVD decomposition of A. Then, $Y = VU^T$ solves $\max\{tr(YA)|Y \in O(n)\}$, .

Proof. Since $A = U\Lambda V^T$ is the SVD decomposition, we have that $VU^TA = VU^TU\Lambda V^T = V\Lambda V^T$. From Lemma 1, we know that $tr(VU^TA) \geq tr(QVU^TA)$ for any orthogonal matrix Q. Hence, it follows that $VU^T \in arg\max\{tr(YA)|Y \in O(n)\}$. □

4. Experimental Results

The performance of alternate iteration algorithm (AIGM) is compared with the performances of the Linear Programming Graph Matching algorithm (LPGM)[9] and the Improved Eigen-Decomposition Graph Matching algorithm (IEGM)[7][8]. The LPGM, IEGM algorithms are chosen, since they are considered the most cited algorithms for graph matching.

Weighted graphs set is randomly generated, where each graph has weights ranging from 0 to 1. Matching graph H is generated from each graph G in the set by adding uniformly distributed noise in the range of 0 to 1.4 to each weight in G. Then, the above three graph matching algorithms are applied to graph matching for every pair H and G. In this experiment, we set the size of graph equal to 8 for the purpose of illustration. The estimated probability of a correct vertex-vertex assignment is calculated for a given noise level ϵ, after 100 trials for each algorithm. From a point of view of probability, this reflects how well an algorithm performs for a given noise level. All programs are written in Matlab7.0 and run by PC with PentiumIV3.0GHz££ CPU and 1G RAM.

In Fig.1,The results of LPGM, IEGM and AIGM for the case $n = 8$ and $\epsilon = [0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7, 0.8, 0.9, 1, 1.2, 1.4]$ are displayed. It is seen that all the three algorithms perform well when the noise level is small,while AIGM is most robust than IEGM and AIGM with the increasing noise. The matching performance with different graph size are showed in Fig.2 and Fig.3 for the case $\epsilon = 0.2$ and $\epsilon = 0.8$,respectively. It shows that AIGM takes advantage over LPGM and IEGM as the size of graph increases.

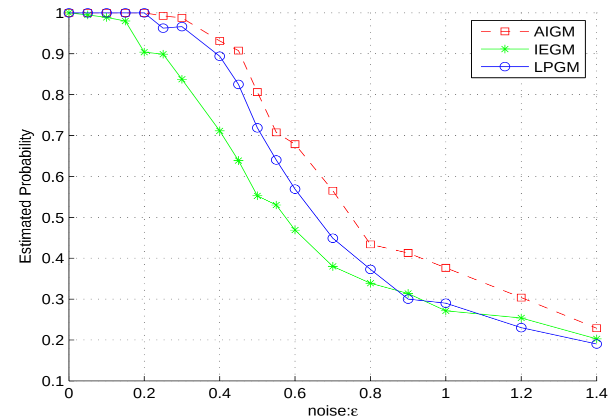

Fig. 1 Estimated Probability of correct vertex-vertex matching versus noise level ϵ[n = 8, $\alpha = 11, \beta = 10$].

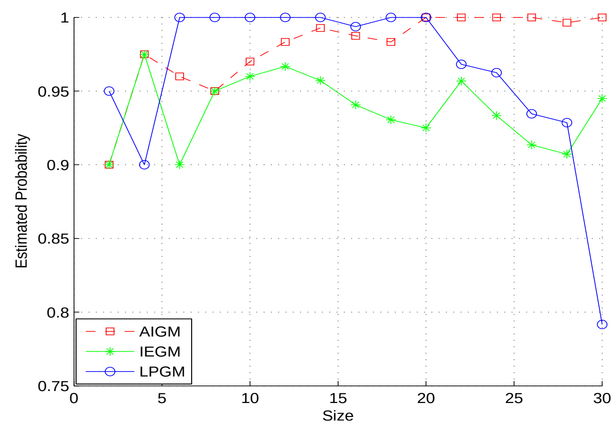

Fig. 2 Estimated Probability of correct vertex-vertex matching versus size of graph [$\epsilon = 0.2$ $\alpha = \beta = \frac{\|\bar{A}\|+\|\tilde{A}\|}{2n}$].

5. Conclusions

In this paper, a new relaxation approach to weighted graph matching has been proposed, by which a new matching algorithm, named alternate iteration algorithm, is designed. It was proved that the algorithm proposed is locally convergent. Moreover, it has been shown that the alternative iteration algorithm works better than the Linear Programming Graph Matching algorithm as well as the Improved Eigen-Decomposition Graph Matching algorithm in the comparison experiments done in this paper. However, different from those two algorithms, the AIGM is local, and as a result, the choice of the initial point of AIGM is very important. Particularly, when the difference between two matching graphs becomes larger, the performance of AIGM remains to be improved. In our future work, we will address to the performance improvement of AIGM.

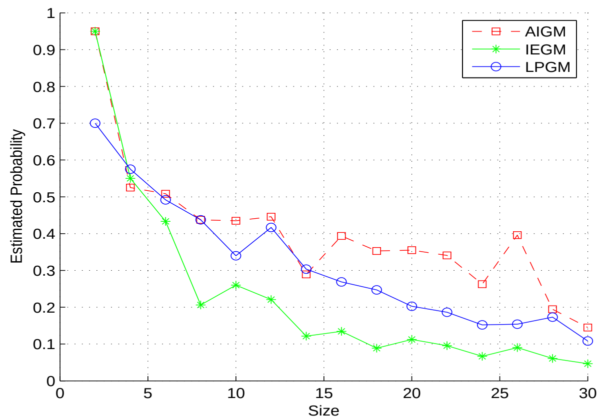

Fig. 3 Estimated Probability of correct vertex-vertex matching versus size of graph [$\epsilon = 0.8$ $\alpha = \beta = \frac{\|\bar{A}\|+\|\tilde{A}\|}{2n}$].

References

[1] S.Omachi, S.Megawa, and H.Aso, Decorative Character Recognition by Graph Matching, IEICE Trans. INF. & SYST.,vol.E-90-D, no.10: 1720-1723, 2007.

[2] M.M. Zavlanos, G.J.Pappas, A Dynamical Systems Approach to Weighted Graph Matching, Procedeing of the 45th IEEE Conference on Decision and Control, 3492-3497, 2006.

[3] M.Mesbahi, On state-dependant dynamic graphs and their controllablity properties, IEEE Trans. Automatics Control, vol.50, no.3: 382-387, 2005.

[4] A. Jadbabaie, J.Lin and A.S.Morse, Coordination Of Groups Of Mobile Autonomous Agents Using Nearest Rules, IEEE Trans. Automatics Control, vol.48, no.6: 988-1001, 2005.

[5] E.M.Loiola, N.M.Abreu, A Survey For The Quadratic Assignment Problem, European Journal of Operational Research, vol.176:657-690, 2007.

[6] D.Conte, P.Foggia, C.Sansone, and M.Vento, Thirty Years of Graph matching in Pattern Recognition, International Journal of Pattern recognition and Artificial Intellgience, vol.18,no.3:265-298, 2004.

[7] S.Umeyama, An Eigendecomposition Approach To Weighted Graph Matching,IEEE Trans. Pattern Analysis and Machine Intelligence. vol.10, no.5:695-703,1988

[8] G.Zhao, B.Luo, J.Tang, and J.Ma, Using Eigen-Decomposition Method for Weighted Graph Matching, Lecture Notes on Computer Science,4681:1283-1294, 2007.

[9] H. A. Almohamad and S. 0. Duffuaa, A Linear ProgrammingApproach for the Weighted Graph Matching Problem, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.15, no.5:522-525, 1993.

[10] S.Gold and A.Rangarian, A Graduated Assignment Algorithm For Graph Matching,IEEE Trans. Pattern Analysis and Machine Iteeligence, vol.18,no.4:,377-388, 1996.

[11] C.H.Papadimitriou and K.Stciglitr, Comhinarorial Optimization Algorithms ans Complexity, Englewood Cliffs, NJ Prentiee-Hall, 1982.

[12] R.A.Horn and C.R.Johnson, Matrix Analysis, Cambridge University Press,1985.

[13] T.F.Coleman and Y. Li, A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on some of the Variables," SIAM Journal on Optimization, vol. 6, no 4, pp. 1040-1058, 1996.

[14] D. Goldfarb and S. Liu, An O(n3L) primal interior point algorithm for convex quadratic programming,Mathematical Programming, vol.49:325-340,1991.