

部分構造類似性の準最適解探索法の提案

Partial Graph Matching by Enumerating Pseudo-Cliques in a Common Space

ジェイ 泓杰^{1*} 原口 誠¹
HongJie Zhai¹ Makoto Haraguchi¹

¹ 北海道大学 情報科学研究科

¹ Graduate School of Information Science and Technology, Hokkaido University

Abstract: We proposed a new method for partial graph matching based on non-negative matrix factorization (NMF) and pseudo-clique mining. Compared with existing researches, it can find several potential matching solutions with different structural and attribute similarities. To achieve this purpose, we firstly map all the vertices of graphs into a common space by a modified non-negative matrix factorization. To find several subspaces in which local similarities between graphs can be observed, we apply pseudo-clique enumerator in this common space. As the similarity between graphs can be a global similarity in each subspace, standard global graph matching techniques is enough to cut off useless vertices and to obtain a targeted similarity as a result of partial graph matching. Our contributions in this research are that: Firstly, we presented a new framework for solving partial graph matching problem. Secondly, we designed a new class of NMF for finding common space. Furthermore, our experiments on both images and text data-sets reveal that our formulation works well on different kinds of data.

1 Introduction

Partial Graph Matching In computer science, Graph Matching (GM) is an important research theme that can be used for formalizing many problems in data mining and signal processing[Berg et al., 2005; Mateus et al., 2008; Brendel and Todorovic, 2011; Zaslavskiy et al., 2009]. In past several decades, the GM problem has been extensively researched in both practical and theoretical aspect[Zhou and De la Torre, 2012; Livi and Rizzi, 2013]. However, most of these researches focus on the matching problem between two graphs with the same number of vertices (i.e. globally matching)[Livi and Rizzi, 2013]. Although there are some works suppose that one side of graphs can only use part of its vertices (i.e. subgraph matching), there are only a little researches focusing on finding common subgraphs. In real-world problems, given two graphs, there usually does not exist a global matching. For example, when we consider the matching problem of CAD models in Figure 1, only a small part of these two model are similar. To apply the existing graph

matching techniques, it is necessary to manually mark the parts (candidates) to be matched. When the data becomes larger, manually marking becomes impractical. Thus, it is necessary to develop an automatical subgraph to subgraph matching algorithm. In this paper, we call this subgraph to subgraph matching problem the **Partial Graph Matching (PGM)**¹. Furthermore, in many cases, there exists more than one possible partial matchings between two data. In the mean time, some possible matching can only be found under some specific aspects. In the field of text mining, it is a very common phenomenon that the meaning of article differs regarding to the readers' aspect. Thus, for these real world graph matching problems, we need not only a subgraph to subgraph matching algorithm but also an algorithm which can find solutions from different aspects.

Technically speaking, the PGM problem to be considered in this paper can be formalized as the following form:

*連絡先：北海道大学 情報科学研究科 情報理工学専攻
〒060-0814 札幌市北区北14条西9丁目
E-mail: zhaihj@kb.ist.hokudai.ac.jp

¹In many papers, subgraph to subgraph matching is called Common Subgraph Isomorphism. However, to make a clear distinction from the subgraph matching problem, we use the term *Partial Graph Matching*.

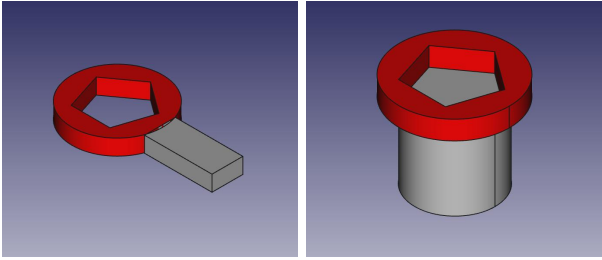


Figure. 1: An example of the partial match problem. The red area represents the partial structure expected to be matched.

$$PS_1G_1S_2^T P^T \approx S_2G_2S_2^T \quad (1)$$

where G_1 and G_2 are the matrix representations (e.g. weighted adjacent matrix, gram matrix, etc.) of two given graphs. S_1 and S_2 are **Selection Matrixes (SM)**. SM is used to select a subset of vertices (i.e. a subset of rows and columns of matrix representations) from the whole graph. P is the permutation matrix, which represents the matched pairs of vertices. The whole equation 1 means that we find a subgraph from G_1 whose permuted matrix representation is approximately equal to a subgraph from G_2 . Additionally, even though G_1 and G_2 may have different number of vertices, the size of matched subgraphs should be the same. In equation 1, when comparing the left and right terms, we use approximately equal instead of equal. The reason is that for real-world data, it is difficult to find two subgraphs which can be matched perfectly. We will discuss more details about partial graph matching in later sections.

2 Previous Works

Although most researches of GM are focus on global matching or subgraph matching, there are many common concepts between the PGM problem and global matching or subgraph matching problem. Thus, we will give a brief review on previous works on PGM as well as global matching and subgraph matching. The global matching problem can be written in the similar form with PGM:

$$PG_1P^T \approx G_2 \quad (2)$$

where P is the permutation matrix and G_1, G_2 are the matrix representations of graphs. This formulation is called the Koopmans-Beckmann's quadratic

assignment problem. From the viewpoint of optimization, combinatorially optimizing QAP of global matching is NP-hard. Thus, recent works on global matching are mainly focused on developing better approximate algorithm under some relaxations. The basic idea of relaxation is replacing the permutation matrix P with a real-valued matrix while trying to keep some key features of P in the mean time. For example, [Umeyama, 1988; Shapiro and Brady, 1992; Caelli and Kosinov, 2004] approximate the permutation matrix with an orthogonal matrix. [Ding et al., 2008] developed an algorithm based on non-negative matrix factorization (NMF) by relaxing the permutation matrix to a non-negative orthogonal matrix. The more general formalization, i.e. Lawler's quadratic assignment problem, can also be approximated by the similar strategy.

Comparing to global matching, the subgraph matching problem is much more difficult. In the subgraph matching problem, the given graphs can have different size. The target of subgraph matching is to find a subgraph from bigger graphs which globally matches the smaller one. In terms of the formula, the subgraph matching can be written as:[Tao and Wang, 2016]

$$PSG_1S^T P^T \approx G_2 \quad (3)$$

In this equation, the graph G_1 is bigger than G_2 . Same with the PGM problem, P is the permutation matrix and S is the selection matrix. The difference between subgraph matching and partial matching is that in subgraph matching, we only need to find the selection matrix for one side. [Ullmann, 1976] firstly proposed an exact algorithm for subgraph matching. After that, much effort has been made in combinatorial methods[Zampelli et al., 2005; Batz, 2006]. For the approximate side, [Tao and Wang, 2016] has done a good work. They use the orthogonal relaxation of permutation matrix and develop a optimization framework based on gradient flow.

For the partial graph matching problem, which is also known as the common subgraph problem, most methods are based on the combinatorial technique. Many of these algorithms[Raymond and Willett, 2002; Krissinel and Henrick, 2004; Grindley et al., 1993] are designed for simple graph without weights for vertices or edges. Although [Bunke, 1997] and [Kriege and Mutzel, 2012] provides the techniques for finding partial matching of labeled graphs, their algorithms need to work in the product space. Generally speaking,

the size of product space will increase greatly with respect to the size of graph. Thus, the algorithms based on product space can only work on very small graphs (e.g. less than 100 vertices). Moreover, as we have discussed previously, in PGM, there usually exists more than one possible matching solutions. Combinatorial techniques are suitable for finding all possible solutions but they suffer the efficiency problem. The approximating techniques from global matching can only find one local optimized solution. Thus, to find multiple partial matching between large graphs (tens of thousands of vertices), it is necessary to adopt a new strategy.

3 Proposal and Contribution

In this paper, we propose a novel framework for approximating the partial graph matching for weighted graphs. Our framework only requires searching in the union space of two graph instead of the big product space. Hence, our framework can be applied on large graphs. Furthermore, our framework is able to find multiple partial graph matching solutions at once. Additionally, this framework can be easily extended for variant types of graphs. As the result, we can get a union view of the partial graph matching problem over different types of data. Figure 2 shows the general flow of our framework: First, we map the given graphs into a common space by a non-negative matrix factorization (NMF) algorithm. The NMF algorithm will map the vertices which have similar features to the similar positions. Second, we build a union graph in this common space based on the euclid distances. Third, by enumerating pseudo-cliques in this union graph, we find several subspaces (i.e. pseudo-cliques) which contains the possible partial matching solutions. Each subspace represents one viewpoint of similarity. Finally, in each subspace, we do a global matching or subgraph matching with existing techniques to cut off the vertices which can not be matched. Because the non-negative matrix factorization algorithm usually works efficiently and the global matching or subgraph matching only happens in a small subspace, we can get good partial matching solutions without spending too much time. Our contribution can be summarized as that: 1. We developed a non-negative matrix factorization for partial graph matching. 2. We discuss the relations between pseudo-clique enumeration and graph matching

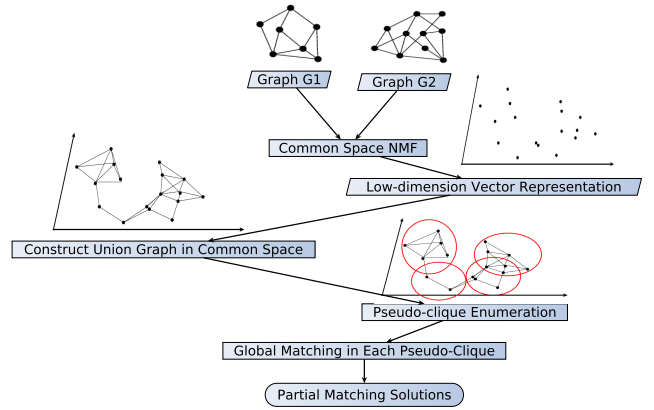


Figure 2: General Flow of Partial Graph Matching Framework

as well as show the potential of combinatorial techniques for efficient subspace searching in approximate algorithms. 3. We show the relation between partial graph matching and existing non-negative matrix factorization algorithms. 4. We experimentally show that our framework works well for different types of data, including artificial dataset, images and text.

4 Terminology Definitions

In this paper, we consider the following type of graph as input data:

Definition 1. Weighted Graph A weighted graph is a tuple $G = (V, E, F)$, where

- V^G is the set of vertices.
- $E^G \subseteq V \times V$ is the set of edges.
- $F^G : V \times V \rightarrow R^1$ is the similarity function which measures the similarity between two vertices.

By given a graph G , we use v_i to represent the i -th vertex and $e_{i,j}$ to represent the edge between i -th and j -th vertex. If v_i and v_j are connected, $e_{i,j}$ is 1, or it will be 0. The notions $|V^G|$ and $|E^G|$ denote the numbers of vertices and edges. $G(\mathcal{V})$ represents the subgraph of G from the vertices set \mathcal{V} .

Definition 2. Weighted Adjacent Matrix (WAM) A weighted graph G is represented by a matrix A^G , where

$$a_{i,j} = s_{i,j} * e_{i,j} \tag{4}$$

Just like many other researches, we define the matrix representation of graph as the weighted adjacent matrix. If two vertices are connected, their similarity comes from the similarity function. In other cases, the similarity becomes 0. With this type of data definition, the partial graph matching problem is formalized as follows:

Definition 3. Partial Graph Matching For two given graphs G_1 and G_2 , the solution of partial graph matching is a tuple of integer-valued matrix (S_1, S_2, P) , where

- $PS_1A^{G_1}S_1^TP^T \approx S_2A^{G_2}S_2^T$
- $PP^T = P^TP = I, S_1S_1^T = I, S_2S_2^T = I$

P is the permutation matrix, S_1 and S_2 are called selection matrixes, which are used to select a subset of vertices from given graphs. The selection matrix is proposed as *Translation Matrix* by [Tao and Wang, 2016]. Here we use almost the same definition with [Tao and Wang, 2016]:

Definition 4. Selection Matrix Given a graph G , SAS^T gives a weighted adjacent matrix of the sub-graph $G(\{v_{i_1}, v_{i_2}, v_{i_3}, \dots, v_{i_j}, \dots, v_{i_m}\})$, where

- $S \in R^{m \times |V^G|}$
- $S = (s_{c,k}) = \begin{cases} 1 & i_c = k \\ 0 & \text{otherwise} \end{cases}$

It is obvious that the selection matrix has the property $SS^T = I$.

5 Algorithms for Partial Graph Matching

Under the definitions in Section 4, we can transform the formulation of partial graph matching in Definition 3 to:

$$PS_1A^{G_1}S_1^T \approx S_2A^{G_2}S_2^TP \quad (5)$$

To solve this problem, we consider the following relaxation:

$$\begin{aligned} S_1 &\in R^{B \times |V^{A^{G_1}}|}, S_2 \in R^{A \times |V^{A^{G_2}}|}, \\ P &\in R^{A \times B}, PP^T = P^TP = I, SS^T = I, \quad (6) \\ P &\geq 0, S_1 \geq 0, S_2 \geq 0 \end{aligned}$$

Finally the formulation in Definition 3 can be transformed to the following problem:

Definition 5. Relaxed Partial Graph Matching (RPGM) Given two graph G_1 and G_2 , the solution of relaxed partial graph matching is a tuple of non-negative matrix $(N_1^\dagger, M_1^\dagger, N_2^\dagger, M_2^\dagger)$, where:

$$\begin{aligned} A^{G_1} &\approx N_1^\dagger C M_1^\dagger \\ A^{G_2} &\approx N_2^\dagger C M_2^\dagger \\ N_1^\dagger N_1^{\dagger T} &= I, N_2^\dagger N_2^{\dagger T} = I, \\ M_2^{\dagger T} M_2^\dagger &= I, M_2^{\dagger T} M_2^\dagger = I, \end{aligned} \quad (7)$$

We notice that because $N_{\{1,2\}}$ and $M_{\{1,2\}}$ are all non-negative, this RPGM problem can be solved by the non-negative matrix factorization algorithm. From the viewpoint of NMF, C is the common space shared by A^{G_1} and A^{G_2} . Each column of $M_{\{1,2\}}$ is the vector representations for the corresponded vertex. $N_{\{1,2\}}$ are the matrixes that transform the common space to the space of each graph (i.e. G_1, G_2). By using the frobenius norm formulation of NMF, the objective function becomens:

$$\begin{aligned} \min_{N_1 \geq 0, M_1 \geq 0, N_2 \geq 0, M_2 \geq 0} & |A^{G_1} - N_1^\dagger C M_1^\dagger|_F + \\ & |A^{G_2} - N_2^\dagger C M_2^\dagger|_F \\ \text{s.t.} & \\ N_1^\dagger N_1^{\dagger T} &= I, N_2^\dagger N_2^{\dagger T} = I, \\ M_2^{\dagger T} M_2^\dagger &= I, M_2^{\dagger T} M_2^\dagger = I, \end{aligned} \quad (8)$$

Moreover, we can easily derive the following update rules for RPGM by following the steps in [Lee and Seung, 2001]:

Algorithm 1. Update Rule of RPGM

$$M_{1jk}^\dagger \leftarrow M_{1jk}^\dagger \frac{(C^T N_1^{\dagger T} A)_{jk}}{(C^T N_1^{\dagger T} N_1^\dagger C M_1^\dagger)_{jk}} \quad (9)$$

$$M_{2jk}^\dagger \leftarrow M_{2jk}^\dagger \frac{(C^T N_2^{\dagger T} A)_{jk}}{(C^T N_2^{\dagger T} N_2^\dagger C M_2^\dagger)_{jk}} \quad (10)$$

$$N_{1jk}^\dagger \leftarrow N_{1jk}^\dagger \frac{(A^{G_1} M_1^\dagger C^T)_{jk}}{(N_1^\dagger C M_1^\dagger M_1^{\dagger T} C^T)_{jk}} \quad (11)$$

$$N_{2jk}^\dagger \leftarrow N_{2jk}^\dagger \frac{(A^{G_2} M_2^\dagger C^T)_{jk}}{(N_2^\dagger C M_2^\dagger M_2^{\dagger T} C^T)_{jk}} \quad (12)$$

$$C_{jk} \leftarrow C_{jk} \frac{(N_1^{\dagger T} A^{G_1} M_1^{\dagger T} + N_2^{\dagger T} A^{G_2} M_2^{\dagger T})_{jk}}{(N_1^{\dagger T} N_1^\dagger C M_1^\dagger M_1^{\dagger T} + N_2^{\dagger T} N_2^\dagger C M_2^\dagger M_2^{\dagger T})_{jk}} \quad (13)$$

By viewing C as the common space, M_1 and M_2 are the vector representations in this common space. If two vectors (vertices) $v_1 \in G_1$ and $v_2 \in G_2$ are close in the common space, they have the chance to be matched. Thus, we can know the clusters in common space represent the possible solution sets of RPGM problem. Because the matched parts of graphs are empirically highly overlapped, we adopt the pseudo-clique model for detecting clusters. Each pseudo-clique represents one subspace that contains a possible PGM solution. As the result, we can apply the global matching algorithms in each pseudo-clique to find the final matching solution. Our algorithm is summarized as follows:

- 1: **procedure** PARTIAL GRAPH MATCHING(G_1, G_2, σ)
- 2: Randomly initialize N_1, N_2, M_1, M_2, C .
- 3: **repeat**
- 4: Update M_1, M_2 with Equation 9,10 by fixing N_1, N_2, C .
- 5: Update N_1, N_2 with Equation 11,12 by fixing M_1, M_2, C .
- 6: Update C with Equation 13 by fixing N_1, N_2, M_1, M_2 .
- 7: **until** Equation 8 is Convergence
- 8: Treat each column of M_1 and M_2 as a new vector of corresponded vertex, calculate euclid distance $d(v_i, v_j)$ between all pairs of vertices.
- 9: $V^{G_C} \leftarrow V^{G_1} \cup V^{G_2}$
- 10: $E^{G_C} \leftarrow \{e_{ij} | d(v_i, v_j) < \sigma\}$ ▷ If two vertices are close in the common space, they are connected.
- 11: Build new graph $G^C(V^{G_C}, E^{G_C})$.
- 12: Perform pseudo-clique searching in G^C .
- 13: **for** Each pseudo-clique PC **do**
- 14: Perform global graph matching algorithm in PC .
- 15: **end for**
- 16: **end procedure**

6 Experiment

To demonstrate the ability and performance of the RPGM algorithm, we applied our algorithm on several types of data, including the artificial dataset, CAD data[Tao and Wang, 2016] and images[Zhou and De la Torre, 2012]. We will report the details of experiment in the presentation.

7 Concluding Remarks and Future Works

In this paper, we presented a novel framework for the partial graph matching problem. We showed that by some relaxation, the RPGM can be transformed to a non-negative matrix factorization problem. Furthermore, to find multiple solutions, we proposed an efficient subspace search approach based on the pseudo-clique enumeration. Under the unified view of PGM, NMF and pseudo-clique model, the efficient PGM algorithm is achieved.

Because of the limitation of time, several questions remain to be investigated in future:

- In this paper, we consider the input data as the weighted adjacent matrix. However, in many cases, we can build the graph and calculate the similarity between vertices separately. For example, by given a set of texts, we can build the co-occurrence graph of words. For each pair of words (vertices), the semantic similarity can be calculated from the dictionary. In this situation, using weighted adjacent matrices will lead to the loss of information. Hence, it is more natural to consider the structure similarity and semantic similarity separately.
- When building the union graph in the common space, a parameter σ is used to control connectedness. Generally speaking, selecting parameters is a difficult task. Thus, we need to provide a guidance for the σ .
- We used the pseudo-clique for enumerating the matching subspaces. The relation between pseudo-clique enumeration and graph matching still remained to be investigated. Moreover, because there exists variant pseudo-clique algorithms, we need to characterize these pseudo-clique algorithms from the viewpoint of partial graph matching.

References

- Batz, G. V. (2006). *An optimization technique for subgraph matching strategies*. Univ., Fak. für Informatik, Bibliothek.
- Berg, A. C., Berg, T. L., and Malik, J. (2005). Shape matching and object recognition using low distor-

- tion correspondences. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 26–33. IEEE.
- Brendel, W. and Todorovic, S. (2011). Learning spatiotemporal graphs of human activities. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 778–785. IEEE.
- Bunke, H. (1997). On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689–694.
- Caelli, T. and Kosinov, S. (2004). An eigenspace projection clustering method for inexact graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(4):515–519.
- Ding, C., Li, T., and Jordan, M. I. (2008). Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 183–192. IEEE.
- Grindley, H. M., Artymiuk, P. J., Rice, D. W., and Willett, P. (1993). Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *Journal of molecular biology*, 229(3):707–721.
- Kriege, N. and Mutzel, P. (2012). Subgraph matching kernels for attributed graphs. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1015–1022.
- Krissinel, E. B. and Henrick, K. (2004). Common subgraph isomorphism detection by backtracking search. *Software: Practice and Experience*, 34(6):591–607.
- Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562.
- Livi, L. and Rizzi, A. (2013). The graph matching problem. *Pattern Analysis and Applications*, 16(3):253–283.
- Mateus, D., Horaud, R., Knossow, D., Cuzzolin, F., and Boyer, E. (2008). Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.
- Raymond, J. W. and Willett, P. (2002). Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of computer-aided molecular design*, 16(7):521–533.
- Shapiro, L. S. and Brady, J. M. (1992). Feature-based correspondence: an eigenvector approach. *Image and vision computing*, 10(5):283–288.
- Tao, S. and Wang, S. (2016). An algorithm for weighted sub-graph matching based on gradient flows. *Information Sciences*.
- Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42.
- Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(5):695–703.
- Zampelli, S., Deville, Y., and Dupont, P. (2005). Approximate constrained subgraph matching. In *Principles and Practice of Constraint Programming-CP 2005*, pages 832–836. Springer.
- Zaslavskiy, M., Bach, F., and Vert, J.-P. (2009). Global alignment of protein-protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):i259–1267.
- Zhou, F. and De la Torre, F. (2012). Factorized graph matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 127–134. IEEE.