# Iterative Methods for Low Rank Approximation of Graph Similarity Matrices

**Keywords**: graph theory, node to node similarity, algorithm, trace maximization, low-rank approximation

**Thomas P. Cason*** THOMAS.CASON@UCLOUVAIN.BE
**Pierre-Antoine Absil*** PA.ABSIL@UCLOUVAIN.BE
**Paul Van Dooren*** PAUL.VANDOOREN@UCLOUVAIN.BE

* Université catholique de Louvain, Department of Mathematical Engineering
Bâtiment Euler, avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium

## Abstract

In this paper, we analyze an algorithm to compute a low-rank approximation of the similarity matrix between nodes of two graphs introduced by Blondel *et al.* in (Blondel et al., 2004). This problem can be reformulated as an optimization problem of a continuous function $\Phi(S) = \mathrm{tr}\big(S^T \mathcal{M}^2(S)\big)$ where $S$ is constrained to have unit Frobenius norm, and $\mathcal{M}^2$ is a non-negative linear map. We restrict the feasible set to the smooth manifold of rank $k$ matrices with unit Frobenius norm and $k$ identical singular values. We analyze the convergence properties of our algorithm and prove that accumulation points are stationary points of $\Phi(S)$. We finally compare our method in terms of speed and accuracy to the full rank algorithm proposed in (Blondel et al., 2004). [1]

## 1. From Similarity to Optimization

Graphs are a powerful tool for many practical problems such as pattern recognition, shape analysis, image processing and data mining. Measures of graph similarity have a broad array of applications, including comparing chemical structures, navigating complex networks like the World Wide Web, and analyzing different kinds of biological data (Zager, 2005).

The $(i, j)$ entry of a similarity matrix expresses how node $i$ from a graph $G_A$ looks like node $j$ from a graph $G_B$ according to some similarity criterion. Similarity matrices can, for instance, be used to extract synonyms out of a dictionary.

Blondel *et al.* introduced the notion of similarity between nodes of two graphs in (Blondel et al., 2004). They proposed to use the following iteration

$$S_+ = \mathcal{M}(S) := ASB^T + A^T SB, \tag{1}$$

where $A$ and $B$ are the adjacency matrices respectively associated to graph $G_A$ and $G_B$ and defined a similarity measure as a fixed point of the normalized iterates. One can see that after one iteration the $(i, j)$ entry of $S_+$ is given by

$$\sum_{k,l} A_{ik} S_{kl} B_{jl} + A_{ki} S_{kl} B_{lj} = \sum_{\substack{k\leftarrow i \\ l\leftarrow j}} S_{kl} + \sum_{\substack{k\rightarrow i \\ l\rightarrow j}} S_{kl}.$$

The first term of the right-hand side sums all $(k, l)$ entries of $S$ such that node $k$ is a child of node $i$ in $G_A$ and node $l$ is a child of node $j$ in $G_B$. Hence the more the children of node $i$ and $j$ are similar, the more $(S_+)_{ij}$ increases. Similarly, the analysis of the second term of the right-hand side yields that the more the parents of node $i$ and $j$ are similar, the more $(S_+)_{ij}$ increases.

Let $\mathrm{vec}(S)$ denote the vector containing the successive columns of the matrix $S$. One can show that

$$\mathrm{vec}(S_+) = M\mathrm{vec}(S) := \big(B \otimes A + B^T \otimes A^T\big)\,\mathrm{vec}(S) \tag{2}$$

where $\otimes$ is the Kronecker product. This iteration is in fact the power method applied to the matrix $M$.

[1] This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

Presented at *ILP-MLG-SRL*, Leuven, Belgium, 2009.

This matrix is non-negative and hence, according to the Perron-Frobenius theorem, there exists a real positive eigenvalue $\rho$ called the Perron root such that any other eigenvalue $\lambda$ satisfies $\lambda \leq \rho$. Since $M$ is symmetric, its eigenvalues are real and can hence only have two extremal eigenvalues, $\rho$ and possibly $-\rho$, and the even iterates of the following recurrence

$$S_0 = \mathbf{1}_{m,n} \ , \qquad S_{k+1} = \mathcal{M}(S_k)/\left\|\mathcal{M}(S_k)\right\| , \tag{3}$$

converge to a unique positive solution. Indeed, $\mathcal{M}^2$ has a unique dominant eigenvalue $\rho^2$ and the iterates converge towards the normalized orthogonal projection of $S_0$ onto the eigenspace associated to $\rho^2$. This solution achieves the extremal value of the eigenvalue equation $\rho^2 S = \mathcal{M}^2(S)$.

Eigenvalue problems can be seen as optimization problems whose stationary points are solution of the eigenvalue problem. Indeed, the stationary points of the following optimization problem

$$\max_{\langle S,S\rangle=1} \left\langle S, \mathcal{M}^2(S)\right\rangle \tag{4}$$

are such that the gradient of its Lagrangian vanishes

$$\operatorname{grad}_S \ L(S,\lambda) \equiv \mathcal{M}^2(S) - \lambda S = 0 \ , \tag{5}$$

which is precisely our eigenvalue problem with $\lambda = \rho^2$.

When $S$ is large, the eigenvalue problem becomes relatively expensive in terms of computational cost. Hence one can think to modify the problem in order to find an approximation of $S$ at lower cost. In this paper, we consider considers the approximation of the similarity matrix $S$ by a matrix of the form $\frac{UV^T}{\|UV^T\|}$ where $U^TU = I_k = V^TV$. This is a rank $k$ matrix with $k$ identical singular values. The norm of $UV^T$ is equal to $k$, but the analysis we further carry out is equivalent up to a scalar multiplication, hence we will for readability consider matrices of the form $UV^T$ in our analysis.

## 2. The Problem and its Geometry

Let us first clearly state the optimization problem

> Let $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$. Find the maximizer of
>
> $$\Phi : \mathcal{S} \to \mathbb{R} : S \mapsto \Phi(S) = \operatorname{tr}\left(S^T\, \mathcal{M}^2(S)\right) \ ,$$
>
> with $\mathcal{M}(S) = ASB^T + A^TSB$, and
>
> $$\mathcal{S} = \left\{UV^T \in \mathbb{R}^{n\times m} : U^TU = I_k = V^TV\right\} \ .$$

This is an optimization problem of a continuous function $\Phi$ on a compact domain. Notice that the constraint set constitutes a smooth manifold $\mathcal{S} = \left\{UV^T \in \mathbb{R}^{n\times m} : (U,V) \in \operatorname{St}(k,m) \times \operatorname{St}(k,n)\right\}$ where

$$\operatorname{St}(k,m) = \left\{U \in \mathbb{R}^{m\times k} : U^TU = I_k\right\}$$

denotes the compact Stiefel manifold (see (Cason et al., 2008) for details).

There always exists a solution $S$ optimizing the function $\Phi$ such that the first-order optimality condition, $\operatorname{grad}\Phi(S) = 0$, is satisfied

$$\begin{aligned}\mathcal{M}^2(UV^T) - U\operatorname{Sym}\left(U^T\mathcal{M}^2(UV^T)V\right)V^T \\ -(I_m - UU^T)\,\mathcal{M}^2(UV^T)\,(I_n - VV^T) = 0\end{aligned} \tag{6}$$

where $\operatorname{Sym}(\cdot) : X \mapsto \left(X + X^T\right)/2$.

The points satisfying equation (6) are called stationary points. Equation (6) is equivalent to say that stationary points are such that

$$\mathcal{M}^2(UV^T) = UHV^T + U_\perp K V_\perp^T \ , \tag{7}$$

where $H \in \mathcal{S}_{Sym}(k)$, the set of symmetric matrices of order $k$, and where $U_\perp$ and $V_\perp$ are any orthogonal complement of respectively $U$ and $V$.

## 3. Algorithm and Convergence Analysis

We propose the following algorithm to find stationary points of $\Phi$

$$S_+ = U_+V_+^T := \arg \max_{\substack{\tilde{S}=\tilde{U}\tilde{V}^T \\ \tilde{U}^T\tilde{U}=I_k \\ \tilde{V}^T\tilde{V}=I_k}} \left\langle \tilde{S}, \mathcal{M}^2(S)\right\rangle \tag{8}$$

One iteration of (8) costs

$$6\ (m^2+n^2)\ k + 17\ (m+n)\ k^2 + 320\ k^3$$

whereas one full rank iteration costs $4(m^2n + n^2m)$.

Let $\mathcal{M}^2(S)$ have an ordered singular value decomposition

$$\mathcal{M}^2(S) = \begin{bmatrix}P_1 & P_2\end{bmatrix}\begin{bmatrix}\Sigma_1 & 0 \\ 0 & \Sigma_2\end{bmatrix}\begin{bmatrix}Q_1^T \\ Q_2^T\end{bmatrix} = P\Sigma Q^T \tag{9}$$

with $P_1 \in \mathbb{R}^{m\times k}$, $P_2 \in \mathbb{R}^{m\times(m-k)}$, $Q_1 \in \mathbb{R}^{n\times k}$, $Q_2 \in \mathbb{R}^{n\times(n-k)}$, $\Sigma_1 \in \mathbb{R}^{k\times k}$ and $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$. We prove that the maximum is achieved when $S = P_1Q_1^T$ and moreover, if $\nu := \sigma_{min}(\Sigma_1) - \sigma_{max}(\Sigma_2) > 0$, then this solution is unique.

If $\nu = 0$, the iteration (8) is not well defined. From now on, we only consider $\nu > 0$. This assumption seems realistic since the case $\nu = 0$ has never been observed in our numerical experimentations. Further

analysis will have to be done in order to take care of this special case.

Fixed points of the iteration are such that the corresponding ordered singular value decomposition of $\mathcal{M}^2(UV^T)$ is

$$\mathcal{M}^2(UV^T) = UQ\Sigma_1 Q^T V^T + U_\perp \Sigma_2 V_\perp^T \qquad (10)$$

with $Q$ a square orthogonal matrix. This expression satisfies equation (6) and fixed points are hence a stationary point.

We prove that the iteration is an ascent iteration for $\Phi$, and more precisely

$$\Phi(S_+) - \Phi(S) \geq \|\mathcal{M}(\Delta_S)\|^2 + \nu \|\Delta_S\|^2 \qquad (11)$$

with $\Delta_S := S_+ - S$, and $\nu := \sigma_{min}(\Sigma_1) - \sigma_{max}(\Sigma_2)$.

Moreover, if $S$ is a non-stationary point of $\Phi$, then there exists an $\epsilon > 0$ such that all $S_\epsilon$ in the open ball $B_\epsilon(S)$ are not stationary points and since $\Phi(S_+) - \Phi(S)$ is a continuous function of $S$, equation (11) guarantees a non-zero increase of the iteration within that ball. Eventually, a careful analysis allows to conclude that every accumulation point of the sequence of iterates is a stationary point of $\Phi$. We experimentally observe that accumulation points are local maxima of $\Phi$.

## 4. Numerical Experiments

We observe that when the rank of the approximation increases, the relative error of the approximation increases and the rate of convergence of the algorithm decreases. These counterintuitive results occur because similarity matrices do not usually have identical eigenvalues. We hence enhance our method with a diagonal positive scaling $D$, *i.e.*

$$S_+ = U_+ D_+ V_+^T := \arg \max_{\substack{\tilde{S}=\tilde{U}\tilde{D}\tilde{V}^T,\ \tilde{D}\geq 0 \\ \tilde{U}^T\tilde{U}=I_k=\tilde{V}^T\tilde{V}}} \left\langle \tilde{S}, \mathcal{M}^2(S) \right\rangle .$$

The problems are structurally different: whereas the set $\mathcal{S}$ is a nice compact submanifold, the feasible set on which we run the experiments, is only a stratified space. We are working on the proof of convergence of this algorithm. The results found in the previous section should be extendable to this modified case.

We look at the performance of this algorithm to compute self-similarity matrices. This means that $A$ and $B$ are equal, and the self-similarity matrix expresses how a node of a graph is similar to other nodes of the same graph. We run several experiments to compute rank-$k$ approximations of self-similarity matrices on random graphs. Figure 1 shows the average computational time. As expected, we clearly notice that the smaller the rank of the approximation $k$, the smaller the computational time. We further notice that, when the order of the graph increases, the algorithm for low-rank approximation converges faster than the full rank algorithm.

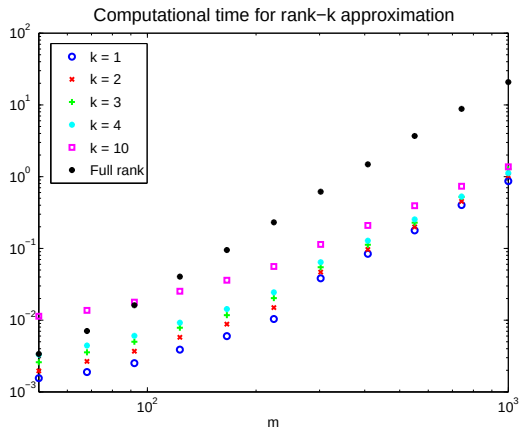


*Figure 1.* This figure shows the average time to compute rank-$k$ approximations of the self-similarity matrix of a connected random graph versus $m$, the order of this graph. The graph is built such that the average number of outgoing edges of a node is 10. The algorithm stops when $\|\Delta_S\| \leq 10^{-6} \|S\|$. The full rank results are obtained using equation (3) which was analyzed in (Blondel et al., 2004).

As far as the relative error is concerned, we observe that it does not vary much with the order of the graph. As expected the relative error decreases when the rank of the approximation increases. For $m = 1000$, we have

| $k$ | 1 | 2 | 3 | 4 | 10 |
|---|---|---|---|---|---|
| Error | $4.5\text{E}-2$ | $3\text{E}-3$ | $2\text{E}-3$ | $2\text{E}-4$ | $1\text{E}-5$ |

These results might be biased since we consider randomly generated graphs. Experiments on real graphs are under investigation.

## References

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

Cason, T., Absil, P.-A., & Van Dooren, P. (2008). Comparing two matrices by means of isometric projections. Summitted to *Numerical Linear Algebra in Signals, Systems and Control*.

Zager, L. (2005). *Graph similarity and matching*. Doctoral dissertation, Massachusetts Institute of Technology.