Graph Matching Formulations

Romain Raveaux

LI lab (EA 6300) – Université de Tours

October 13, 2016

International Master of Research in Computer Science: Computer Aided Decision Support

Laboratoire d'Informatique
EA 6300

UNIVERSITÉ
FRANÇOIS - RABELAIS

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
    - Machine Learning
    - Pattern Recognition
    - Computer Vision
    - Machine Learning with graphs
    - Graph Matching
    - Graph-Based representation
    - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
    - Machine Learning
    - Pattern Recognition
    - Computer Vision
    - Machine Learning with graphs
    - Graph Matching
    - Graph-Based representation
    - Image analysis

About Me I

- Romain Raveaux
- Teacher at the university of Tours. Polytech'Tours
- Researcher at the Computer Science Laboratory
- Researcher Activity:
  - Machine Learning
  - Pattern Recognition
  - Computer Vision
  - Machine Learning with graphs
  - Graph Matching
  - Graph-Based representation
  - Image analysis

About Me II

- Main publications : Referenced International Journal
  - Romain Raveaux et al. Structured representations in a content based image retrieval context. Journal of Visual Communication and Image Representation, Volume 24, Issue 8, November 2013, Pages 1252-1268.
  - Zeina Abu-Aisheh et al. Anytime graph matching. Pattern Recognition Letters. In press (2016).
  - Romain Raveaux et al. Learning graph prototypes for shape recognition. Computer Vision and Image Understanding 115(7): 905-918 (2011)
  - Romain Raveaux et al. A graph matching method and a graph matching distance based on subgraph assignments. Pattern Recognition Letters 31(5): 394-406 (2010)

About Me II

- Main publications : Referenced International Journal
  - Romain Raveaux et al. Structured representations in a content based image retrieval context. Journal of Visual Communication and Image Representation, Volume 24, Issue 8, November 2013, Pages 1252-1268.
  - Zeina Abu-Aisheh et al. Anytime graph matching. Pattern Recognition Letters. In press (2016).
  - Romain Raveaux et al. Learning graph prototypes for shape recognition. Computer Vision and Image Understanding 115(7): 905-918 (2011)
  - Romain Raveaux et al. A graph matching method and a graph matching distance based on subgraph assignments. Pattern Recognition Letters 31(5): 394-406 (2010)

About Me II

- Main publications : Referenced International Journal
  - Romain Raveaux et al. Structured representations in a content based image retrieval context. Journal of Visual Communication and Image Representation, Volume 24, Issue 8, November 2013, Pages 1252-1268.
  - Zeina Abu-Aisheh et al. Anytime graph matching. Pattern Recognition Letters. In press (2016).
  - Romain Raveaux et al. Learning graph prototypes for shape recognition. Computer Vision and Image Understanding 115(7): 905-918 (2011)
  - Romain Raveaux et al. A graph matching method and a graph matching distance based on subgraph assignments. Pattern Recognition Letters 31(5): 394-406 (2010)

About Me II

- Main publications : Referenced International Journal
  - Romain Raveaux et al. Structured representations in a content based image retrieval context. Journal of Visual Communication and Image Representation, Volume 24, Issue 8, November 2013, Pages 1252-1268.
  - Zeina Abu-Aisheh et al. Anytime graph matching. Pattern Recognition Letters. In press (2016).
  - Romain Raveaux et al. Learning graph prototypes for shape recognition. Computer Vision and Image Understanding 115(7): 905-918 (2011)
  - Romain Raveaux et al. A graph matching method and a graph matching distance based on subgraph assignments. Pattern Recognition Letters 31(5): 394-406 (2010)

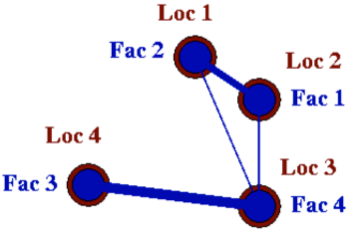
About Me II

- Main publications : Referenced International Journal
  - Romain Raveaux et al. Structured representations in a content based image retrieval context. Journal of Visual Communication and Image Representation, Volume 24, Issue 8, November 2013, Pages 1252-1268.
  - Zeina Abu-Aisheh et al. Anytime graph matching. Pattern Recognition Letters. In press (2016).
  - Romain Raveaux et al. Learning graph prototypes for shape recognition. Computer Vision and Image Understanding 115(7): 905-918 (2011)
  - Romain Raveaux et al. A graph matching method and a graph matching distance based on subgraph assignments. Pattern Recognition Letters 31(5): 394-406 (2010)

QAP (Koopmans and Beckman in 1957)

Summary: The objective of the Quadratic Assignment Problem (QAP) is to assign nn facilities to nn locations in such a way as to minimize the assignment cost. The assignment cost is the sum, over all pairs, of the flow between a pair of facilities multiplied by the distance between their assigned locations.

QAP (Koopmans and Beckman in 1957)



QAP (Koopmans and Beckman in 1957)

Flows between facilities

facility i	facility j	flow(i, j)
1	2	3
1	4	2
2	4	1
3	4	4

Distances between locations

location i	location j	distance(i, j)
1	3	53
2	1	22
2	3	40
3	4	55

QAP (Koopmans and Beckman in 1957)

Sets

$N = \{1, 2, \cdots, n\}$

$S_n = \phi : N \to N$ is the set of all permutations

Parameters

$F = (f_{ij})$ is an $n \times n$ matrix where f_{ij} is the required flow between facilities i and j

$D = (d_{ij})$ is an $n \times n$ matrix where d_{ij} is the distance between locations i and j

Optimization Problem

$\min_{\phi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot d_{\phi(i)\phi(j)}$

Unless $P = NP$, the problem is NP-hard, so there is no known algorithm for solving this problem in polynomial time, and even small instances may require long computation time. Unlike the linear assignment problem, which can be efficiently solved with the Hungarian algorithm, the Quadratic Assignment Problem is known to be NP-hard. Therefore, the main body of research in GM has focused on devising more accurate algorithms to solve it approximately. Nevertheless, approximating GM by relaxing the combinatorial constraints is still a challenging problem. This is mainly because the objective function is in general non-convex and thus existing methods are prone to local optima.

Minimisation problem in [1]

Given two attributed graphs $G = (V, E, \mu_G, \zeta_G)$ and $Q = (U, T, \mu_Q, \zeta_Q)$. Suppose a match between the two graphs is represented by a matrix M such that$m_{ij} = 1$ if $v_i \in V$ is matched to $u_j \in U$ and 0 otherwise. The optimal solution to the inexact graph matching problem is the matrix M^* such that:

Minimisation problem

Definition

Graph Matching Problem

$$
\begin{aligned}
M^* \quad = \quad & arg \min_M \sum_i \sum_j f(\mu_G(v_i), \mu_Q(u_j)) m_{ij} + \\
& \sum_i \sum_j \sum_{i'} \sum_{j'} f(\zeta_G(e_{ii'}), \zeta_Q(t_{jj'})) m_{ij} m_{i'j'} \\
& subjected\ to\ the\ constraints: \\
& \forall u_j \sum_i m_{ij} \leq 1 \\
and \quad & \forall v_i \sum_j m_{ij} \leq 1
\end{aligned}
$$

Minimisation problem in [1]

Where $f(v_i, u_j)$ is the distance function between two vertices $v_i \in V$ and $u_j \in U$.
Where $f(e_{ii'}, t_{jj'})$ is the distance function between the edges $e_{ii'} = (v_i, v_{i'}) \in E$ and $t_{jj'} = (u_j, u_{j'}) \in T$.
The constraints in the above equation require that a vertex $v_i \in V$ can only be matched to a vertex $u_j \in U$, and vice versa. Also, v_i may have no correspondent vertex from U. In such a case, v_i is considered an outlier vertex. This applies also to the vertices from U. Based on Definition 1, a solution to the inexact graph matching problem solves both **subgraph** and **common subgraph** matching.

Maximisation problem in [2]

A solution of graph matching is defined as a subset of possible correspondences $y \subset V_1 \times V_2$, represented by a binary assignment matrix $Y \in \{0,1\}^{n_1 \times n_2}$, where n_1 and n_2 denote the number of nodes in V_1 and V_2, respectively. If $v_{1,i} \in V_1$ matches $v_{2,a} \in V_2$, then $Y_{i,a} = 1$, and $Y_{i,a} = 0$ otherwise. We denote by $y \in \{0,1\}^{n_1 n_2}$, a column-wise vectorized replica of Y. With this notation, graph matching problems can be expressed as finding the assignment vector y^* that maximizes a score function $S(G_1, G_2, y)$ as follows:

Maximisation problem in [2]

Graph Matching formulation

$$y^* = \operatorname*{argmax}_{y} \quad S(G_1, G_2, y), \tag{1a}$$

$$\text{subject to} \quad y \in \{0,1\}^{n_1 n_2} \tag{1b}$$

$$\sum_{i=1}^{n_1} y_{i,a} \leq 1 \quad \forall a \in [1, \cdots, |V_2|] \tag{1c}$$

$$\sum_{a=1}^{n_2} y_{i,a} \leq 1 \quad \forall i \in [1, \cdots, |V_1|] \tag{1d}$$

Maximisation problem in [2]

The score function $S(G, G', y)$ measures the similarity of graph attributes, and is typically decomposed into a first order similarity function $s_V(a_i, a'_a)$ for a node pair $v_i \in V and v'_a \in V'$, and a second-order similarity function $s_E(a_{ij}, a'_{ab})$ for an edge pair $e_{ij} \in E$ and $e'_{ab} \in E'$. Similarity functions are usually represented by a symmetric similarity matrix A, where a non-diagonal element $A_{ia;jb} = s_E(a_{ij}, a'_{ab})$ contains the edge similarity of two correspondences (v_i, v'_a) and (v_j, v'_b) and a diagonal term $A_{ia;ia} = s_V(a_i, a'_a)$ represents the node similarity of a correspondence (v_i, v'_a).
Thus, the score function of graph matching is defined as:

Maximisation problem in [2]

Score function

$$S(G, G', y) = \sum_{y_{ia}=1} s_V(a_i, a'_a) + \sum_{y_{ia}=1} \sum_{y_{jb}=1} s_E(a_{ij}, a'_{ab}) \tag{2a}$$

$$= y^T A y \tag{2b}$$

Maximisation problem in [2]

In essence, the score accumulates all the similarity values relevant to the assignment. The formulation in Eq. 1a is referred to as an integer quadratic programming. More precisely, it is the quadratic assignment problem, which is known to be NP-hard. Due to its generality and flexibility, this formulation has been favored in recent graph matching research. Many efficient approximate algorithms have been proposed for the formulation

Graph Edit Distance as Quadratic Assignment Problem [3]

In order to reformulate GED as QAP, two challenging points have been considered. First, having equal cardinality matrices taking into account the unequal cardinality of vertices and edges in the involved graphs of GED. Second, GED is more general than QAP since it does not necessarily assign each vertex or edge in G_1 to a vertex or an edge in G_2. That is, GED also allows the deletion of vertices and edges of G_1 as well as the insertion of vertices and edges of G_2.

Graph Edit Distance as Quadratic Assignment Problem [3]

These issues have been solved by adding empty vertices and so edges in the list of vertices, as stated here:

$V_1^{\Delta} = V_1 \cup \{\epsilon_1, \epsilon_2, \cdots \epsilon_m\}$

$V_2^{\Delta} = V_2 \cup \{\epsilon_1, \epsilon_2, \cdots \epsilon_n\}$

Graph Edit Distance as Quadratic Assignment Problem [3]

The adjacency matrices of G_1 and G_2 (i.e., A and B respectively) are defined as follows:

$$A_{(n+m)\times(n+m)} = \left|\begin{array}{cccc|cccc} a_{11} & \dots & \dots & a_{1n} & \epsilon & \epsilon & \dots & \epsilon \\ \dots & \dots & \dots & \dots & \epsilon & \epsilon & \dots & \epsilon \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nn} & \epsilon & \dots & \epsilon & \epsilon \\ \hline \epsilon & \epsilon & \dots & \epsilon & 0 & \dots & \dots & 0 \\ \epsilon & \epsilon & \dots & \epsilon & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \epsilon & \dots & \epsilon & \epsilon & 0 & \dots & \dots & 0 \end{array}\right|$$

Graph Edit Distance as Quadratic Assignment Problem [3]

The adjacency matrices of G_1 and G_2 (i.e., A and B respectively) are defined as follows:

$$B_{(n+m)\times(n+m)} = \left|\begin{array}{cccc|cccc} b_{11} & \dots & \dots & b_{1m} & \epsilon & \epsilon & \dots & \epsilon \\ \dots & \dots & \dots & \dots & \epsilon & \epsilon & \dots & \epsilon \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_{m1} & \dots & \dots & b_{mm} & \epsilon & \dots & \epsilon & \epsilon \\ \hline \epsilon & \epsilon & \dots & \epsilon & 0 & \dots & \dots & 0 \\ \epsilon & \epsilon & \dots & \epsilon & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \epsilon & \dots & \epsilon & \epsilon & 0 & \dots & \dots & 0 \end{array}\right|$$

Graph Edit Distance as Quadratic Assignment Problem [3]

Based on V_1^Δ and V_2^Δ, the cost matrix C can be established as follows:

$$C_{(n+m)\times(n+m)} = \left|\begin{array}{cccc||cccc}
c_{1,1} & \dots & \dots & c_{1,m} & c_{1,\epsilon} & \infty & \dots & \infty \\
\dots & \dots & \dots & \dots & \infty & c_{2,\epsilon} & \dots & \infty \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
c_{n,1} & \dots & \dots & c_{n,m} & \infty & \dots & \infty & c_{n,\epsilon} \\
\hline\hline
c_{\epsilon,1} & \infty & \dots & \infty & 0 & \dots & \dots & 0 \\
\infty & c_{\epsilon,2} & \dots & \infty & \dots & \dots & \dots & \dots \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\infty & \dots & \infty & c_{\epsilon,m} & 0 & \dots & \dots & 0
\end{array}\right|$$

Graph Edit Distance as Quadratic Assignment Problem [3]

$$d_{min} = \min_{(\varphi_1,\varphi_2,\cdots\varphi_{n+m})\in\Pi_{n+m}} \sum_{i=1}^{n+m} c_{i\varphi(i)} + \sum_{i=1}^{n+m}\sum_{j=1}^{n+m} c(a_{ij} \rightarrow b_{\varphi_i\varphi_j}) \quad (3)$$

where Π_{n+m} refers to the set of all $(n+m)!$ possible permutations of the integers $1, 2, \cdots, (n+m)$. The first linear term $\sum_{i=1}^{n+m} c_{i\varphi(i)}$ is dedicated to the sum of vertex edit costs while the second quadratic term $\sum_{i=1}^{n+m}\sum_{j=1}^{n+m} c(a_{\varphi_i\varphi_j} \rightarrow b_{\varphi_i\varphi_j})$ refers to the underling edge cost resulted from the permutation $(\varphi_1, \varphi_2, \cdots \varphi_{n+m})$. For instance, if vertex $u_i \in V_1^\Delta$ is matched with vertex $v_k \in V_2^\Delta$ and vertex $u_j \in V_1^\Delta$ is matched with vertex $v_z \in V_2^\Delta$, then edge $e(u_i, u_j)$ has to be matched with edge $e(v_k, v_z)$. These edges are kept in a_{ij} and b_{kz}, respectively. As previously mentioned, edges might be empty.

Graph Edit Distance as Quadratic Assignment Problem [4]

A binary linear programming formulation of GED is proposed in [4]. GED between graphs is treated as finding a subgraph of a larger graph referred to as the edit grid. The edit grid only needs to have as many vertices as the sum of the total number of vertices in the graphs being compared. The edit grid is a complete graph $G_\Omega = (\Omega, \Omega X \Omega, \mu_\Omega)$ where Ω denotes a set of vertices with N elements. Accordingly, $\Omega X \Omega$ is the set of undirected edges connecting all pairs of vertices. GED between $G_1 = (V_1, E_1, L_{V_1}, L_{E_1}, \mu_1)$ and $G_2 = (V_2, E_2, L_{V_2}, L_{E_2}, \mu_2)$ can be expressed by:

Graph Edit Distance as Quadratic Assignment Problem [4]

$$GED(G_1, G_2) = \min_{P \in \{0,1\}^{NXN}} \sum_{i=1}^{N} \sum_{j=1}^{N} c(l(A_1^i), l(A_2^j))P^{ij} + \frac{1}{2} c(0,1) |A_1 - PA_2P^T|^{ij} \quad (4)$$

where P is a permutation matrix representing all possible permutations of the elements of edit grid. $A_n \in \{0,1\}^{NXN}$ is the adjacency matrix corresponding to G_n in the edit grid. P represents N^2 boolean variables. N needs to be no larger than $|V_1| + |V_2|$ where $|V_1|$ and $|V_2|$ are the numbers of vertices of the involved graphs. $l(A_n^i)$ is the attribute assigned to the i^{th} row/column of A_n. Finally, the function c is a metric between two vertex attributes.

Binary Linear Programing

Binary Linear Programing is a restriction of integer linear programming (ILP) with binary variables. Its general form is :

$$\min_{x} \; c^T x \tag{5a}$$

$$\text{subject to } Ax \leq b \tag{5b}$$

$$x \in \{0,1\}^n \tag{5c}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^m$ are data of the problem.

Binary Linear Programing

A feasible solution is a vector x of n binary variables (5c) which respects linear inequality constraints (5b). The constraint (5c) which enumerates the admissible values of variables is called a domain constraint. If the program has at least a feasible solution, then the optimal solutions are the ones that minimize the objective function (5a) which is a linear combination of variables of x weighted by the components of the vector c.

Graph Edit Distance as Binary Linear Programing [4]

Formulation 4 is quadratic since it holds the product of P variables (PA_2P^T). In order to linearize it, two matrices, S and T, are introduced (inspired by [**?**]), and thus a binary linear formulation is obtained:

$$GED(G_1, G_2) = \min_{P,S,T \in \{0,1\}^{NXN}} \sum_{i=1}^{N} \sum_{j=1}^{N} c(l(A_1^i), l(A_2^j))P^{ij} + \frac{1}{2} c(0,1)(S+T)^{ij} \quad (6)$$

$$s.t. \begin{cases} (A_1P - PA_2 + S - T)^{ij} = 0, \ \forall i,j & (2.10.1) \\ \sum_i P^{ik} = \sum_j P^{kj} = 1, \forall k & (2.10.2) \end{cases}$$

Graph Edit Distance as Binary Linear Programing [4]

P, S and T represent $3 \times N^2$ boolean variables where N is the number of vertices $|V_1| + |V_2|$. Two types of constraints are applied to the objective function. In the first type of constraints, the GM problem is formulated as the minimization of the difference in adjacency matrix norms for unattributed graphs with the same number of vertices. The second constraints limit the set of acceptable permutation where one element of grid (i.e., vertex) must be permuted with exactly one element. There are N^2 constraints of type 1 and $2N$ constraints of type 2. Finally, the model is solved by a mathematical programming solver (lpsolve). One drawback of this method is that it does not take into account attributes on edges which limits the range of application.

Graph Edit Distance as Binary Linear Programing Lerouge et al SSPR 2016

For each type of edit operation, we define a set of corresponding binary variables:

- $\forall (i,k) \in V_1 \times V_2, x_{i,k} = \begin{cases} 1 \text{ if } i \text{ is substituted with } k, \\ 0 \text{ otherwise.} \end{cases}$
- $\forall (ij,kl) \in E_1 \times E_2, y_{ij,kl} = \begin{cases} 1 \text{ if } ij \text{ is substituted with } kl, \\ 0 \text{ otherwise.} \end{cases}$
- $\forall i \in V_1, u_i = \begin{cases} 1 \text{ if } i \text{ is deleted from } G_1 \\ 0 \text{ otherwise.} \end{cases}$
- $\forall ij \in E_1, e_{ij} = \begin{cases} 1 \text{ if } ij \text{ is deleted from } G_1 \\ 0 \text{ otherwise.} \end{cases}$
- $\forall k \in V_2, v_k = \begin{cases} 1 \text{ if } k \text{ is inserted in } G_1 \\ 0 \text{ otherwise.} \end{cases}$
- $\forall kl \in E_2, f_{kl} = \begin{cases} 1 \text{ if } kl \text{ is inserted in } G_1 \\ 0 \text{ otherwise.} \end{cases}$

Graph Edit Distance as Binary Linear Programing Lerouge et al SSPR 2016

The objective function is :

$$d(G_1, G_2) = \min_{\mathbf{x},\mathbf{y},\mathbf{u},\mathbf{v},\mathbf{e},\mathbf{f}} \Bigg(\sum_{i \in V_1} \sum_{k \in V_2} c(i \to k) \cdot x_{i,k} + \sum_{ij \in E_1} \sum_{kl \in E_2} c(ij \to kl) \cdot y_{ij,kl} \\ + \sum_{i \in V_1} c(i \to \epsilon) \cdot u_i + \sum_{k \in V_2} c(\epsilon \to k) \cdot v_k \\ + \sum_{ij \in E_1} c(ij \to \epsilon) \cdot e_{ij} + \sum_{kl \in E_2} c(\epsilon \to kl) \cdot f_{kl} \Bigg) \tag{7}$$

Graph Edit Distance as Binary Linear Programing Lerouge et al SSPR 2016

Vertices matching constraints

The constraint (8) ensures that each vertex of G_1 is either matched to exactly one vertex of G_2 or deleted from G_1, while the constraint (9) ensures that each vertex of G_2 is either matched to exactly one vertex of G_1 or inserted in G_1:

$$u_i + \sum_{k \in V_2} x_{i,k} = 1 \quad \forall i \in V_1 \tag{8}$$

$$v_k + \sum_{i \in V_1} x_{i,k} = 1 \quad \forall k \in V_2 \tag{9}$$

Graph Edit Distance as Binary Linear Programing Lerouge et al SSPR 2016

Edges matching constraints

Similar to the vertex matching constraints, the constraints (10) and (11) guarantee a valid mapping between the edges:

$$e_{ij} + \sum_{kl \in E_2} y_{ij,kl} = 1 \quad \forall ij \in E_1 \tag{10}$$

$$f_{kl} + \sum_{ij \in E_1} y_{ij,kl} = 1 \quad \forall kl \in E_2 \tag{11}$$

Graph Edit Distance as Binary Linear Programing Lerouge et al SSPR 2016

Topological constraints

In order to respect the graph topology in the matching, an edge $ij \in E_1$ can be matched to an edge $kl \in E_2$ only if the head vertices $i \in V_1$ and $k \in V_2$, on the one hand, and if the tail vertices $j \in V_1$ and $l \in V_2$, on the other hand, are respectively matched. This quadratic constraint can be expressed linearly with the following constraints (12) and (13):

- ij and kl can be matched if and only if their head vertices are matched:

$$y_{ij,kl} \leq x_{i,k} \quad \forall (ij, kl) \in E_1 \times E_2 \tag{12}$$

- ij and kl can be matched if and only if their tail vertices are matched:

$$y_{ij,kl} \leq x_{j,l} \quad \forall (ij, kl) \in E_1 \times E_2 \tag{13}$$

Conclusion

- There are many formulations for a given problem.
- So there are many solvers.
- Graph matching is closely related to :
  - QAP problem
  - MAP inference problem (maximum a posteriori probability)
- Progress on these problems can be applied to GM problems

Conclusion

The MAP-inference problem for a graphical model over an undirected graph $G = (\mathcal{V}, \mathcal{E})$, reads

$$\min_{x \in X_{\mathcal{V}}} E_{\mathcal{V}}(x) := \sum_{v \in \mathcal{V}} \theta_v(x_v) + \sum_{uv \in \mathcal{E}} \theta_{uv}(x_u, x_v), \quad (2.1)$$

where x_u belongs to a finite *label set* X_u for each node $u \in \mathcal{V}$, $\theta_u : X_u \to \mathbb{R}$ and $\theta_{uv} : X_u \times X_v \to \mathbb{R}$ are the *unary* and *pairwise potentials* associated with the nodes and edges of G. The label space for $A \subset \mathcal{V}$ is

Ayser Armiti and Michael Gertz.
Geometric graph matching and similarity: a probabilistic approach.
In Christian S. Jensen, Hua Lu, Torben Bach Pedersen, Christian Thomsen, and Kristian Torp, editors, *Conference on Scientific and Statistical Database Management, SSDBM '14, Aalborg, Denmark, June 30 - July 02, 2014*, pages 27:1–27:12. ACM, 2014.

Minsu Cho, Karteek Alahari, and Jean Ponce.
Learning graphs to match.
In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 25–32. IEEE Computer Society, 2013.

Kaspar Riesen.

Structural Pattern Recognition with Graph Edit Distance - Approximation Algorithms and Applications.
Advances in Computer Vision and Pattern Recognition. Springer, 2015.

Derek Justice and Alfred O. Hero III.
A binary linear programming formulation of the graph edit distance.
IEEE Trans. Pattern Anal. Mach. Intell., 28(8):1200–1214, 2006.