Multi-Attributed Graph Matching: a Spectral Clustering Approach

Timothee R. Cour  Jianbo Shi
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
{timothee,jshi}@seas.upenn.edu

Abstract

We present a Multiple Attribute Graph (MARG) Matching algorithm. The algorithm operates on a strong product graph, where vertices correspond to hypothesized matchings between two nodes and edges specify compatibilities between two matches. We show how the matching problem reduces to partitioning the strong product graph, which we solve with spectral clustering. We test our algorithm on a few images and handwritten digits, building a MARG for each image with spatial edge attributes.

1 Problem Setup

Multi-Attributed Graph (MARG)  We define a MARG[3] as a graph with multiple attributes attached to each edge and each vertex. Formally, we write an MARG as $G = (V, E, (W^r)_{r \in R}, (F^a)_{a \in A})$, where $V$ is the node set, $E$ is the edge set, $(W^r)_{r \in R}$ is a set of binary attributes defined for each edge $e = (i, j) \in E$, $(F^a)_{a \in A}$ is a set of unary attributes defined on each node. For example, the nodes could represent image regions, image edges, or text labels. Node attributes could represent their color/texture, position/orientation and other geometric properties, and edge attributes could represent logical relationships ("belong to", "Is a") or spatial relationships ("Is connected to", "Is South of"). Figure 1 illustrates this concept in the case of human body representation with both body parts and labels as graph nodes. Fuzzy relations can be captured with numeric values instead of boolean attributes. For example, if nodes $i$ and $j$ represent image regions located at $(x_i, y_i)$ and $(x_j, y_j)$, we can specify the algebraic displacement between $i$ and $j$ as $W^\text{translation}(i, j) = (x_j - x_i, y_j - y_i)$. Another example is $W^\text{label}(i, j) = P(j|i)$ if a classifier computes a probability distribution over labels $j$ given the observation node $i$. Note that in general, the graph constructed will be large, sparse and with nonsymmetric edge attributes.

MARG Matching Cost  Let $G = (V, E, (W^r)_{r \in R}, (F^a)_{a \in A})$, $G' = (V', E', (W'^r)_{r \in R}, (F'^a)_{a \in A})$ be two MARGs, where we temporarily assume that $n = |V| = |V'|$. We want to find a bijection from $V$ to $V'$ that will preserve edge and vertex attributes as much as possible. Equivalently, we want to find a permutation $\rho$ of order $n$ so as to minimize the multi-attribute graph matching error, defined as:
Figure 1: Multi-Attribute Relational Graph (MARG) representing a human body through its parts and their relations.

\[ \epsilon_1(p) = \sum_{ij} d_R^2(W_{ij}, W'_{p(i)p(j)}) + \sum_i d_A^2(F_i, F'_{p(i)}), \quad (1) \]

where \( d_R(\cdot, \cdot) \) (resp. \( d_A(\cdot, \cdot) \)) is a distance defined over edge attribute vectors (resp. vertex attribute vectors).

Euclidian MARG (E-MARG) Matching Cost If \( d_R \) and \( d_A \) are euclidian distances, we can simplify (1) via an equivalent permutation matrix \( P \in Perm(n) \) with \( P_{vi} = \delta_{i'p(i)} \):

\[ \epsilon_2(P) = \sum_{r \in R} \|W^r - PTW'^rP\|^2 + \sum_{a \in A} \|F^a - PT'F'^a\|^2, \quad \text{for} \ P \in Perm(n) \quad (2) \]

2 Single Attributed Graph Matching

We present here a brief review of two landmark methods to solve the graph matching problem in the case of single attribute graphs. Both are based computing singular vectors of some weight matrices related to the input graphs.

Umeyama (UME)[5] considers the single edge attribute case with euclidian distance, \(|R| = 1, |A| = 0\) in (2):

\[ P^* = \arg\min ||W - PTW'P||, \ \text{s.t.} \ P \in Perm(n), \quad (3) \]

To solve the problem, the discrete constraint on \( P \) is relaxed, noting that permutation matrices are a special case of orthogonal matrices:

\[ P^*_{UME} = \arg\min ||W - PTW'P||, \ \text{s.t.} \ P \in O(n) \quad (4) \]
The solution of (4) can be computed using an eigendecomposition of \( W \) and \( W' \), which are assumed symmetric:\(^1\)

\[
P_{UME}^* = U'U^T, \quad \text{with } W = USU^T, \quad W' = U'S'U'^T
\]

An approximate solution to (3) can then be computed as a bipartite assignment problem. UME can be thought of \(^6\) as finding an alignment of vertices after embedding them in an euclidian space. Embedding coordinates are given by the eigenvectors of \( H \) are assumed symmetric. The solution of (4) can be computed using an eigendecomposition of \( H \) is converted to a similarity matrix \( H_{ii'} = \exp(-|F_i - F_{i'}|^2/(2\sigma^2)) \) for \((i,i') \in V \times V'\), so that the problem becomes:

\[
p^* = \arg \max \sum_i H_{ip(i)} \cdot \text{with } p \text{ a permutation}
\]

As with UME, the problem is converted into finding a permutation matrix and then relaxing to orthogonal matrices:

\[
P_{SLH}^* = \arg \max \sum_{ij} P_{ij} H_{ij} = \arg \max \text{tr}(P^T H), \quad \text{s.t. } P \in O(n),
\]

The solution is analog to UME: a singular value decomposition of \( H \) is used to retrieve an orthogonal matrix:

\[
P_{SLH}^* = U'U^T, \quad \text{with } H = U'SU^T
\]

Note that we can extend this result to the case \(|V| \neq |V'|\) with the rectangular orthogonal matrix \( P_{SLH}^* = U'E_1 U^T \). \( E \) is formed by replacing each diagonal entry in \( S \) by a 1.

3 Single-Attributed graph matching as a quadratic form minimization

We show that those two formulations, UME and SLH, are special cases of a more general framework which we exploit in the rest of the article. More precisely, we show that under a quadratic constraint,

1. UME is equivalent to minimizing a quadratic form \( X \mapsto X^T(W' \otimes W)X \)
2. SLH is equivalent to minimizing a quadratic form \( X \mapsto X^T(HH^T \otimes HTH^T)X \)

For the first point, let us rewrite (1) under the UME hypothesis: \( \epsilon_1(p) = \sum_{ij} W_{ij}^2 + \sum_{ij} W_{p(i)p(j)}^2 - 2 \sum_{ij} W_{ij} W_{p(i)p(j)} \). Using the Frobenius norm \( \| \cdot \| \), and the fact that \( p \) is a permutation, we have \( \sum_{ij} W_{p(i)p(j)}^2 = \|W'\|^2 \). The third term can be written with the Kronecker product \( W' \otimes W \): \( W_{ij} W_{p(i)p(j)} = (W' \otimes W)_{ip(i),jp(j)} \), with the abuse of notation \( ii' := i + n \cdot (i' - 1) \) to index rows and columns of \( W' \otimes W \). If we reshape \( P^T \) (where \( P \) is the permutation matrix equivalent to \( p \)) into a vector \( X \in \{0,1\}^{n^2 \times 1} \) with

\(^1\)UME algorithm can also be extended to non-symmetric matrices, by first converting them to equivalent hermitian matrices

\(^2\)SLH algorithm can be extended to vertices with multiple attributes, but this distinction is irrelevant since in the end we convert everything to a similarity matrix
We now show the second point involving SLH. Letting $H$ also be written in this form:

$$X = \delta_{p(i)i'}, \text{we get } \sum_{ij} (W' \otimes W)_{ip(i),jp(j)} = X^T (W' \otimes W) X.$$  

Putting everything together, we get the following final expression of the matching cost function:

$$c_1(p) = c_3(X) = ||W||^2 + ||W'||^2 - 2X^T (W' \otimes W) X, \text{ with } X_{ii'} = \delta_{p(i)i'} \quad (9)$$

We can drop the constant terms to get the following equivalent optimization problem in $X$:

$$X^* = \arg \max X^T (W' \otimes W) X \text{ subject to } X^R \in Perm(n) \quad (10)$$

where $X^R$ denotes the $n^2 \times 1$ vector $X$ reshaped as a $n \times n$ matrix: $X^R(i, i') = X_{ii'}$ and $X^R = P^T$. With similar arguments we can prove that the relaxed formulation in (4) can also be written in this form:

$$X^*_{UME}(W, W') = \arg \max X^T (W' \otimes W) X \text{ subject to } X^R \in O(n) \quad (11)$$

We now show the second point involving SLH. Letting $H = U'SU^T$ be the SVD decomposition of $H$, we have $H^TH = US^2U^T$ and $HH^T = U'S^2U^T$. Notice that the SLH solution, $U'SU^T$ is the same as the UME solution to the problem where we have $W = H^TH$ and $W' = HH^T$. Hence, by (11), the solution to SLH is given by:

$$X^*_{SLH}(H) = \arg \max X^T (H^TH \otimes H^TH) X \text{ subject to } X^R \in O(n) \quad (12)$$

This concludes our proof: both UME and SLH can be interpreted as minimizing a quadratic form with a special structure, arising from a Kronecker product. This reformulation leads to a direct generalization to the multi-attributed graph matching with arbitrary graph sizes.

4 Extension to E-MARG Matching

We first reformulate the E-MARG Matching cost to simplify the derivation below. Noting that for any permutation matrix $P$, we have $\text{diag}(P^TF^T) = P^T \text{diag}(F^T) P$, we can simplify (2) by dropping the vertex attributes and replacing them with additional edge attributes on self-loops. Using a similar derivation as above, E-MARG matching is equivalent to the following problem:

$$X^*_{E-MARG}(G, G') = \arg \max X^T (\sum_r W^r \otimes W^r) X \text{ subject to } X^R \in Perm(n) \quad (13)$$

Here, $\sum_r W^r \otimes W^r$ cannot in general be expressed as a kronecker product, and it is not at all clear how the UME algorithm can solve the problem in the continuous domain $X^R \in O(n)$. We need a different solution to handle an arbitrary quadratic form. As we will see in the next section, the kernel of the quadratic form, which we will denote as $W \otimes W' \in \mathbb{R}^{nn' \times nn'}$, can be interpreted as an affinity matrix in a bigger graph whose nodes are all the hypothesized matchings.

5 General Case: MARG Matching via partitioning on a strong product graph $G \odot G'$

This section explains the main concept in this paper, namely that we can match two MARGs $G, G'$ by partitioning a strong product graph $G \odot G'$. We present the details of our algorithm and compare it with other approaches. As with any matching algorithm there are three issues to consider:

1. Class of matching, i.e. one-to-one, one-to-many, etc.
2. Criterion for matching, as defined by the quadric form in our case and in many other formulations.
3. Computational solution for finding the optimal matching.
5.1 Class of matching

Up to now we assumed that a matching is a bijection between the vertex sets $V$ and $V'$. In practice, it is not necessarily a good thing to force a one-to-one correspondence, and the choice of relaxation for finding an approximate continuous solution is not obvious. As hinted in the previous section, we can represent any matching (discrete or continuous) with a matrix $X = (X_{ii'}) \in \mathbb{R}^{n \times n'}$; permutation matrices are regarded as a special case. A strong value of $X_{ii'}$ indicates a good match between $i$ and $j$. Table 1 summarizes different possible definitions for the set of feasible matchings, which we will denote $\Omega$. We choose for $\Omega$ a linear analog of the set of bistochastic matrices, relaxing the positivity constraint and forcing the rows and columns to sum up to 0 instead of 1:

$$\Omega = \{ X \in \mathbb{R}^{n \times n'} : X 1 = 0, X^T 1 = 0, \| X \| = 1 \} \quad (14)$$

One advantage of this choice against all the others is that it accounts for the common case where there can be extra (resp. missing) nodes in the first graph $G$, whether $|V| = |V'|$ or not: this will be indicated by a row (resp. a column) with zeros or very small values.

<table>
<thead>
<tr>
<th>feasible set $\Omega \subset \mathbb{R}^{n \times n'}$</th>
<th>Constraint type</th>
<th>Class of Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Perm(n)$</td>
<td>discrete</td>
<td>one-to-one</td>
</tr>
<tr>
<td>$O(n)$</td>
<td>quadratic</td>
<td>UME, SLH approximations</td>
</tr>
<tr>
<td>bistochastic matrices</td>
<td>affine</td>
<td>many-to-many with two-way constraint, graduated assignment [2]</td>
</tr>
<tr>
<td>stochastic matrices</td>
<td>affine</td>
<td>one-to-many with one-way constraint (asymmetric matching)</td>
</tr>
<tr>
<td>$X \in \mathbb{R}^{n \times n'} : X 1 = 0, X^T 1 = 0, | X | = 1$</td>
<td>linear &amp; quadratic</td>
<td>spectral matching with subspace projection (this paper)</td>
</tr>
</tbody>
</table>

Table 1: Possible definitions for the set of feasible matchings $\Omega$, in decreasing order of strictness

5.2 Matching criterion

Based on the results of the previous sections, it is natural at this point to introduce the strong product graph $G \otimes G'$, whose vertices represent matching hypothesis $(i, i')$ (for short $ii'$) between the two MARGs[4]. It is defined as\footnote{The usual definition of a strong product graph is slightly different, but matches our definition in the case when $E$ and $E'$ contain all self loops $(i, i)$ and $(i', i')$. This will be our case, w.l.o.g.}:

$$G \otimes G' = (V \times V', E \otimes E', W \otimes W') \quad (15)$$

$$(ii', jj') \in E \otimes E' \text{ iff } (i, j) \in E \text{ and } (i', j') \in E' \quad (16)$$

$$W \otimes W' \in \mathbb{R}^{nn' \times nn'} \quad (17)$$

We define the MARG optimization problem as:

$$X^*_\text{MARG}(G, G') = \arg \max X^T W \otimes W' X \text{ subject to } X^R \in \Omega \quad (18)$$

As summarized in Table 2, we can cast of lot of graph matching algorithms in this framework, including ours. Note also that this cost function is equivalent to the original cost function defined in (1) when we take $\Omega = Perm(n)$ and $(W \otimes W')_{ii', jj'} = -d_R^2(W_{ij}, W'_{i'j'})$ (with vertex attributes folded into edge attributes as described earlier). The main difficulty is on the design of the compatibility matrix $W \otimes W'$. 

$$3$$

\footnote{The usual definition of a strong product graph is slightly different, but matches our definition in the case when $E$ and $E'$ contain all self loops $(i, i)$ and $(i', i')$. This will be our case, w.l.o.g.}
Table 2: Pairwise matching compatibilities, in increasing order of generality

<table>
<thead>
<tr>
<th>Pairwise matching compatibility</th>
<th>Matching criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagonal binary matrix</td>
<td>Bipartite matching</td>
</tr>
<tr>
<td>$HH^T \otimes H^T H$</td>
<td>Feature association (SLH) with correspondence matrix $H$</td>
</tr>
<tr>
<td>$W' \otimes W$</td>
<td>Weighted graph alignment between $W$ and $W'$ (UME)</td>
</tr>
<tr>
<td>$\sum_r W^{r'} \otimes W^r$</td>
<td>E-MARG with edge attributes $r \in R$</td>
</tr>
<tr>
<td>Arbitrary matrix</td>
<td>Graduated assignment, spectral matching (this paper)</td>
</tr>
</tbody>
</table>

Advantages of this formulation There is no hard distinction between matching single or multiple attributed relational graphs. The ordering of nodes is irrelevant, as is the case with UME and SLH. Finally, the algorithm works the same whether or not the graphs to match are of the same size.

5.3 Computational Solution

We can rewrite the optimization problem (18) using (14) as:

$$X^*_{\text{MARG}}(G, G') = \arg \max X^T W \otimes W' X \quad \text{subject to} \quad CX = 0, X \in \mathbb{R}^{nn'}$$

(19)

where $C$ is the matrix defined by the constraints $X^R 1_{n'} = 0$, $(X^R)^T 1_n = 0$. The explicit form of constraint matrix $C$ is given by:

$$C = \begin{pmatrix} \mathbf{I}_n & \ldots & \mathbf{I}_n \\ 1_{1 \times n} & \ldots & 0 \\ 0 & \ldots & 1_{1 \times n'} \end{pmatrix} \in \mathbb{R}^{(n+n') \times nn'}$$

(20)

We recognize a cost function used in graph segmentation, namely the Average Affinities. We have therefore transformed our original graph matching problem into a graph segmentation problem, defined on the strong product graph $G \odot G'$. The foreground and background partitions are given by discretisation of the matching indicator vector $X$ (or equivalently, the matching indicator matrix $X^R$). A foreground (resp. background) vertex $(i, i') \in V \times V'$ corresponds to a matched (resp. unmatched) pair $(i, i')$.

We can push further the analogy with segmentation to get some interesting results. If instead of bipartitioning the graph we seek a $K$-way segmentation, this will translate into disjoint matchings, each one involving different pairs of matched nodes. The $K^{th}$ partition consists of the remaining unmatched pairs. This is a remarkable property, that contrasts our approach to other graph matching methods. For example, given an image with two human faces A and B (one familiar and the other less familiar) and a target face image C, the 2-way matching ($K=3$) would ideally find two distinct matchings from A to C and B to C, instead of finding a best and a second-best matching from A to C.

The constrained optimization problem in (19) has been addressed in [7], which showed how to compute the exact solution efficiently. Here we can use the particular form of the constraint to derive a simpler solution. Let

$$c = I_n - 1/n \cdot 1_n \otimes 1_n \in \mathbb{R}^{n \times n}$$

(21)

$$c' = I_{n'} - 1/n' \cdot 1_{n'} \otimes 1_{n'} \in \mathbb{R}^{n' \times n'}$$

(22)

Then we have the following property that $\forall X \in \mathbb{R}^{nn'}, C(c \otimes c')X = 0$. Using this fact, we reduce (19) to the following problem:
\[
X_{\text{MARG}}(G, G') = \arg \max_X \frac{X^T(c \otimes c')(W \odot W')(c \otimes c')X}{X^TX} \quad \text{with } X \in \mathbb{R}^{nn'}
\] (23)

The optimal continuous solution is found by computing the leading eigenvector of the system:
\[
(c \otimes c')(W \odot W')(c \otimes c')X = \lambda X
\] (24)

To discretize the reshaped matching indicator matrix \(X^R\), we can use non maximum suppression. As a slight technicality, since the sign of \(X\) is arbitrary, we need to choose \(X^R\) or \(-X^R\), whichever corresponds to a good matching.

The cost of this algorithm is dominated by the computation of the eigenvectors of \((c \otimes c')(W \odot W')(c \otimes c')\), which is function of two terms: 1) number of matrix-vector operations required in an eigensolver, and 2) cost per matrix-vector operation, which is proportional to the number of non-zero elements in \(W \odot W'\). If we assume a full-matching, this is \(O(|E| \cdot |E'|)\).

6 Results

We test our algorithm on a few images and handwritten digits from the MNSIT database, see Figure 2. For each image, we extract a few landmark points from the image skeleton or the image edges. We build a MARG with landmark points as vertices and simple edge attributes representing the spatial relations between the landmark points, as described in the figure caption. The results show that the algorithm can recover from missing or extra image parts, as well as differences in the point set topology of the two graphs. Note that the matching results are blind to any translation.

7 Summary and Contrast

In conclusion, we have developed a new framework for graph matching between two Attributes Relational Graphs. We show how to reduce the graph matching problem to a segmentation problem in a graph that incorporates both unary and binary constraints between matching hypothesis. Spectral partitioning techniques allow us to efficiently solve the matching problem in continuous space solutions without resorting to expensive graph search algorithms or ad-hoc pruning heuristics.

References

Figure 2: Matching results on a few images (top half) and digits from MNIST database (bottom half). In each case, a few landmark points are extracted from the edges or the skeleton of two images to match. Those are used as vertices in a Multiple Attribute Relational Graph (MARG). There are no vertex attributes, and the edge attributes for each graph are defined as the relative position of points: $W^{dx}(i,j) = x_j - x_i$, $W^{dy}(i,j) = y_j - y_i$. The compatibility matrix is defined as $(W \odot W')_{ij,ij'} = \exp(-||W_{ij} - W'_{ij'}||^2/\sigma^2)$. The matching in each case is displayed as follows: in the 1st and 2nd row, two landmark points that are matched have the same color. The 3rd shows the matching links. We can see that the algorithm detects with some success extra or missing nodes (dog tail, shadow of the horse, extra line in the bottom 8 and the bottom 2).