Available online at www.sciencedirect.com
ScienceDirect

ELSEVIER

Linear Algebra and its Applications 429 (2008) 1229–1242

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

A gradient-type algorithm optimizing the coupling between matrices ☆

Catherine Fraikin [a,b,*], Yurii Nesterov [a,c], Paul Van Dooren [a,b]

[a] *Université catholique de Louvain, Department of Mathematical Engineering, Belgium*
[b] *Bâtiment Euler 4-6, Avenue Georges Lemaître, B-1348 Louvain-la-Neuve, Belgium*
[c] *Bâtiment CORE 34, voie du Roman Pays, B-1348 Louvain-la-Neuve, Belgium*

Received 30 October 2006; accepted 15 October 2007
Available online 20 February 2008
Submitted by J. Brandts

Abstract

In this paper, we consider the problem of maximizing the coupling between the isometric projections of two square matrices of dimensions m and n. This coupling is defined as an inner product between the matrices. This is a non-convex optimization problem with isometry constraints on the variables. The optimization set is an equinormed set and we develop a gradient-type algorithm to solve the problem. Numerical experiments and an application to graph matching are also presented.

AMS classification: 15A18; 15A60; 65K15; 65K10; 05C50

Keywords: Optimization; Gradient method; Singular value decomposition; Graph matching

☆ This paper presents research supported by the Concerted Research Action (ARC) “Large Graphs and Networks” of the French Community of Belgium, and by the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister’s Office for Science, Technology and Culture. The scientific responsibility rests with the authors.

* Corresponding author. Address: Université catholique de Louvain, Department of Mathematical Engineering, Belgium.

E-mail addresses: catherine.fraikin@uclouvain.be (C. Fraikin), yurii.nesterov@uclouvain.be (Y. Nesterov), paul.vandooren@uclouvain.be (P. Van Dooren).

0024-3795/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2007.10.015

1. Introduction

Many problems and applications in various areas use projections of matrices in lower-dimensional subspaces. In this paper, we consider the "coupling" between the isometric projections of two square matrices A and B, respectively of dimensions m and n. This coupling between A and B with respect to U and V, is defined as

$$c(U, V) = \mathrm{tr}(U^{\mathrm{T}} A U V^{\mathrm{T}} B^{\mathrm{T}} V) \tag{1.1}$$

where U and V are isometries, i.e. matrices satisfying $U^{\mathrm{T}}U = V^{\mathrm{T}}V = I_k$, and where I_k denotes the identity matrix of dimension k with $k \leqslant (m, n)$. A similar expression can also be defined for complex matrices [6] but for simplicity we restrict ourselves here to the real case.

In this paper, we consider the problem of maximizing the coupling $c(U, V)$ over all isometries U and V. A numerical procedure will be given to find an extremal point (U, V) of this function.

The problem of maximizing $c(U, V)$ over all isometries U and V constitutes an extension of well-known problems from the linear algebra literature, e.g. maximizing the real part of the generalized numerical range or the products of the elements from the fields of values of A and B (see [6] for a survey of these particular cases). For particular matrices and dimensions, optimizing (1.1) can also be linked to various applications, especially in graph theory. We cite as a first example the graph similarity matrix [1] expressing how similar the nodes of two graphs are. A second application concerns the problem of graph matching, and in particular the set of spectral graph matching methods, see [2,9–12] for a description of various spectral methods. In [2], Caelli and Kosinov introduced a spectral method where the nodes of the two graphs are projected, by solving a problem similar to the maximization of (1.1). Let us remark that all the papers describing spectral methods in the field of graph matching are restricted to undirected graphs. Numerical methods for finding extremal points (U, V) of the function $c(U, V)$ will be developed in this paper. These methods have the advantage of allowing a comparison between directed graphs with different number of nodes.

The paper is organized as follows. Section 2 explains the relevance of maximizing the coupling between two square matrices with respect to isometries (1.1). The initial motivation of this problem, coming from graph theory, is also mentioned. In Section 3, the main mathematical properties of the optimization problem are summarized and the theoretical aspects of optimization over equinormed sets are presented. The computational aspects and two algorithms for solving the problem are presented in Section 4. The first algorithm is based on the Singular value decomposition (see [6]) and the second one is a new gradient-based method. Numerical experiments are given in Section 5. In particular experimental comparisons of the two algorithms and a graph matching experiment are presented.

2. The relevance of optimizing the coupling between matrices

This section explains the pertinence of maximizing (1.1), expressing a coupling between the matrices A and B, over all isometries U and V. This was introduced in [6] for complex matrices, but for simplicity, we restrict ourselves here to the real case. Let us first introduce some notation used in the rest of the paper. Let $\mathbb{R}$ denote the real field, $\mathbb{R}^{m\times n}$ denote the set of all $m \times n$ real matrices, and X^{T} represent the transpose of X. For $X, Y \in \mathbb{R}^{m\times n}$, the Frobenius *inner product* is defined by

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

and its corresponding norm by

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

A classical problem in the field of matrix theory concerns the determination of "dominant" subspaces of square matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$, expressed as two independent optimization problems:

$$\begin{aligned}\max_U\{\langle U^{\mathrm{T}}AU, I_k\rangle : U \in \mathcal{Q}_{m,k}\} &= \max_U \left\{\left\langle U^{\mathrm{T}}\frac{A+A^{\mathrm{T}}}{2}U, I_k\right\rangle : U \in \mathcal{Q}_{m,k}\right\},\\ \max_V\{\langle V^{\mathrm{T}}BV, I_k\rangle : V \in \mathcal{Q}_{n,k}\} &= \max_V \left\{\left\langle V^{\mathrm{T}}\frac{B+B^{\mathrm{T}}}{2}V, I_k\right\rangle : V \in \mathcal{Q}_{n,k}\right\},\end{aligned} \tag{2.2}$$

where we assume $k \leqslant (m, n)$. The set of matrices $\mathcal{Q}_{m,k}$, also known as the Stiefel manifold, is defined by

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

The optimal U and V correspond to the "dominant" invariant subspaces of dimension k of the symmetric parts of A and B respectively. Let us remark that replacing U by UQ_1 and V by VQ_2 with Q_1 and Q_2 orthogonal matrices of dimension k does not change the value of the objective function. The degrees of freedom of this problem (represented by the orthogonal transformations Q_1 and Q_2) often need to be fixed when comparing the matrices A and B or their symmetric parts $\frac{A+A^{\mathrm{T}}}{2}$ and $\frac{B+B^{\mathrm{T}}}{2}$.

We therefore consider the following optimization problem:

$$\max_{U,V}\{f(U, V) := \langle U^{\mathrm{T}}AU, V^{\mathrm{T}}BV\rangle : U \in \mathcal{Q}_{m,k}, V \in \mathcal{Q}_{n,k}\}. \tag{2.3}$$

The matrices A and B are now projected simultaneously which allows to fix the degree of freedom of one projection w.r.t. the other one. Indeed, in this, case, the only degree of freedom is to replace the product UV^{T} by $UQ^{\mathrm{T}}QV^{\mathrm{T}}$ with Q an orthogonal matrix of dimension k. This degree of freedom represents a simultaneous orthogonal transformation of the pair of projections U and V. Moreover, the problem yields different results for symmetric and non-symmetric matrices.

The initial motivation for the definition of the optimization problem (2.3) was the extension of the work on inexact graph matching proposed by Caelli and Kosinov [2] which was applicable to undirected graphs. Their method uses the k leading eigenvectors of the adjacency matrices of both graphs, which is equivalent to solving (2.2) for A and B symmetric adjacency matrices of the graphs. Indeed, it is shown that these eigenvectors contain interesting properties for the graph structure (see [5] or [3] for a summary concerning the graph properties connected to the spectrum of the graph). In order to compare graphs of different sizes, a normalization procedure of U and V is proposed in [2]. This normalization corresponds to the projection of the vertices onto a unit hypersphere of dimension k. This method provides good results but is applicable only to undirected graphs. Moreover, the two graphs are projected independently of each other which may lead to an ambiguity of the projections. The coupled projection presented in this paper removes this ambiguity and allows to compare two directed graphs of dimensions m and n represented by non-symmetric matrices A and B.

3. The gradient method and its convergence

The mathematical properties of the optimization problem (2.3) are presented in [6]. This section first recalls some important results of the problem. Secondly, some new results concerning a gradient method for the optimization over equinormed sets are developed.

3.1. Recollection of some properties

Problem (2.3) is an optimization problem of a differentiable function on a compact domain. Therefore there always exists an optimal solution (U, V) such that the first-order conditions are satisfied. These first-order derivative conditions can be derived from the Lagrangian $L(U, V, X, Y)$:

$$\begin{aligned} L(U, V, X, Y) = \frac{1}{2}\{&\langle U^{\mathrm{T}}AU, V^{\mathrm{T}}BV\rangle + \langle U^{\mathrm{T}}A^{\mathrm{T}}U, V^{\mathrm{T}}B^{\mathrm{T}}V\rangle \\ &+\langle X, (I - U^{\mathrm{T}}U)\rangle + \langle Y, (I - V^{\mathrm{T}}V)\rangle\}, \end{aligned} \tag{3.4}$$

where X and Y are symmetric matrices of Lagrange multipliers for the isometry constraints.

Definition 1. Let $X \in \mathbb{R}^{m\times n}$ and $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a differentiable real-valued function with matrix argument X. Then the gradient $\nabla f(X)$ is the $m \times n$ matrix whose (i, j) entry is $\frac{\partial f(X)}{\partial X_{i,j}}$ and the first-order approximation of f at a point X can be expressed as

$$f(X + \Delta) = f(X) + \langle \nabla f(X), \Delta\rangle + \mathrm{O}(\|\Delta\|).$$

Related to this, we provide some gradients needed later on

$$\nabla\langle A, X^{\mathrm{T}}X\rangle = X(A + A^{\mathrm{T}}), \tag{3.5}$$
$$\nabla\langle X^{\mathrm{T}}AX, B\rangle = AXB^{\mathrm{T}} + A^{\mathrm{T}}XB. \tag{3.6}$$

By setting the partial gradients of $L(U, V, X, Y)$ to zero (using (3.5) and (3.6)), the first order conditions can be expressed by

$$UX = \nabla_U f(U, V) := A^{\mathrm{T}}U(V^{\mathrm{T}}BV) + AU(V^{\mathrm{T}}B^{\mathrm{T}}V), \tag{3.7}$$
$$VY = \nabla_V f(U, V) := B^{\mathrm{T}}V(U^{\mathrm{T}}AU) + BV(U^{\mathrm{T}}A^{\mathrm{T}}U), \tag{3.8}$$

with $X = Y$ because of the symmetry of both matrices (see [6]). The cost function (2.3) is independent of a common transformation $(U, V) \to (UQ, VQ)$ with Q orthogonal. This degree of freedom can be used to choose $X = Y = \Lambda$ to be diagonal. Notice that the above conditions are equivalent to

$$(I - UU^{\mathrm{T}})\nabla_U f(U, V) = 0, \tag{3.9}$$
$$(I - VV^{\mathrm{T}})\nabla_V f(U, V) = 0, \tag{3.10}$$

which are now independent of X and Y. These conditions will also be used later to check convergence to a stationary point.

For special matrices, the optimal value of (2.3) can be found analytically. For the underlying case of non-symmetric matrices and $k > 1$, only bounds for the solution of the problem could be formulated [6]. This motivates the study of numerical methods for (2.3).

3.2. Numerical approximation by optimization over equinormed sets

In this section we develop an optimization technique for compact equinormed sets. Let us first provide a definition of an equinormed set.

Definition 2. We call the compact set $\mathscr{Q} \subset \mathbb{R}^{m\times n}$ equinormed for a given norm $\|\cdot\|$ if

$$\|X\| = r \quad \forall X \in \mathscr{Q}.$$

The value r is called the radius of $\mathscr{Q}$.

Clearly, a nontrivial equinormed set *cannot* be convex. Our main problem of interest will be to find a local minimum of a smooth function $f(X)$ over an equinormed set $\mathscr{Q}$:

$$\min_X \{f(X) : X \in \mathscr{Q}\}. \tag{3.11}$$

Since $\mathscr{Q}$ is non-convex, we are restricted in the optimization steps that are implementable over this set. In this paper, we consider the sets for which the optimization problem

$$\lambda_{\mathscr{Q}(C)} = \max_X \{\langle C, X\rangle : X \in \mathscr{Q}\} \tag{3.12}$$

with $C \in \mathbb{R}^{m\times k}$ is simple and can easily be solved. Denote by $\mathscr{L}_{\mathscr{Q}(C)}$ any of its global solutions. They always exist since $\mathscr{Q}$ is compact and $\langle C, X\rangle$ is continuous.

In order to find a local solution to (3.11), we use the concept of *gradient mapping* defined in [8]. Denote by $\pi_{\mathscr{Q}}(X, \ell) \in \mathscr{Q}$, with ℓ a scalar, any point from the set of *global* maxima

$$\operatorname{Arg}\max_{Y\in\mathscr{Q}} [\Phi_\ell(X, Y) \overset{\text{def}}{=} \langle \nabla f(X), X - Y\rangle - \tfrac{1}{2}\ell\|Y - X\|^2]. \tag{3.13}$$

Then the gradient mapping $\phi_\ell(X)$ is defined as $\phi_\ell(X) = \Phi_\ell(X, \pi_{\mathscr{Q}}(X, \ell))$.

Applied to connected equinormed sets, we obtain the following main properties of the gradient mapping. Note that we impose here the additional condition that $\mathscr{Q}$ is connected.

Lemma 3. *Let $X \in \mathscr{Q}$, $\mathscr{Q}$ be connected and $\ell > 0$. Then $\phi_\ell(X) \geqslant 0$. Moreover, $\phi_\ell(X) = 0$ implies that X satisfies the first-order optimality conditions for problem* (3.11).

Proof. Since the choice $Y = X$ yields $\Phi_\ell(X, Y) = 0$ in (3.13), we get the first statement of the lemma. Further, condition $\phi_\ell(X) = 0$ implies

$$\langle \nabla f(X), Y - X\rangle \geqslant -\frac{1}{2}\ell\|Y - X\|^2 \quad \forall Y \in \mathscr{Q}. \tag{3.14}$$

Denote by $\mathscr{N}_{\mathscr{Q}}(X)$ the set of directions at some $X \in \mathscr{Q}$:

$$\mathscr{N}_{\mathscr{Q}}(X) = \left\{ U : U = \lim_{\substack{Y\to X,\\ Y\in\mathscr{Q}\setminus\{X\}}} \frac{Y - X}{\|Y - X\|} \right\}.$$

Dividing both sides of (3.14) by $\|Y - X\|$ and taking the limit as $Y \to X$ in the connected set $\mathscr{Q}$, we get $\langle \nabla f(X), U\rangle \geqslant 0$ for all $U \in \mathscr{N}_{\mathscr{Q}}(X)$. That is the first-order optimality condition for problem (3.11) at X. □

Lemma 4. *Let the function f have a Lipschitz-continuous gradient*:

$$\|\nabla f(X) - \nabla f(Y)\| \leqslant m\|X - Y\|, \quad \forall X, Y \in \overline{\mathscr{Q}} \overset{\text{def}}{=} \operatorname{Conv}(\mathscr{Q}), \tag{3.15}$$

with a certain $m > 0$. *Then for any* $X \in \mathscr{Q}$ *and* $\ell \geqslant m$ *we have*

$$f(X) - f(\pi_{\mathscr{Q}}(X, \ell)) \geqslant \phi_\ell(X). \tag{3.16}$$

Proof. From (3.15), it follows that [8]

$$f(Y) \leqslant f(X) + \langle \nabla f(X), Y - X\rangle + \frac{1}{2}\ell\|Y - X\|^2$$

for all $Y \in \overline{\mathscr{Q}}$ and $\ell \geqslant m$. Taking $Y = \pi_{\mathscr{Q}}(X, \ell)$, we get (3.16). □

The statement of the last lemma leads to the following *gradient method*: choose $X_0 \in \mathscr{Q}$, $\ell \geqslant m$ and iterate

$$X_{i+1} = \pi_{\mathscr{Q}}(X_i, \ell), \quad i \geqslant 0. \tag{3.17}$$

The following theorem expresses a convergence statement of the algorithm.

Theorem 1. *The process* (3.17) *decreases the objective function in a monotone way. Moreover the set of* $\mathscr{Q}$ *is bounded, then the set of accumulation points* $\mathscr{X}^*$ *of the sequence* $\{X_i\}_{i=0}^{\infty}$ *is nonempty. For any* $X_* \in \mathscr{X}^*$ *we have the first-order optimality conditions satisfied*: $\phi_\ell(X_*) = 0$.

Proof. Indeed, since $\mathscr{Q}$ is bounded, f is bounded below on $\mathscr{Q}$ by some constant f_*. Therefore, by (3.16) we have

$$\sum_{i=0}^{k} \phi_\ell(X_i) \leqslant f(X_0) - f(X_k) \leqslant f(X_0) - f_*.$$

Hence, $\lim_{k\to\infty} \phi_\ell(X_k) = 0$. It remains to use compactness of $\mathscr{Q}$. □

Note that for general sets the auxiliary optimization problem (3.13) can be very difficult. However, in the case of an equinormed feasible set, the situation is much better.

Lemma 5. *Let* $\mathscr{Q}$ *be an equinormed set. Then* $\pi_{\mathscr{Q}}(X, \ell) = \mathscr{L}_{\mathscr{Q}}(\ell X - \nabla f(X))$.

Proof. Indeed, denote by r the radius of the set $\mathscr{Q}$. Then for X and Y from $\mathscr{Q}$ the objective function of problem (3.13) can be rewritten in the following way:

$$\begin{aligned}\langle \nabla f(X), X - Y\rangle - \frac{1}{2}\ell\|Y - X\|^2 &= \langle \nabla f(X), X - Y\rangle - \frac{1}{2}\ell[2r^2 - 2\langle X, Y\rangle] \\ &= \langle \ell X - \nabla f(X), Y\rangle + \langle \nabla f(X), X\rangle - \ell r^2. \quad □\end{aligned}$$

The gradient method (3.17), combined with Lemma 5, will be used for a numerical approximation of the solution of (2.3). In view of Theorem 1, this process ensures a monotone decrease of the objective function. Moreover any accumulation point of the generated sequence satisfies the first-order optimality condition. The convergence is therefore guaranteed.

4. Computational aspects

In this section we compare two iterative algorithms to find a critical point of (2.3). The first one is a simple recursive algorithm based on the singular value decomposition (SVD) for which, unfortunately, the convergence has not yet been proved. The second algorithm is the gradient method of Section (3.2). This method is guaranteed to converge to a minimum of the function. We also compare the cost of the two algorithms.

4.1. Algorithms

The iterative algorithm developed in [6] to solve (2.3) is the following. Choose initial isometries U_0, V_0 and compute at each step:

$$U_{i+1}\Sigma_{+_{i+1}}V_{i+1}^{\mathrm{T}} + U_{\perp_{i+1}}\Sigma_{-_{i+1}}V_{\perp_{i+1}}^{\mathrm{T}} = AU_iV_i^{\mathrm{T}}B^{\mathrm{T}} + A^{\mathrm{T}}U_iV_i^{\mathrm{T}}B + aU_iV_i^{\mathrm{T}} \tag{4.18}$$

where $U_\perp$ and $V_\perp$ are orthogonal complements of U and V ($U^{\mathrm{T}}U_\perp = 0$, U has dimensions $m \times k$ and $U_\perp$ dimensions $m \times (m-k)$). The scalar a is a positive number sufficiently large and Σ_+ is a diagonal matrix with positive elements. The subscripts $i+1$ represent the new stage of the iteration while the subscript i is the current stage. In the iteration, $U_{i+1}\Sigma_{i+1}V_{i+1}^{\mathrm{T}}$ is the best approximation of rank k of $AU_iV_i^{\mathrm{T}}B^{\mathrm{T}} + A^{\mathrm{T}}U_iV_i^{\mathrm{T}}B + aU_iV_i^{\mathrm{T}}$. In practice the iteration (4.18) is realized by computing an SVD of the right-hand side of the equality. The following step is applied iteratively for $i = 0, 1, \ldots$ until the error estimate $stepsize$ gets below a certain tolerance.

Basic iteration step:

(1) $U\Sigma V^{\mathrm{T}} = AU_iV_i^{\mathrm{T}}B^{\mathrm{T}} + A^{\mathrm{T}}U_iV_i^{\mathrm{T}}B + aU_iV_i^{\mathrm{T}}$,

(2) $U_{i+1} = U\begin{bmatrix} I_k \\ 0 \end{bmatrix}$; $V_{i+1} = V\begin{bmatrix} I_k \\ 0 \end{bmatrix}$,

(3) $stepsize_{i+1} = \|U_{i+1}V_{i+1}^{\mathrm{T}} - U_iV_i^{\mathrm{T}}\|$.

When the product $U_iV_i^{\mathrm{T}}$ converges (i.e. $\|U_{i+1}V_{i+1}^{\mathrm{T}} - U_iV_i^{\mathrm{T}}\| \to 0$), there exists a diagonal matrix $\Lambda = \Sigma_+ - aI_k$ such that

$$U\Lambda V^{\mathrm{T}} = AUV^{\mathrm{T}}B^{\mathrm{T}} + A^{\mathrm{T}}UV^{\mathrm{T}}B - U_\perp\Sigma_-V_\perp^{\mathrm{T}}, \tag{4.19}$$

where Σ_- is a diagonal matrix with elements which are all smaller than the elements of Σ_+. In practice, a must be sufficiently large to make $\Sigma_+ = \Lambda + aI_k$ positive. It can be shown that taking $a \geqslant 4\|A\|\|B\|$ guarantees this condition [6]. The convergence is not proved but in all experiments the process always converged linearly to a stationary point. It was shown that a pair (U, V) satisfies (4.19) if and only if it solves the optimization problem (2.3) [6].

The second algorithm is based on the gradient method developed in Section 3.2 and applied to (2.3). Notice that the gradients are Lipschitz-continuous with a certain $\ell > 0$. This algorithm takes into account the following simplifications.

- For any $X \in \mathcal{Q}_{m,k}$ we have $\|X\|^2 = \langle X, X\rangle = \langle X^{\mathrm{T}}X, I_k\rangle = k$. Thus, $\mathcal{Q}_{m,k}$ is an equinormed set with radius $\sqrt{k}$.
- Let $C \in \mathbb{R}^{m\times k}$ be of full row rank. Consider $\mathcal{Q}_{m,k}$. It is easy to see that

$$\mathscr{L}_{\mathcal{Q}_{m,k}}(C) = C[C^{\mathrm{T}}C]^{-1/2}. \tag{4.20}$$

Moreover, $\lambda_{\mathcal{Q}_{m,k}}(C) = \langle [C^{\mathrm{T}}C]^{1/2}, I_k\rangle = \mathrm{tr}([C^{\mathrm{T}}C]^{1/2})$.

Therefore, we can choose initial isometries U_0, V_0 and iterate, for $i \geqslant 0$:

$$\begin{cases} Y_{1i} = \ell U_i - \nabla_U f(U_i, V_i), \\ Y_{2i} = \ell V_i - \nabla_V f(U_i, V_i), \\ U_{i+1} = Y_{1i}[Y_{1i}^{\mathrm{T}} Y_{1i}]^{-1/2}, \\ V_{i+1} = Y_{2i}[Y_{2i}^{\mathrm{T}} Y_{2i}]^{-1/2}. \end{cases} \tag{4.21}$$

In practice, because U_{i+1} and V_{i+1} are respectively the polar factors of Y_{1i} and Y_{2i}, the following iteration step is applied iteratively until $stepsize$ gets below a certain tolerance.

Basic iteration step:

(1) $Y_{1i} = \ell U_i + (AU_i V_i^{\mathrm{T}} B^{\mathrm{T}} V_i + A^{\mathrm{T}} U_i V_i^{\mathrm{T}} B V_i), Y_{2i} = \ell V_i + (B V_i U_i^{\mathrm{T}} A^{\mathrm{T}} U_i + B^{\mathrm{T}} V_i U_i^{\mathrm{T}} A U_i),$

(2) $P_1 \Sigma_1 Q_1^{\mathrm{T}} = Y_{1i}, U_{i+1} = P_1 \begin{bmatrix} I_k \\ 0 \end{bmatrix} Q_1^{\mathrm{T}},$

(3) $P_2 \Sigma_2 Q_2^{\mathrm{T}} = Y_{2i}, V_{i+1} = P_2 \begin{bmatrix} I_k \\ 0 \end{bmatrix} Q_2^{\mathrm{T}},$

(4) $stepsize_{i+1} = \|U_{i+1} V_{i+1}^{\mathrm{T}} - U_i V_i^{\mathrm{T}}\|.$

The scalar ℓ is obtained as follows: we choose a small initial value of ℓ and, after each step, we check if the condition (3.15) is satisfied. In the positive case, we update the variables and we compute the next step of the algorithm. In the negative case, we increase the value of ℓ by a factor 2 and start again the current step of the iteration until the condition (3.15) is verified. Thus, we can recover from a bad choice of this parameter after a small number of short steps which depends logarithmically on our initial guess.

In view of Theorem 1 and contrary to the SVD algorithm, the convergence of the gradient method to a fixed point is guaranteed.

4.2. *Complexity*

In this section, we analyze the complexity of the two algorithms in order to compare their cost (see [7] for more details about the amount of work for some matrix operations or decompositions). The costs rely on the following basic operations.

(1) A product between two matrices M_1, M_2 of dimensions $m_1 \times m_2$ and $m_2 \times m_3$ respectively. It is well known that this requires

$2m_1m_2m_3$ flops,

where a flop is a basic arithmetic operation like an addition or a multiplication. If the matrix M_1 is sparse and has e.g. α nonzero elements in total, then the number of required operations becomes

$2\alpha m_3$ flops.

(2) The SVD of a $m \times n$ matrix N given in a factored form

$N = N_1 \cdot N_2^{\mathrm{T}}$

where N_1 is of dimension $m \times k$ and N_2 of dimension $n \times k$, with $k \ll (m, n)$. The procedure recommended in [4] is to first compute the QR factorizations

$N_1 = Q_1 R_1, \qquad N_2 = Q_2 R_2$

and then the SVD of the $k \times k$ matrix

$$R_1 R_2^{\mathrm{T}} = U_k \Sigma_k V_k^{\mathrm{T}}.$$

The complexity of this approach is given in [4] or [7] and equals

$$6mk^2 + 6nk^2 + \mathrm{O}(k^3)\text{flops}.$$

We now apply this to our two algorithms. In the following, we assume that $n \sim m$ and $k \ll (m, n)$.

4.2.1. *Cost per iteration step for the SVD algorithm*

The cheapest way to compute the first algorithm is to proceed as follows:

(1) construct

$$N_1 = [AU_i | A^{\mathrm{T}} U_i | aU_i],$$
$$N_2 = [BV_i | B^{\mathrm{T}} V_i | V_i],$$

which requires respectively $4\alpha k + mk$ and $4\beta k$ flops, where α and β are respectively the number of nonzero elements of A and B,

(2) compute the SVD of a matrix given in a factored form as explained above, where N_1 and N_2 are respectively of dimensions $m \times 3k$ and $n \times 3k$. This requires $6m(3k)^2 + 6n(3k)^2 + \mathrm{O}(k^3)$ flops and provides U_{i+1} and V_{i+1}.

The total amounts of work for one step of this algorithm is thus $4\alpha k + 4\beta k + 54k^2(m + n) + mk + \mathrm{O}(k^3)$. For dense matrices A and B ($\alpha = m^2$, $\beta = n^2$), the complexity is given by $4m^2k + 4n^2k + \mathrm{O}(mk^2, nk^2)$, while for sparse matrices, in general $\alpha = \mathrm{O}(m)$ and $\beta = \mathrm{O}(n)$, and the cost of the algorithm is thus cheaper.

4.2.2. *Cost per iteration step for the gradient algorithm*

The algorithm is computed in the following way:

(1) construct

$$Y_{1i} = \ell U_i + (AU_i V_i^{\mathrm{T}} B^{\mathrm{T}} V_i + A^{\mathrm{T}} U_i V_i^{\mathrm{T}} BV_i),$$
$$Y_{2i} = \ell V_i + (BV_i U_i^{\mathrm{T}} A^{\mathrm{T}} U_i + B^{\mathrm{T}} V_i U_i^{\mathrm{T}} AU_i),$$

which requires $4\alpha k + 4\beta k + 2k^2(m + n) + 3k(m + n)$ flops,

(2) compute the modified SVD of Y_{1i} and Y_{2i} which results in $6mk^2 + 6nk^2 + \mathrm{O}(k^3)$ flops,

(3) determine U_{i+1} and V_{i+1} which requires $2mk^2 + 2nk^2$ flops.

The total amounts of work for one step of this algorithm equals $4\alpha k + 4\beta k + 10k^2(m + n) + 3k(m + n) + \mathrm{O}(k^3)$. For dense matrices A and B ($\alpha = m^2$, $\beta = n^2$), the complexity is given by $4m^2k + 4n^2k + \mathrm{O}(mk^2, nk^2)$ which is similar to the cost per iteration step for the SVD algorithm. On the other hand, for sparse matrices, the cost per iteration step for the gradient algorithm is cheaper than the SVD one. Let us remark that adjacency matrices of graphs are often sparse.

5. Numerical experiments

In this section we present some numerical experiments. First a comparison of the behavior and the cost of both algorithms is realized. Secondly an application to graph matching is proposed to illustrate the usefulness of numerically approximate (2.3).

5.1. *Comparison of the two algorithms*

In the aim to compare both algorithms, we apply them to randomly generated matrices A and B of dimensions $m = n = 50$ for $k = 2$. The matrices are of different types: sparse non-symmetric and sparse symmetric with 150 nonzero elements in total, dense non-symmetric and finally dense symmetric. For each type of matrices, we ran 100 experiments and took the mean number of steps over all 100 computations. We imposed a termination criterion based on the convergence measure that the gradients must be perpendicular to the manifold (see (3.9) and (3.10)):

$$\mathrm{error}_i = \sqrt{\|(I - U_i U_i^{\mathrm{T}})\nabla_U F(U_i, V_i)\|^2 + \|(I - V_i V_i^{\mathrm{T}})\nabla_V F(U_i, V_i)\|^2}. \tag{5.22}$$

Initial isometries U_0 and V_0 are also randomly generated, but are the same for the computation of both algorithms. The results are described in Tables 1–4. The mean number of flops for each case is obtained by multiplying the mean number of steps with the theoretical number of flops per iteration step described in Section 4.2. The process always converges to a fixed point, but this fixed point could be different for the two algorithms. Tables 1–4 give also the number of similar matchings, i.e. the number of times (over the 100 experiments) that both algorithms converge to the same fixed point. Note that in these experiments the algorithms often converge to different fixed points. This is probably due to the fact that for random matrices we can not expect to have a global maximum with a large basin of attraction and hence that one may easily converge to local minima.

Finally, the mean numbers of flops for the given termination criteria are plotted, for each case, in Fig. 1. We observe the linear convergence of the algorithms. Moreover, in these experiments, the gradient method converges in less steps than the SVD method and appears to be cheaper in three of the four cases.

5.2. *Application to graph matching*

In the aim to test this problem to graph matching, we provide an example on graphs composed of groups of vertices that are strongly connected. A few directed edges link these groups. The

Table 1
Results for sparse non-symmetric matrices

tol	Gradient method		SVD method	
	♯ steps	♯ flops	♯ steps	♯ flops
10^{-2}	94	66×10^4	136	327×10^4
10^{-4}	271	189×10^4	355	857×10^4
10^{-6}	473	331×10^4	608	1465×10^4
10^{-8}	678	475×10^4	874	2106×10^3
10^{-10}	885	619×10^3	1142	2753×10^3
	58 Similar matchings			

Table 2
Results for dense non-symmetric matrices

tol	Gradient method		SVD method	
	♯ steps	♯ flops	♯ steps	♯ flops
10^{-2}	504	2016×10^4	1741	6963×10^4
10^{-4}	1037	4149×10^4	3349	$13,398\times10^4$
10^{-6}	1578	6314×10^4	4995	$19,979\times10^4$
10^{-8}	2121	8483×10^4	6643	$26,570\times10^4$
10^{-10}	2663	$10,653\times10^4$	8290	$33,162\times10^4$
	42 Similar matchings			

Table 3
Results for sparse symmetric matrices

tol	Gradient method		SVD method	
	♯ steps	♯ flops	♯ steps	♯ flops
10^{-2}	236	165×10^4	173	418×10^4
10^{-4}	504	353×10^4	346	834×10^4
10^{-6}	779	545×10^4	523	1260×10^4
10^{-8}	1055	738×10^4	701	1690×10^4
10^{-10}	1331	931×10^4	881	2122×10^4
	70 Similar matchings			

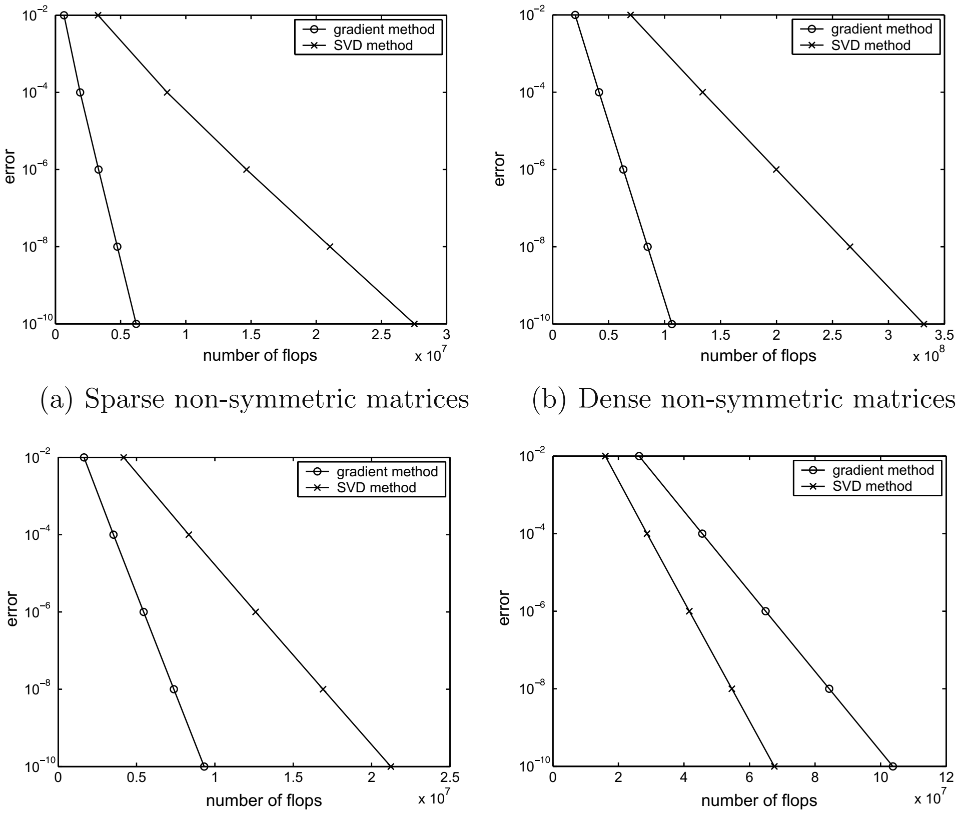
Table 4
Results for dense symmetric matrices

tol	Gradient method		SVD method	
	♯ steps	♯ flops	♯ steps	♯ flops
10^{-2}	236	945×10^4	173	694×10^4
10^{-4}	504	2017×10^4	346	1384×10^4
10^{-6}	779	3115×10^4	523	2092×10^4
10^{-8}	1055	4219×10^4	701	2805×10^4
10^{-10}	1331	5325×10^4	881	3523×10^4
	74 Similar matchings			

projection of the graphs emphasizes the importance of the direction between the clusters in order to compare the graphs.

We generate randomly two directed graphs with three groups of vertices that are strongly connected and with weak connections between these groups. The graphs can be roughly represented by path graphs with three vertices. Each vertex represents a group of strongly connected nodes (see Fig. 2). E.g. the vertex D of the path graph models the group of nodes 1–10 of Graph B. Graph A is composed of three groups of 20 nodes while Graphs B is composed of groups of 10 nodes. The direction of the links between the clusters represented on the path graphs is the dominant direction between these clusters, e.g. there exist two edges from A to B and only one between B and A. The dominant direction is thus from A to B.

In order to compare pairs of graphs, we project Graphs A and B in a 3-dimensional space for an easier visualization of the results ($k = 3$). The matrices A and B represent in this case the graph adjacency matrices. For the given initial isometries U_0 and V_0, the two algorithms converge



(a) Sparse non-symmetric matrices

(b) Dense non-symmetric matrices

(c) Sparse symmetric matrices.

(d) Dense symmetric matrices.

Fig. 1. Mean number of steps for the given termination criteria.

in this case to the same solution U and V. This is probably due to the fact that, for these particular matrices, the objective function has a global maximum with a sufficiently large basin of attraction. The error gets below a tolerance of 10^{-6} in respectively 824 and 2600 iterations for the gradient and the SVD methods.

Because the graphs are of different dimension, we normalize the projections U and V by a row normalization similar to the one from [2], i.e. we project the vertices on a sphere in a k-dimensional space ($k = 3$). We plot the normalized projections U and V, i.e. the first column of U (resp. V) as x, the second column of U (resp. V) as y and the third column of U (resp. V) as z on Fig. 3. The points have the property that similar nodes or groups of nodes are close in the projection space. We remark that strongly connected vertices are close in the projection space and that allows to find clusters in a graph. We observe also that the clusters are matched according to the direction between them, i.e. group A is close to group D, B–E and C–F. The direction of the edges plays thus an important role for the result of this example. Indeed if we had compared the symmetric graphs (by removing the directions of the edges), the group A would compare equally well to D or F without altering the objective function.

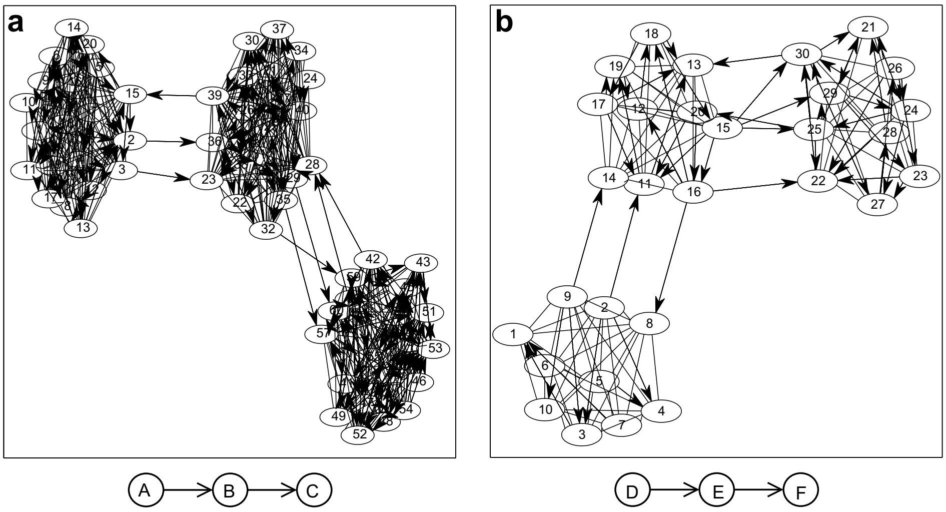


Fig. 2. The graphs and their representation by a path graph: (a) Graph A (60 vertices): the clusters A, B and C represent respectively the nodes 1–20, 21–40 and 41–60; (b) graph B (30 vertices): the clusters D, E and F represent respectively the nodes 1–10, 11–20 and 21–30.

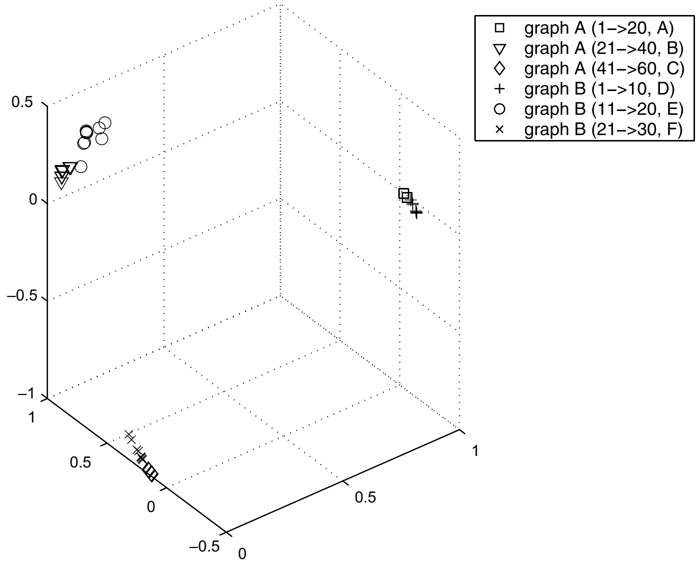


Fig. 3. Result of the projection of graphs A and B in a 3-dimensional subspace.

6. Conclusion

In this paper, we analyze the problem of maximizing the coupling between two square matrices A and B with respect to isometries U and V, i.e. the problem (2.3). An new iterative algorithm is developed in order to numerically approach the critical points of the optimization problem.

It consists of a gradient-type procedure based on the concept of gradient mapping applied to an optimization problem over equinormed set. Numerical experiments showed that in most of the cases the gradient algorithm outperforms the SVD-based algorithm presented in [6]. We also applied the method to the problem of directed graph matching. Our technique retrieves clusters in graphs depending on the direction of the edges. Further investigations of this problem can be pursued in several directions. An interesting perspective in the field of graph matching is to develop a systematic measure of similarity between the projected graphs. In this paper, the similarity was verified on a plot by the proximity of the projected nodes in a k-dimensional space. But there is a need for a proper distance measure in order to obtain a systematic procedure for comparing and classifying graphs.

References

[1] V.D. Blondel, A. Gajardo, M. Heymans, P. Senellart, P. Van Dooren, A measure of similarity between graph vertices: applications to synonym extraction and Web searching, SIAM Rev. 46 (4) (2004) 647–666.

[2] T. Caelli, S. Kosinov, An eigenspace projection clustering method for inexact graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 26 (4) (2004) 515–519.

[3] T. Caelli, S. Kosinov, Inexact graph matching using eigen-subspace projection clustering, Intern. J. Pattern Recog. Artif. Intell. 18 (3) (2004) 329–354.

[4] T. Chan, An improved algorithm for computing the singular value decomposition, ACM Trans. Math. Soft. 8 (1) (1982) 72–83.

[5] D. Cvetkovic, P. Rowlinson, S. Simic, Eigenspaces of Graphs, Cambridge University Press, Cambridge, 1997.

[6] C. Fraikin, Yu. Nesterov, P. Van Dooren. Optimizing the coupling between two isometric projections of matrices, SIAM J. Matrix Anal. Appl., accepted for publication.

[7] G.H. Golub, C.F. Van Loan, Matrix Computations. The Johns Hopkins University Press, Baltimore, MD, second ed., 1989.

[8] Yu. Nesterov, Introductory Lectures on Convex Optimization A Basic Course, Kluwer, Boston/Dordrecht/London, 2004.

[9] G. Scott, H. Longuet-Higgins, An algorithm for associating the features of two patterns, Proc. Royal Soc. London B 244 (1988) 21–26.

[10] L. Shapiro, J. Brady, Feature-based correspondence – an eigenvector approach, Image Vision Comput. 10 (1992) 268–281.

[11] S. Umeyama, An eigen decomposition approach to weighted graph matching problems, IEEE Trans. Pattern Anal. Mach. Intell. 10 (5) (1988) 695–703.

[12] L. Xu, I. King, A PCA approach for fast retrieval of structural patterns in attributed graphs, IEEE Trans. Systems Man Cybernetics 31 (5) (2001) 812–817.