Graph matching with type constraints

Catherine Fraikin and Paul Van Dooren
Université catholique de Louvain
Department of Mathematical Engineering
B-1348 Louvain-la-Neuve, Belgium
fraikin@inma.ucl.ac.be; vdooren@inma.ucl.ac.be

Abstract— **We consider the problem of comparing two directed graphs with nodes that have been subdivided into classes of different type. The matching process is based on a constrained projection of the nodes of the graphs in a lower dimensional space. This procedure is formulated as a non-convex optimization problem. The objective function uses the two adjacency matrices of the graphs where the nodes are adequately numbered. The constraints on the problem impose the isometry of the so-called projections. An iterative algorithm is proposed to solve the optimization problem. As illustration, we give an example of graph matching for graphs with two types of nodes. Finally, an extension for comparing both groups of nodes in a directed bipartite graph is presented.**

Index Terms— **Graph matching, Optimization, Typed nodes**

I. Introduction

Graphs are a powerful tool for many practical problems such as pattern recognition, shape analysis, image processing and data mining. A fundamental task in this context is that of graph matching. Many approaches have been proposed for graph matching, but one can distinguish between two broad classes : the first one tries to find a one-to-one correspondence between some of the vertices of the two graphs (exact graph matching); the second one allows inexact matching and looks for an optimal match even if the considered graphs are structurally different (a survey can be found in [1]). In practice, the second class of methods is the most interesting one because it is more flexible and often gives rise to algorithms that are cheaper to implement.

In this paper we consider an inexact graph matching method to compare two graphs with nodes that have been subdivided into classes of different type. The nodes of the same type in the two graphs are compared to each other, but taking into account the complete interconnection pattern of the graphs. The proposed method is based on the optimization of a certain cost function. The method specializes to the spectral method of Caelli and Kosinov in the case that the graphs to be compared are undirected and contain only one type of nodes [2]. It is also an extension of the method described in [3] which handles the directed graph case for nodes of one type only, which in turn is a low rank approximation of ideas developed in [4]. The computational technique that we propose is also very similar to that of the above two methods and is essentially a modified power method with special correction applied at each step of the iteration. Since the basic operation to be performed at each step is that of multiplying certain bases with the adjacency matrices of the two graphs, the basic step of the computational procedure can be implemented at low cost for large sparse graphs.

We illustrate the comparison technique with an application to graphs that are essentially bipartite, and for which there are clearly two groups of nodes in each graph. We also use this example to illustrate the convergence behavior of our proposed algorithm.

Finally, we propose a modified problem for comparing both groups of nodes in a directed bipartite graph.

II. Cost function

Let $\mathbb{R}$ denote the real field and $\mathbb{R}^{m\times n}$ denote the set of all $m \times n$ real matrices. X^T represents the transpose of X. For $X, Y \in \mathbb{R}^{m\times n}$, the Frobenius *inner product* is defined by

$$\langle X, Y\rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}Y_{ij}.$$

and its corresponding norm by

$$\|X\| = \langle X, X\rangle^{1/2}.$$

This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, and a grant Action de Recherche Concertée (ARC) of the Communauté Française de Belgique. The scientific responsibility rests with its authors.

ISBN: 978-960-89028-5-5

Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a differentiable (real-valued) function with matrix argument X. Then the first-order approximation of f at a point X can be expressed as

$$f(X+\Delta) = f(X) + \langle \nabla f(X), \Delta\rangle + o(\|\Delta\|)$$

where the gradient $\nabla f(X)$ is the $m \times n$ matrix whose (i,j) entry is $\frac{\partial f(X)}{\partial X_{i,j}}$. Related to this, we provide some gradients needed in the rest of the paper

$$\nabla\langle A, X^T X\rangle = X(A + A^T), \qquad \text{(II.1)}$$
$$\nabla\langle X^T AX, B\rangle = AXB^T + A^T XB. \quad \text{(II.2)}$$

The approach presented in this paper is an extension of the work on inexact graph matching proposed by the authors [3] for graph matching of two arbitrary directed graphs. In that work two graphs G_A and G_B of dimensions m and n, respectively, were compared by projecting the two graphs into a k-dimensional space, where $k \le \min(m,n)$ is typically very small (for visualization purpose, one chooses e.g. $k = 2, 3$). The projected nodes are obtained as follows: each node i of G_A is mapped to the normalized row i of a matrix $U \in \mathbb{R}^{m\times k}$, and each node j of G_B is mapped to the normalized row j of $V \in \mathbb{R}^{n\times k}$. The projected nodes are thus mapped on the unit sphere in a k-dimensional space, and can then be compared using any preferred matching technique on that manifold (e.g. using nearest-neighbor ideas). We give an example of this in Section V.

The matrices U and V are obtained from the optimization of a cost function which uses the adjacency matrices A and B of the two graphs. It is given by

$$\begin{aligned}\max\nolimits_{U,V}\{\langle U^T AU, V^T BV\rangle :&\\ U \in \mathcal{Q}_{m,k}, V \in \mathcal{Q}_{n,k}\}.& \quad \text{(II.3)}\end{aligned}$$

where $\mathcal{Q}_{m,k}$ denotes the set of $m\times k$ matrices with orthonormal columns :

$$\mathcal{Q}_{m,k} = \{X \in \mathbb{R}^{m\times k} : \; X^T X = I_k\}.$$

The mathematical properties of the non convex optimization problem (II.3) are presented in [3]. In general, there is no closed form for the optimal value, except for special matrices A and B (e.g. if A or B is symmetric). For arbitrary matrices, only an upper bound to the problem is obtained. This value is an adequate combination of the eigenvalues of the symmetric and skew-symmetric parts of A and B.

If we now have graphs with different types of nodes, we assume that they have been relabelled such that in the corresponding adjacency matrices

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1\ell}\\ A_{21} & A_{22} & \dots & A_{2\ell}\\ \vdots & \vdots & \ddots & \vdots\\ A_{\ell 1} & A_{\ell 2} & \dots & A_{\ell\ell}\end{bmatrix},$$
$$B = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1\ell}\\ B_{21} & B_{22} & \dots & B_{2\ell}\\ \vdots & \vdots & \ddots & \vdots\\ B_{\ell 1} & B_{\ell 2} & \dots & B_{\ell\ell}\end{bmatrix},$$

the nodes of the same type $i = 1, \cdots, \ell$ correspond to the same blocks in both matrices A and B. The blocks $A_{i,j} \in \mathbb{R}^{m_i\times m_j}$ and $B_{i,j} \in \mathbb{R}^{n_i\times n_j}$ thus describe the edges between nodes of type i to nodes of type j in both A and B.

The rationale behind cost function (II.3) is that the so-called *projections* U and V describe the dominant behavior of both adjacency matrices A and B, but in terms of a *joint cost function* (II.3), which emphasizes the *correlation* between both projected matrices. Since our projections U and V should not mix nodes of different types, we will constrain them to have a block diagonal form :

$$U = \begin{bmatrix} U_1 & 0 & \dots & 0\\ 0 & U_2 & \dots & \vdots\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \dots & 0 & U_\ell\end{bmatrix},$$
$$V = \begin{bmatrix} V_1 & 0 & \dots & 0\\ 0 & V_2 & \dots & \vdots\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \dots & 0 & V_\ell\end{bmatrix}, \qquad \text{(II.4)}$$

where $U_i \in \mathcal{Q}_{m_i\times k_i}$ and $V_i \in \mathcal{Q}_{n_i\times k_i}$, which implies that $k_i \le \min(m_i, n_i)$. Notice that this is essentially the same optimization problem as in (II.3) except for the additional constraint that U and V are block diagonal *projections*, which of course prevents the mixing of types in the projected nodes. We will subsequently have to match nodes within each type i, using again a preferred matching algorithm, on a sphere in a k_i-dimensional space, and this for $i = 1, \cdots, \ell$. In the rest of this paper, we will assume for ease of notation that $\ell = 2$, i.e. that there are only two different types of nodes. All results extend trivially to the case of arbitrary number of types ℓ.

III. Theoretical aspects

The mathematical properties of the optimization problem (II.3) were presented in [3] for $\ell = 1$. We give here a very short proof for the extension to the constrained case with $\ell = 2$ blocks. We thus consider two graphs with partitioned adjacency matrices

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

with $A_{i,j} \in \mathbb{R}^{m_i \times m_j}$ and $B_{i,j} \in \mathbb{R}^{n_i \times n_j}$ and orthogonal matrices $U_i \in \mathcal{Q}_{m_i \times k_i}$ and $V_i \in \mathcal{Q}_{n_i \times k_i}$, for $i = 1, 2$ and $j = 1, 2$. The optimization problem (II.3) then becomes

$$\begin{aligned} \max_{U_i, V_i} \{ \langle & \begin{bmatrix} U_1^T A_{11} U_1 & U_1^T A_{12} U_2 \\ U_2^T A_{21} U_1 & U_2^T A_{22} U_2 \end{bmatrix}, \\ & \begin{bmatrix} V_1^T B_{11} V_1 & V_1^T B_{12} V_2 \\ V_2^T B_{21} V_1 & V_2^T B_{22} V_2 \end{bmatrix} \rangle : \\ & U_i \in \mathcal{Q}_{m_i,k_i}, V_i \in \mathcal{Q}_{n_i,k_i}, i = 1, 2 \}. \quad \text{(III.5)} \end{aligned}$$

which is a continuous function on a compact domain. Therefore, there always exists an optimal solution $(U_i, V_i), i = 1, 2$ such that the first order conditions are satisfied. By using some of the trace properties[1], the objective function becomes

$$\begin{aligned} & \langle U_1^T A_{11} U_1, V_1^T B_{11} V_1 \rangle \\ + & \langle U_2^T A_{22} U_2, V_2^T B_{22} V_2 \rangle \\ + & \langle U_1^T A_{12} U_2, V_1^T B_{12} V_2 \rangle \\ + & \langle U_2^T A_{21} U_1, V_2^T B_{21} V_1 \rangle. \quad \text{(III.6)} \end{aligned}$$

The first-order derivative conditions can be derived from the Lagrangian:

$$\begin{aligned} L(U_1, U_2, V_1, V_2, X_1, X_2, Y_1, Y_2) = \\ & \langle U_1^T A_{11} U_1, V_1^T B_{11} V_1 \rangle \\ & + \langle U_1^T A_{12} U_2, V_1^T B_{12} V_2 \rangle \\ & + \langle U_2^T A_{21} U_1, V_2^T B_{21} V_1 \rangle \\ & + \langle U_2^T A_{22} U_2, V_2^T B_{22} V_2 \rangle \\ & + \langle X_1, (I - U_1^T U_1) \rangle + \langle X_2, (I - U_2^T U_2) \rangle \\ & + \langle Y_1, (I - V_1^T V_1) \rangle + \langle Y_2, (I - V_2^T V_2) \rangle \end{aligned}$$

where X_i and Y_i are symmetric matrices of Lagrange multipliers for the orthogonality constraints. By setting the partial gradients of this Lagrangian to zero (using (II.1) and (II.2)), the first order conditions are found to be :

$$\begin{aligned} U_1 X_1 = & [A_{11} U_1 V_1^T B_{11}^T + A_{11}^T U_1 V_1^T B_{11} \\ & + A_{12} U_2 V_2^T B_{12}^T + A_{21}^T U_2 V_2^T B_{21}] V_1, \\ V_1 Y_1 = & [B_{11} V_1 U_1^T A_{11}^T + B_{11}^T V_1 U_1^T A_{11} \\ & + B_{12} V_2 U_2^T A_{12}^T + B_{21}^T V_2 U_2^T A_{21}] U_1, \\ U_2 X_2 = & [A_{22} U_2 V_2^T B_{22}^T + A_{22}^T U_2 V_2^T B_{22} \\ & + A_{21} U_1 V_1^T B_{21}^T + A_{12}^T U_1 V_1^T B_{12}] V_2, \\ V_2 Y_2 = & [B_{22} V_2 U_2^T A_{22}^T + B_{22}^T V_2 U_2^T A_{22} \\ & + B_{21} V_1 U_1^T A_{21}^T + B_{12}^T V_1 U_1^T A_{12}] U_2, \end{aligned}$$

where $X_i = Y_i$ are symmetric matrices. Clearly the cost in (III.5) does not change if one multiplies each pair U_i, V_i by a common $k_i \times k_i$ orthogonal matrix Q_i. That degree of freedom can also be used to impose that $X_i = Y_i$ are also equal to diagonal matrices Λ_i (see [3] for more details).

IV. Computational aspects

In this section we present an iterative algorithm to find a critical point of (III.5). It is a simple recursive algorithm based on the Singular Value Decomposition, which provides interesting results. A convergence is still missing but numerical experiments always show linear convergence to an equilibrium point, provided the shifts $s_i, i = 1, 2$ are chosen appropriately.

The proposed iterative algorithm to compute a critical point of (III.5) is as follows

$$\begin{aligned} U_{1+} \Sigma_{1+} V_{1+}^T + U_{1-} \Sigma_{1-} V_{1-}^T = & \\ & A_{11} U_1 V_1^T B_{11}^T + A_{11}^T U_1 V_1^T B_{11} \\ & + A_{12} U_2 V_2^T B_{12}^T + A_{21}^T U_2 V_2^T B_{21} \\ & + s_1 U_1 V_1^T \quad \text{(IV.7)} \\ U_{2+} \Sigma_{2+} V_{2+}^T + U_{2-} \Sigma_{2-} V_{2-}^T = & \\ & A_{22} U_2 V_2^T B_{22}^T + A_{22}^T U_2 V_2^T B_{22} \\ & + A_{21} U_1 V_1^T B_{21}^T + A_{12}^T U_1 V_1^T B_{12} \\ & + s_2 U_2 V_2^T \quad \text{(IV.8)} \end{aligned}$$

where U_{i+} and V_{i+} are orthogonal complements of U_{i-} and V_{i-}. The scalars s_i are positive numbers sufficiently large (see [3] for more details) and $\Sigma_{i\pm}$ are diagonal matrices. The updating of the matrices U_i, V_i is done at each iteration according to

$$\begin{aligned} U_1 := U_{1+}, \quad & U_2 := U_{2+}, \\ V_1 := V_{1+}, \quad & V_2 := V_{2+}. \quad \text{(IV.9)} \end{aligned}$$

It is easy to see that $U_{1+} \Sigma_{1+} V_{1+}^T$ and $U_{2+} \Sigma_{2+} V_{2+}^T$ are the best rank k_i approximations of the respective right hand sides. In practice the previous iteration is realized by the application of the SVD algorithm to the right-hand side of the equation,

[1] $trA = trA^T, \quad trAB = trBA, \quad tr(A + B) = trA + trB.$

which is a matrix of rank at most $5k_i$. This can be exploited of course in the application of the SVD algorithm to these right hand sides. For sparse matrices A and B, the complexity of the computation of one iteration step (IV.7), (IV.8), (IV.9) is then only linear in m and n.

One can show that the critical points of the cost function (III.5) are fixed points of the iteration (IV.7), (IV.8), (IV.9) and vice-versa. The proof of this is very similar to that of the one block case [3] and is omitted here.

V. Experiments in typed graph matching

In order to apply this problem to graph matching, we use it to compare two directed graphs, the nodes of which have been divided into two types, labelled group 1 and 2.

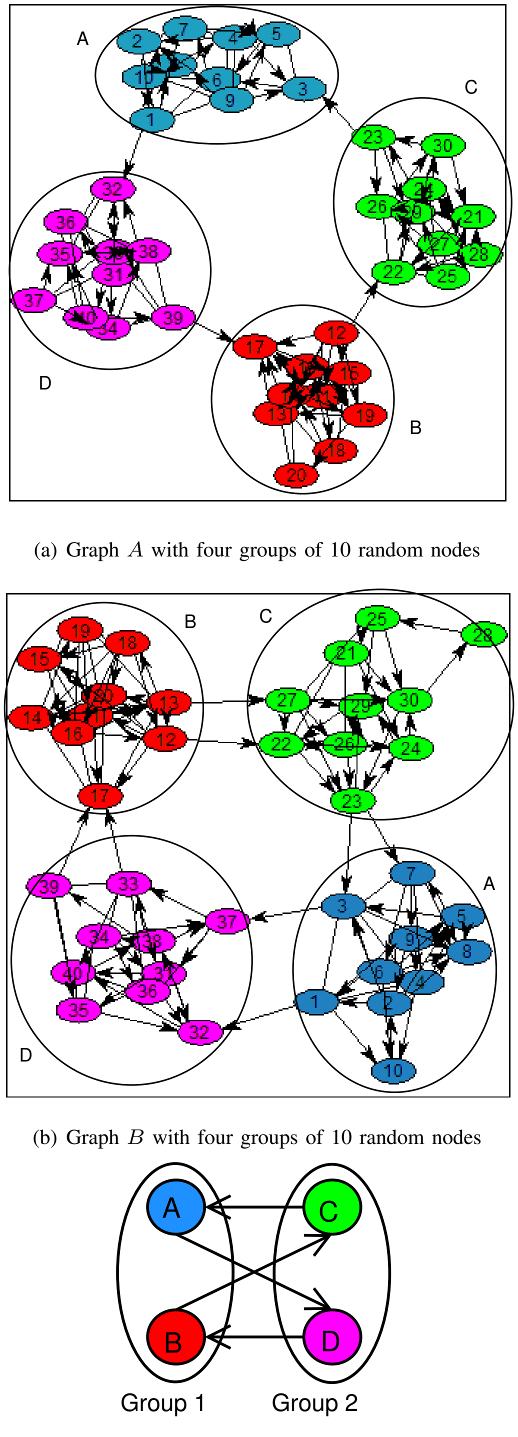
In Figure 1 we show two graphs that are *essentially* bipartite in the sense that nodes from clusters A and B (denoted as group 1 in Figure 1(c)) point to nodes from clusters C and D (denoted as group 2 in Figure 1(c)), and vice versa. These are not *true* bipartite graphs because within the clusters, there are random connections between the nodes of that cluster. But clearly one hopes to detect a close connection between the groups of nodes of type A and B in both graphs and those of type C and D in both graphs. Indeed, when imposing the constraint that only nodes of group 1 and 2 can be compared with each other in both graphs, then there is a clear distinction between the subgroups of type A and B in group 1, on the one hand, and those of type C and D in group 2, on the other hand. This is also what is observed in Figure 2 when projecting both groups of nodes in a two dimensional space.

When one zooms in on each of the four clusters in Graphs A and B, one clearly sees in Figure 3 that the nodes of these clusters are different, but quite close to each other.

We also show the convergence behavior of our algorithm for this example. We measured the convergence by checking the maximal residual norm at each step

$$Residual(step) :=$$
$$\max(\|U_1V_1^T - U_{1+}V_{1+}^T\|_2, \|U_2V_2^T - U_{2+}V_{2+}^T\|_2)$$

which is what we plot in Figure 4(a) below. One can clearly observe linear convergence of our algorithm. Finally, we show in Figure 4(b) the convergence of the actual value of the cost, as a function of the iteration step.



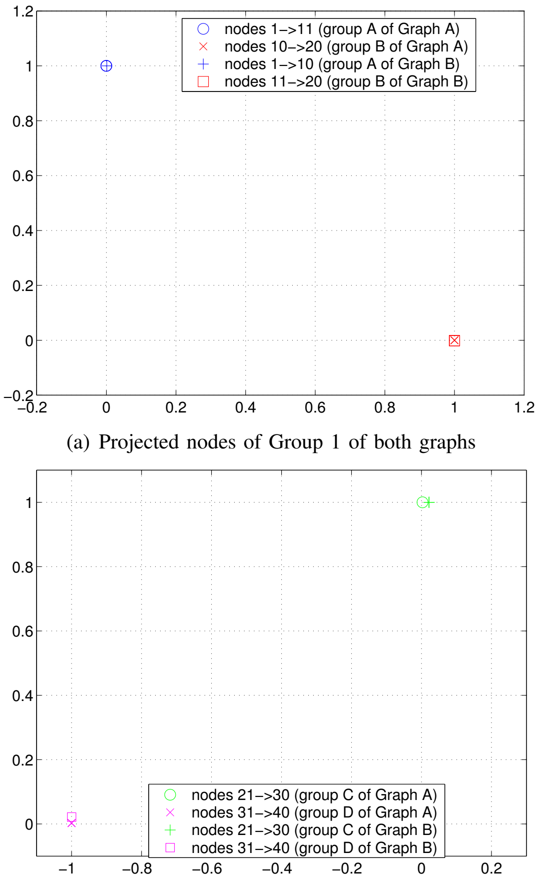
(a) Graph A with four groups of 10 random nodes

(b) Graph B with four groups of 10 random nodes

(c) Bipartite partition

Fig. 1. Two *essentially* bipartite graphs

Let us remark that we have presented one solution corresponding to a local minima of problem (III.5). For this solution, the nodes of cluster A in Graph A are compared to nodes of cluster A in Graph B, and so on for the four clusters of nodes. For another local minimum or solution, the



(a) Projected nodes of Group 1 of both graphs

(b) Projected nodes of Group 2 of both graphs

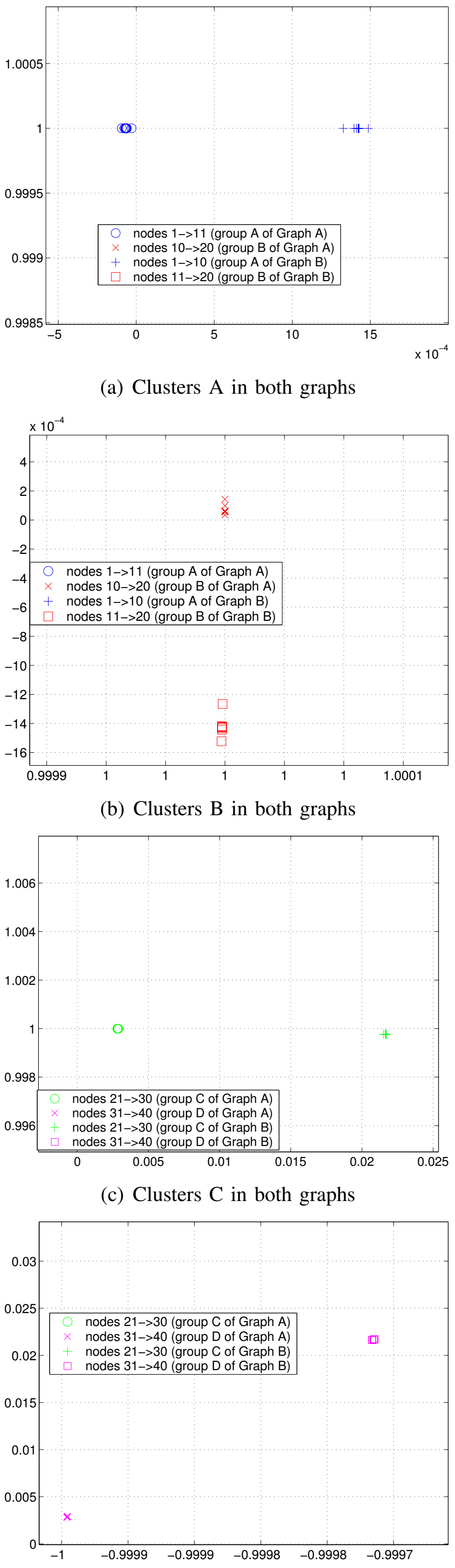
Fig. 2. Closeness between nodes of the same clusters in each group

nodes of cluster A in Graph A would have been compared to nodes of cluster B in Graph B, the nodes of cluster B in Graph A to nodes of cluster A in Graph B, the nodes of cluster C in Graph A to nodes of cluster D in Graph B and finally, the nodes of cluster D in Graph A would have been compared to nodes of cluster C in Graph B. Depending on the initial condition for the iteration, the solution will be different, but the nodes will always be compared with nodes of the same type.

If we suppress the types in the previous example. Clusters of nodes will now be compared without any constraints of type. One then automatically obtains four local minima that correspond to cyclic permutations of the clusters A, B, C and D in graphs A and B.

VI. Case of bipartite graphs

A directed bipartite graph of dimension m is a special graph where the set of nodes can be divided into two disjoint sets $\{M_1, M_2\}$ of dimensions m_1



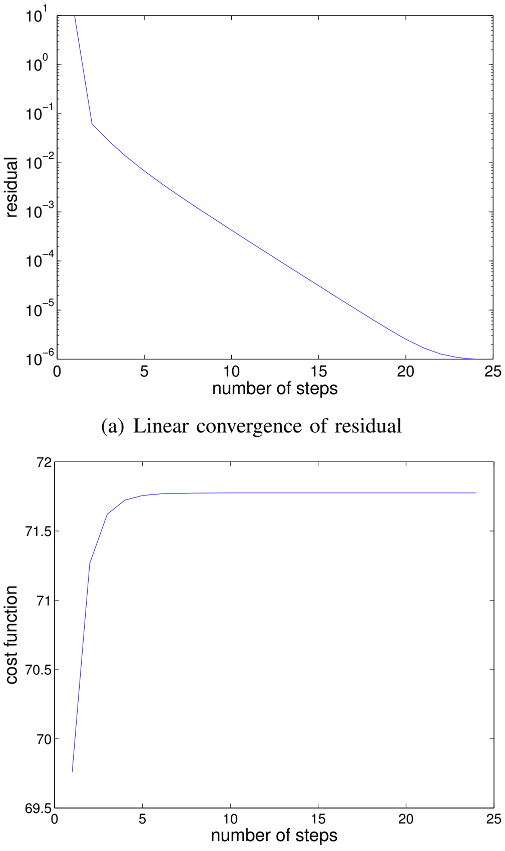
(a) Clusters A in both graphs

(b) Clusters B in both graphs

(c) Clusters C in both graphs

(d) Clusters D in both graphs

Fig. 3. Zooming in on the four different groups in Graphs A and B



(a) Linear convergence of residual

(b) Convergence of the cost

Fig. 4. Convergence behavior

and m_2, with $m_1+m_2 = m$, such that no edge has both end-points in the same set. A bipartite graph has then the following adjacency matrix

$$A_B = \left(\begin{array}{cc} 0 & A_1 \\ A_2^T & 0 \end{array} \right) \qquad \text{(VI.10)}$$

with $A_1, A_2 \in \mathbb{R}^{m_1 \times m_2}$. In this section, we treat the problem of bipartite graph matching, in the sense of comparing together both sets of nodes, i.e. comparing nodes of group M_1 with nodes of group M_2. In this case, we will see that the problem is reduced to a smaller one.

In problem (III.5), we choose $A = A_B$ and $B = A_B^T$ and the projections U and V are constrained to have the block diagonal form (II.4)

$$U = V = \left[\begin{array}{cc} U_1 & 0 \\ 0 & U_2 \end{array} \right] \qquad \text{(VI.11)}$$

where $U_i \in \mathcal{Q}_{m_i \times k_i}$, $i = 1, 2$, with $k_i \leq m_i$. By using some of the trace properties, the objective function becomes

$$2\langle U_1^T A_1 U_2, U_1^T A_2 U_2 \rangle$$

where $U_1^T U_1 = I_k$ and $U_2^T U_2 = I_k$. The optimization problem can be formulated

$$\begin{aligned} &\max_{U_1,U_2} \{ \langle U_1^T A_1 U_2, U_1^T A_2 U_2 \rangle : \\ &\qquad U_1 \in \mathcal{Q}_{m_1,k_1}, U_2 \in \mathcal{Q}_{m_2,k_2} \}. \qquad \text{(VI.12)} \end{aligned}$$

Let us remark that, for $m_1 = k_1 = m_2 = k_2$, this optimization problem is equivalent to the problem presented in [5]. The first-order derivative conditions can be derived from the Lagrangian

$$\begin{aligned} L(U_1, U_2, X, Y) &= \langle U_1^T A_1 U_2, U_1^T A_2 U_2 \rangle \\ &+ \langle X, I_k - U_1^T U_1 \rangle + \langle Y, I_k - U_2^T U_2 \rangle \end{aligned}$$

where X and Y are symmetric matrices of Lagrange multipliers for the orthogonality constraints. By setting the partial gradients of $L(U_1, U_2, X, Y)$ to zero, the first-order conditions are found to be:

$$\begin{aligned} U_1 X &= [A_1 U_2 U_2^T A_2^T + A_2 U_2 U_2^T A_1^T] U_1, \\ U_2 Y &= [A_1^T U_1 U_1^T A_2 + A_2^T U_1 U_1^T A_1] U_2. \end{aligned}$$

In this case the matrices X and Y are not equal but satisfy $\operatorname{tr} X = \operatorname{tr} Y$. These symmetric matrices X and Y could be chosen diagonal and are denoted D_X and D_Y respectively. A proposed iterative algorithm to compute a critical point of (VI.12) is the following:

$$\begin{aligned} U_{1+} \Lambda_{X+} U_{1+}^T + U_{1-} \Lambda_{X-} U_{1-}^T &= \\ A_1 U_2 U_2^T A_2^T + A_2 U_2 U_2^T A_1^T & \\ + s U_1 U_1^T, & \qquad \text{(VI.13)} \\ U_{2+} \Lambda_{Y+} U_{2+}^T + U_{2-} \Lambda_{Y-} U_{2-}^T &= \\ A_1^T U_1 U_1^T A_2 + A_2^T U_1 U_1^T A_1 & \\ + s U_2 U_2^T, & \qquad \text{(VI.14)} \end{aligned}$$

where U_{i+} are orthogonal complements of U_{i-}, $i = 1, 2$. The scalar s is a positive number sufficiently large (see [3] for more details) and Λ_{X+}, Λ_{Y+} are diagonal matrices. These iterations correspond to the iteration (IV.7) for one type of nodes ($l = 1$), where we have replaced the matrices A, B, U and V according to (VI.10) and (VI.11). The role of the shift s is to select the adequate part of the spectra. The updating of the matrices U_1, U_2 is done at each iteration according to

$$U_1 := U_{1+}, \quad U_2 := U_{2+}. \qquad \text{(VI.15)}$$

In practice the previous iteration is realized by application of an eigendecomposition to the right-hand side of the equation. The dominant spaces correspond to the rightmost part of the spectra. One can easily see that a fixed point (U_1, U_2) of (VI.13) and (VI.14) is a critical point of (VI.12) with $D_X = \Lambda_X - sI$ and $D_Y = \Lambda_Y - sI$. There is no proof of convergence but numerical experiments always converge to an equilibrium point.

VII. Conclusion

In this paper, we extend the projected correlation method described in [3], which can be used to perform graph matching between two graphs represented by their adjacency matrices A and B. The proposed extension constrains the matching such that the nodes of both graphs can only be compared when their types are the same : both graphs are thus assumed to be partitioned in subgraphs of equal type. The proposed cost function nevertheless uses the connectivity between nodes of different type in the global cost, to be minimized. A modified problem for bipartite graphs is also presented.

The problem of matching graphs with nodes of different types could be extended for graphs with different types of edges. The case of graphs with different types of nodes and edges could then be obtained by combining both approaches.

References

[1] D. Conte, P. Foggia, C. Sansone and M. Vento, “Thirty years of graph matching in pattern recognition”, *Int. J. Pattern Recognition and Artificial Intelligence*, Vol.18(3), pp. 265–298 , 2004.

[2] T. Caelli and S. Kosinov, “An eigenspace projection clustering method for inexact graph matching,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.26(4), pp. 515–519, 2004.

[3] C. Fraikin, Y. Nesterov and P. Van Dooren, “Optimizing the coupling between two isometric projections of matrices”, submitted to *SIAM J. Matrix Anal. Appl.*, 2006.

[4] V. Blondel, A. Gajardo, M. Heymans, P. Senelart and P. Van Dooren, “Measure of similarity between graph vertices. Applications to synonym extraction and web searching”. *SIAM Review*, pp. 647–666, Vol.46, 2004.

[5] R. Brockett, “Singular values and least squares matching”. *Proc. 36th IEEE Conf. on Dec. and Control*, San Diego, CA, pp. 1121–1124, 1997.