EFFICIENT BELIEF PROPAGATION FOR GRAPH MATCHING

Efe Onaran

Electrical and Computer Engineering
Tandon School of Engineering
New York University

Soledad Villar *

Center for Data Science and
Courant Institute of Mathematical Sciences
New York University

ABSTRACT

In this short note we derive a novel belief propagation algorithm for graph matching and we numerically evaluate it in the context of matching random graphs. The derived algorithm has a lower asymptotic time-complexity without significantly compromising the accuracy compared to leading available algorithms in the literature. An extended version of this article, with further theory and numerical simulations is in preparation.

Index Terms— Belief propagation, graph matching.

1. INTRODUCTION

Graph matching is a fundamental problem in theoretical computer science, and it is useful to model problems arising in different scientific applications. Given two graphs with n vertices and adjacency matrices A and B, the graph matching problem asks for an $n \times n$ permutation matrix P that best *aligns* both graphs by optimizing the objective

$$\min_{P \in \Pi_n} \|A - PBP^\top\| \tag{1}$$

where the norm can be the Frobenius or element-wise L_1 norm, and Π_n is the set of all $n \times n$ permutation matrices.

The graph matching problem has especially been of interest in computer vision [1], protein interaction modelling [2], and in social network privacy [3], among many scientific applications. In practice there exist several heuristic efficient algorithms, tailored to different domains, mostly without theoretical guarantees [4].

In this work we derive a belief propagation algorithm for graph matching. And we numerically evaluate its performance on Erdős-Rényi random graphs. An extended version of this work will further study this algorithm in different data models, compare it with other algorithms in the literature, and analyze linearizations of the BP algorithm with reduced computational complexity.

*Thanks to NSF-DMS 1913134, EOARD FA9550-18-1-7007 and the Simons Algorithms and Geometry (A&G) Think Tank.

2. SETTING AND PRIOR WORK

Let A be an Erdős-Rényi graph on n vertices ($A \sim \mathrm{ER}(n,p)$). Namely, we construct the graph A by drawing an i.i.d. Bernoulli random variable with probability p for each pair of vertices to decide whether there is an edge between them. To simplify the notation we will use the same name for the graph and its adjacency matrix, that is the entry at the i'th row and j'th column of the symmetric, hollow matrix A is 1 iff there is an edge between the i'th and j'th vertices of the graph A.

We define $\tilde{A}$ as a graph that is a small perturbation of A, the result of flipping some edges to non-edges and non-edges to edges independently with small probabilities:

$$p_{1|0} \coloneqq \mathbb{P}[\tilde{A}_{ij} = 1 | A_{ij} = 0] = (1-\lambda)p \tag{2}$$

$$p_{0|1} \coloneqq \mathbb{P}[\tilde{A}_{ij} = 0 | A_{ij} = 1] = (1-\lambda)(1-p) \tag{3}$$

$$p_{0|0} \coloneqq \mathbb{P}[\tilde{A}_{ij} = 0 | A_{ij} = 0] = 1-(1-\lambda)p \tag{4}$$

$$p_{1|1} \coloneqq \mathbb{P}[\tilde{A}_{ij} = 1 | A_{ij} = 1] = 1-(1-\lambda)(1-p). \tag{5}$$

For simplicity we assume $0 \leq \lambda \leq 1$ (negative correlation is also possible but we don't consider it here). Note that $p_{1|0}$, $p_{0|1}$ are the respective probabilities of flipping an edge to a non-edge, and flipping a non-edge to an edge, whereas $p_{1|1}$ and $p_{0|0}$ are the probabilities of not flipping them. Therefore note that if $\lambda = 1$ both graphs coincide, whereas if $\lambda = 0$ then $\tilde{A}$ is independent of A.

We consider the graph B defined as a random permutation of $\tilde{A}$. Namely, $B = \hat{P}^T \tilde{A} \hat{P}$ for some uniformly random $n \times n$ permutation matrix $\hat{P}$. We denote this model as $A, B \sim \mathrm{GM}(n,p,\lambda)$.

A natural question that arises from this setting is: for what values of n, p and λ is it possible to recover $\hat{P}$ from A and B? Information theoretical thresholds for exact recovery were investigated in [5], [6], [7]. They show that if $p(1-p) = \Theta(1)$ exact recovery is possible for

$$\lambda^*_{stat} = \Theta\left(\sqrt{\frac{\log n}{n}}\right). \tag{6}$$

On the other hand if $p(1-p) = o(1)$ and also $p(1-p) =$

$o(\lambda)$ then

$$p(1-p)\lambda^*_{stat} = 2\frac{\log n}{n} + \omega(n^{-1}) = \Theta\left(\frac{\log n}{n}\right) \quad (7)$$

Although information theoretic bounds have been well studied for random graph matching, no polynomial-time algorithm is known to achieve or approximate the statistical bound. In the recent work [8] the authors consider the regime where $p = o(1)$ and constant λ, and provide a quasi-polynomial time algorithm that they show achieves exact recovery of $\hat{P}$ when

$$p^*_{[8]} = \Omega(n^{\frac{(\log\log n)^2}{\sqrt{\log n}}}). \quad (8)$$

The statistical threshold for exact recovery in their regime is

$$np^*_{\text{stat}} = \Theta(\log n) = \Theta(n^{\frac{\log\log n}{\log n}}) \quad (9)$$

Authors of [8] also provide a poly-time detection algorithm to work with probability at least 0.9 when

$$np_{[8]} = n^{\delta}$$

where $\delta > 0$ is arbitrarily small.

The recent work [9] proposes a poly-time algorithm for the seeded case (where correspondences of certain nodes are given a priori). In order to extend the seeded case to the general case one may need to search for the seeds by brute force, which may take sub-exponential time. Remarkably this sub-exponential time algorithm achieves the perfect recovery information theoretic bound

$$np^*_{[9]} = \Theta(\log n). \quad (10)$$

An extension of a canonical labeling algorithm originally proposed for graph isomorphism in [10] which uses the node degrees was recently analyzed in [11] for random graph matching in the sparse regime, $p = o(1)$, where the threshold for perfect recovery was shown to be

$$\lambda^*_{[11]} = 1 - o(\log^{-6} n). \quad (11)$$

In [12], power of $1/\log n$ in (11) was improved to a smaller constant by using *degree profiles*, degrees of neighbors, to match the nodes. In addition, some weak results on the spectral algorithms have recently been derived in [13] and [14].

A belief propagation approach to graph matching similar to the one we introduce in this note was proposed in [15]. The main difference between our algorithm and the one from [15] is that the latter relies on a given non-uniform prior distribution over assignments for each node. The lack of such pre-assignment would increase the number of variable nodes in their algorithm to $\mathcal{O}(n^4)$ which would make it not suitable for large graphs.

Moreover, two related belief propagation schemes for graph matching were introduced in [16] where either the permutation constraint was enforced pairwise or it was softened to a potential on the Hamiltonian leading to different updates from ours.

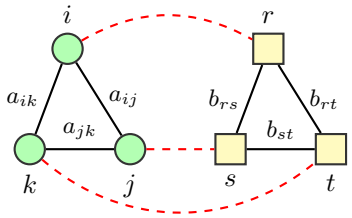
3. BELIEF PROPAGATION ALGORITHM

We rephrase graph matching as quadratic assignment according to [17]:

$$\begin{aligned} \min_{\pi\in\Pi_n} \quad & \sum_{i\in A}\sum_{j\in A}\sum_{r\in B}\sum_{s\in B} E_{irjs}P_{ir}P_{js} \\ \text{subject to} \quad & \sum_r P_{ir} = 1 \quad \forall i \in A \\ & \sum_i P_{ir} = 1 \quad \forall r \in B \\ & P_{ir} \in \{0,1\} \end{aligned} \quad (12)$$

The goal is to find the permutation between two sets of objects, A and B, that minimize the sum of individual costs, each of which depends on pairs of nodes that are matched under the permutation. There are different choices for the cost E_{irjs}. In the case of graph matching we typically have $E_{irjs} = |a_{ij} - b_{rs}|$, where a_{ij} and b_{rs} are the weights of edges in A and B respectively. We will not make this assumption and write the algorithm for general E. A and B are both assumed to have n elements.

The setting is illustrated below, where i, j and k are nodes in A and r, s, t in B. A candidate permutation illustrated with red dashed curves.



We start with writing the posterior probability of a permutation P.

$$\mathbb{P}_{\hat{P}}(P|E) = \frac{1}{Z}\prod_i \mathbb{1}\{\sum_t P_{it} = 1\}\prod_r \mathbb{1}\{\sum_k P_{kr} = 1\} \prod_j\prod_s e^{-\beta P_{ir}P_{js}E_{irjs}} \quad (13)$$

We illustrate our proposed belief propagation (BP) scheme based on the posterior (13) in Figure 1.

We consider the variable node ir to keep track of the estimates of $\mathbb{P}[\hat{P}_{ir} = 1]$ (probability of i being matched to r under $\hat{P}$). The factor node $*$ enforces the hard constraints that $\sum_s \hat{P}_{is} = 1$, $\sum_k \hat{P}_{kr} = 1$ and $\hat{P}_{ir} \in \{0,1\}$ for all $i \in A$ and $r \in B$. The factor node $irjs$ is for the soft constraint for the cost E_{irjs}. We write the sum-product message updates as follows.

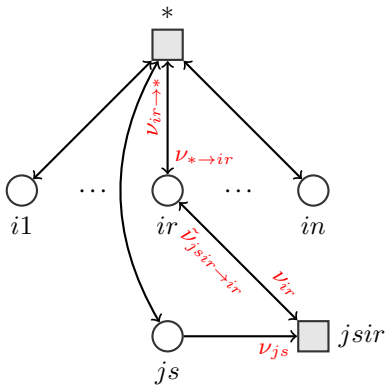


Fig. 1. Factor graph representation of our BP algorithm

- From check to variable nodes:

$$\tilde{\nu}_{jsir\to ir}(1) \cong \nu_{js}^{(\mathcal{I}-1)}(1)e^{-\beta E_{jsir}} + \nu_{js}^{(\mathcal{I}-1)}(0) \quad (14)$$

$$\tilde{\nu}_{jsir\to ir}(0) \cong \nu_{js}^{(\mathcal{I}-1)}(1) + \nu_{js}^{(\mathcal{I}-1)}(0) = 1 \quad (15)$$

- From variable to the permutation-check node(*):

$$\nu_{ir\to *}(1) \cong \nu_{ir}^{(\mathcal{I}-1)}(1) \prod_{js} \tilde{\nu}_{jsir\to ir}(1) \quad (16)$$

$$\nu_{ir\to *}(0) \cong \nu_{ir}^{(\mathcal{I}-1)}(0) \prod_{js} \tilde{\nu}_{jsir\to ir}(0) \quad (17)$$

- From the permutation-check(*) to variable nodes:

$$\nu_{*\to ir}(1) \cong \nu_{ir\to *}(1) \prod_{j\neq i} \nu_{jr\to *}(0) \prod_{s\neq r} \nu_{is\to *}(0) \quad (18)$$

$$\nu_{*\to ir}(0) \cong \nu_{ir\to *}(0) \sum_{s\neq r, j\neq i} \nu_{is\to *}(1)\nu_{jr\to *}(1) \prod_{t\neq s,r} \nu_{it\to *}(0) \prod_{k\neq j,i} \nu_{kr\to *}(0) \quad (19)$$

- From variable back to check nodes:

$$\nu_{ir}^{\mathcal{I}}(1) \cong \nu_{*\to ir}(1)\nu_{ir\to *}(1) \quad (20)$$

$$\nu_{ir}^{\mathcal{I}}(0) \cong \nu_{*\to ir}(0)\nu_{ir\to *}(0) \quad (21)$$

The algorithm has two cycles at each iteration. First cycle is exchanging messages through soft constraint factor nodes and second cycle exchanges messages through the hard constraint factor node(*). $\nu_{ir}^{\mathcal{I}}(1)$ denotes the probability of i in A being matched to r in B at iteration $\mathcal{I}$ whereas $\nu_{ir}^{\mathcal{I}}(0)$ denotes the probability of i *not* being matched to r.

Enforcing $\nu_{ir}(0) + \nu_{ir}(1) = 1$, combining two cycles of the algorithm (soft constraint and hard constraint message exchanges) and changing messages to log-likelihood ratio, $L_{ir} = -\log(\nu_{ir}(0)/\nu_{ir}(1))$, we simplify each iteration of the algorithm to one update rule as follows.

$$\begin{aligned} L_{ir} \leftarrow & 2\left(L_{ir} + \sum_{j,s} \log \frac{1+e^{L_{js}-\beta E_{jsir}}}{1+e^{L_{js}}}\right) \\ & - \log \sum_{k\neq i} \exp\left\{L_{kr} + \sum_{j,s} \log \frac{1+e^{L_{js}-\beta E_{jskr}}}{1+e^{L_{js}}}\right\} \\ & - \log \sum_{t\neq r} \exp\left\{L_{it} + \sum_{j,s} \log \frac{1+e^{L_{js}-\beta E_{jsit}}}{1+e^{L_{js}}}\right\} \end{aligned} \quad (22)$$

Note presented above is the Sum-Product BP update. For Max-Product, simply maximization replaces summation.

3.1. Stopping condition

We set the norm difference of beliefs, $\nu_{ir}(1) = 1/(1+e^{-L_{ir}})$ which will also be denoted as V_{ir}, between two consecutive iterations as our stopping condition. That is, iterations stop when

$$\sum_{i,r} (V_{ir}^{\mathcal{I}} - V_{ir}^{(\mathcal{I}-1)})^2 < c^2 \quad (23)$$

where typically $c = 10^{-2}$. After the algorithm stops, the beliefs are projected onto a permutation by a greedy assignment algorithm which approximately solves

$$\arg\max_{P\in\Pi_n} \mathrm{Tr}(P^T V). \quad (24)$$

It is described in Algorithm 2 below.

3.2. Computational complexity and faster implementation for binary graphs

Complexity of one iteration is $\mathcal{O}(n^4)$ and the bottleneck step is the computation of the matrix M. If a special structure on the cost matrix E is assumed though, complexity can be reduced. If all entries of E are assumed to be 0 or 1, which is the case for graph matching, update of matrix M, can be rewritten as

$$M_{ir} = L_{ir} + \sum_{j,s} \log\left(1 - (1-e^{-\beta})V_{js}\right) E_{jsir} \quad (25)$$

Note for graph matching costs further satisfy (27), which lets us succinctly write the M update as (28).

$$E_{irjs} := \mathbb{1}\{i\neq j\}\,\mathbb{1}\{r\neq s\}|A_{ij} - B_{rs}| \quad (26)$$

$$= \mathbb{1}\{i\neq j\}\,\mathbb{1}\{r\neq s\}(A_{ij} + B_{rs} - 2A_{ij}B_{rs}) \quad (27)$$

Algorithm 1: Belief Propagation for min-QAP

Input: cost matrix E
Initialize Beliefs (V^0);
Compute initial LLRs: $L_{ir} = \log\left(\frac{1}{1-V^0_{ir}} - 1\right)$;
$\mathcal{I} = 0$; $c = 10^{-1}$;
while $\|V^{\mathcal{I}} - V^{\mathcal{I}-1}\|_F > c$ **do**
  $\mathcal{I} = \mathcal{I} + 1$;
  $M_{ir} = L_{ir} + \sum_{j,s} \log \frac{1+e^{L_{js}-\beta E_{jsir}}}{1+e^{L_{js}}}$;
  Sum-Product:
    $L_{ir} = 2M_{ir} - \log \sum_{t \neq r} e^{M_{it}} - \log \sum_{k \neq i} e^{M_{kr}}$;
  Max-Product:
    $L_{ir} = 2M_{ir} - \max_{t \neq r} M_{it} - \max_{k \neq i} M_{kr}$;
  Compute Beliefs: $V^{\mathcal{I}}_{ir} = \frac{1}{1+e^{-L_{ir}}}$;
end
Output: greedyAssign(V)

Algorithm 2: greedyAssign

Input: V
$P = \text{zeros}(V.\text{shape})$;
$\tilde{V} = \text{sort}(V)$;
while $\max(\tilde{V}) > -\infty$ **do**
  row_arg, col_arg $= \arg\max(\tilde{V})$;
  P[row_arg, col_arg] $= 1$;
  $\tilde{V}$[row_arg, :] $= -\infty$;
  $\tilde{V}$[:, col_arg] $= -\infty$
end
Output: P

$$M = A^T V^\beta (J - I - B) + (J - I - A^T) V^\beta B + L \quad (28)$$

where entries of the matrix V^β are given as

$$V^\beta_{js} = \log\left(1 - (1 - e^{-\beta}) V_{js}\right) \quad (29)$$

and J and I are the $n \times n$ all-one and identity matrices, respectively.

Complexity of the approximate greedy assignment algorithm used at the end is $\mathcal{O}(n^2 \log n)$ which keeps the overall complexity of the algorithm at that of computing M, which is just the complexity of multiplication of n-by-n matrices. Multiplication of n-by-n matrices can be done more efficiently than $\mathcal{O}(n^3)$ with off-the-shelf packages, e.g. BLAS [18]. Another option for the rounding step is to use the exact $\mathcal{O}(n^3)$ Hungarian Algorithm instead [19] which would be the bottleneck in efficiency and increase the overall complexity to $\mathcal{O}(n^3)$ as a result. Run-times with Hungarian and greedy roundings did not differ much in the GM experiments discussed below.

4. NUMERICAL PERFORMANCE

We evaluate the performance of our belief propagation algorithm on the graph matching model $GM(n, p, \lambda)$ described in Section 2. We compare the performance of our algorithm with leading algorithms in the literature. Due to space constraints we report a limited set of experiments, a more extensive report is in preparation.

Figure 2 shows the normalized overlap $\text{Tr}(P^T \hat{P})/n$ between the output of our algorithm(P) and the true planted permutation($\hat{P}$) averaged over 30 i.i.d experiments of GM$(n, 0.5, \lambda)$ for varying n and λ values.

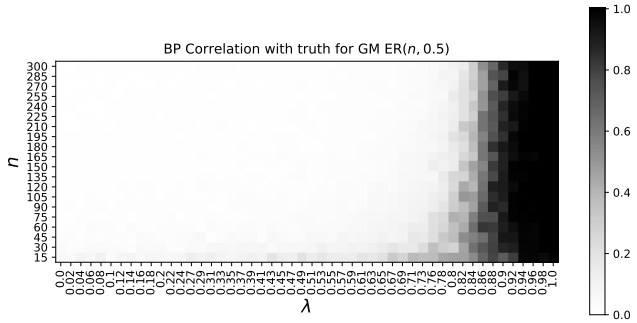


Fig. 2. Normalized overlap with planted permutation for BP in GM$(n, 0.5, \lambda)$.

Figure 3 illustrates the performance of the convex-concave relaxation algorithm, *PATH* [20] for the GM$(n, 0.5, \lambda)$ model. Although PATH seems to outperform belief propagation in terms of accuracy, it's less efficient. PATH needs more iterations to converge and since every step takes $\mathcal{O}(n^3)$ operations, we could not go beyond $n = 90$ in our experiments in a reasonable time.

We observe a worse recovery performance by Frank-Wolfe & Gradient-Descent type algorithms for graph matching [21] and an earlier eigendecomposition approach [22] (both have $\mathcal{O}(n^3)$ scaling). An extensive set of experiments will be reported at an extended version of this work.

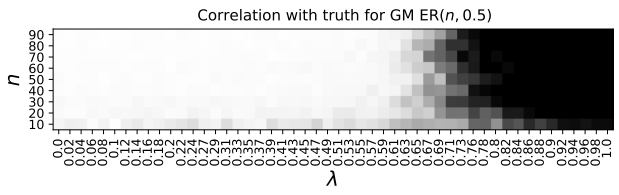


Fig. 3. *PATH* algorithm[20] in GM$(n, 0.5, \lambda)$

5. REFERENCES

[1] Feng Zhou and Fernando De la Torre, "Factorized graph matching," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 127–134.

[2] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert, "Global alignment of protein–protein interaction networks by graph matching methods," *Bioinformatics*, vol. 25, no. 12, pp. i259–1267, 2009.

[3] Arvind Narayanan and Vitaly Shmatikov, "De-anonymizing social networks," in *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE, 2009, pp. 173–187.

[4] Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento, "Thirty years of graph matching in pattern recognition," *International journal of pattern recognition and artificial intelligence*, vol. 18, no. 03, pp. 265–298, 2004.

[5] Pedram Pedarsani and Matthias Grossglauser, "On the privacy of anonymized networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 1235–1243.

[6] Efe Onaran, Siddharth Garg, and Elza Erkip, "Optimal de-anonymization in random graphs with community structure," in *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE, 2016, pp. 709–713.

[7] Daniel Cullina and Negar Kiyavash, "Improved achievability and converse bounds for erdos-rényi graph matching," in *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. ACM, 2016, pp. 63–72.

[8] Boaz Barak, Chi-Ning Chou, Zhixian Lei, Tselil Schramm, and Yueqi Sheng, "(nearly) efficient algorithms for the graph matching problem on correlated random graphs," *arXiv preprint arXiv:1805.02349*, 2018.

[9] Elchanan Mossel and Jiaming Xu, "Seeded graph matching via large neighborhood statistics," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2019, pp. 1005–1014.

[10] László Babai, Paul Erdos, and Stanley M Selkow, "Random graph isomorphism," *SIAM Journal on Computing*, vol. 9, no. 3, pp. 628–635, 1980.

[11] Osman Emre Dai, Daniel Cullina, Negar Kiyavash, and Matthias Grossglauser, "Analysis of a canonical labeling algorithm for the alignment of correlated erdos-rényi graphs," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 2, pp. 36:1–36:25, June 2019.

[12] Jian Ding, Zongming Ma, Yihong Wu, and Jiaming Xu, "Efficient random graph matching via degree profiles," *arXiv preprint arXiv:1811.07821*, 2018.

[13] Zhou Fan, Cheng Mao, Yihong Wu, and Jiaming Xu, "Spectral graph matching and regularized quadratic relaxations i: The gaussian model," *arXiv preprint arXiv:1907.08880*, 2019.

[14] Luca Ganassali, Marc Lelarge, and Laurent Massoulié, "Spectral alignment of correlated gaussian random matrices," *arXiv preprint arXiv:1912.00231*, 2019.

[15] Mohsen Bayati, David F Gleich, Amin Saberi, and Ying Wang, "Message-passing algorithms for sparse network alignment," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 7, no. 1, pp. 3, 2013.

[16] Serena Bradde, Alfredo Braunstein, Hamed Mahmoudi, Francesca Tria, Martin Weigt, and Riccardo Zecchina, "Aligning graphs and finding substructures by a cavity approach," *EPL (Europhysics Letters)*, vol. 89, no. 3, pp. 37009, 2010.

[17] Facundo Memoli, "On the use of Gromov-Hausdorff Distances for Shape Comparison," in *Eurographics Symposium on Point-Based Graphics*, M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, Eds. 2007, The Eurographics Association.

[18] L Susan Blackford, Antoine Petitet, Roldan Pozo, Karin Remington, R Clint Whaley, James Demmel, Jack Dongarra, Iain Duff, Sven Hammarling, Greg Henry, et al., "An updated set of basic linear algebra subprograms (blas)," *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 135–151, 2002.

[19] Harold W Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[20] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert, "A path following algorithm for the graph matching problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2227–2242, 2008.

[21] Vince Lyzinski, Donniell E Fishkind, Marcelo Fiori, Joshua T Vogelstein, Carey E Priebe, and Guillermo Sapiro, "Graph matching: Relax at your own risk," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 60–73, 2016.

[22] Shinji Umeyama, "An eigendecomposition approach to weighted graph matching problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 695–703, 1988.