




VU THI HUONG¹, IDA LITZEL², AND THORSTEN KOCH³

**Similarity-based fuzzy clustering
scientific articles: potentials and
challenges from mathematical and
computational perspectives**

¹  0009-0007-4869-0505

²  0009-0000-0166-7809

³  0000-0002-1967-0077

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30 84185-0
Telefax: +49 30 84185-125

E-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Similarity-based fuzzy clustering scientific articles: potentials and challenges from mathematical and computational perspectives

Vu Thi Huong ¹, Ida Litzel ², and Thorsten Koch ³

June 11, 2025

Dedicated to Professor Yuriy Nesterov on the occasion of his 70th birthday

Abstract. Fuzzy clustering, which allows an article to belong to multiple clusters with soft membership degrees, plays a vital role in analyzing publication data. This problem can be formulated as a constrained optimization model, where the goal is to minimize the discrepancy between the similarity observed from data and the similarity derived from a predicted distribution. While this approach benefits from leveraging state-of-the-art optimization algorithms, tailoring them to work with real, massive databases like OpenAlex or Web of Science – containing about 70 million articles and a billion citations – poses significant challenges. We analyze potentials and challenges of the approach from both mathematical and computational perspectives. Among other things, second-order optimality conditions are established, providing new theoretical insights, and practical solution methods are proposed by exploiting the problem’s structure. Specifically, we accelerate the gradient projection method using GPU-based parallel computing to efficiently handle large-scale data.

Keywords: fuzzy clustering, large-scale publication data, non-convex optimization, second-order optimality, gradient projection methods, Nesterov acceleration, GPU-based parallel computing, bibliometrics

2020 Mathematics Subject Classification: 90C26, 90C30, 90C90, 62H30, 68W10, 68T05, 68T09

¹Digital Data and Information for Society, Science, and Culture, Zuse Institute Berlin, 14195 Berlin, Germany; and Institute of Mathematics, Vietnam Academy of Science and Technology, 10072 Hanoi, Vietnam. Email: huong.vu@zib.de

²Digital Data and Information for Society, Science, and Culture, Zuse Institute Berlin, Germany. Email: Litzel@zib.de

³Software and Algorithms for Discrete Optimization, Technische Universität Berlin, Germany; and Applied Algorithmic Intelligence Methods, Zuse Institute Berlin, Germany. Email: koch@zib.de

1 Introduction

The exponential growth of scientific literature has intensified the need for effective computerized methods to map and understand the structure of research fields. Bibliometric analysis, which quantitatively examines scholarly communication, plays a crucial role in this endeavor; see, e.g., [1]. A key task within bibliometrics is clustering scientific publications into coherent groups that reflect topical, methodological, or intellectual similarities. This facilitates applications such as literature recommendation [2] or identification of emerging research areas [3].

Popular clustering methods, including the Louvain [4] and Leiden [5] algorithms, typically assign each publication to a single cluster. However, in reality, areas often overlap, as seen, for example, in the intersection of theoretical computer science and mathematics, and interdisciplinary research can easily span multiple topics. This motivates the use of *fuzzy clustering*, which allows for overlapping cluster memberships. To the best of our knowledge, we found no prior studies that have specifically applied fuzzy clustering methods to the clustering of scientific publications – a gap that this study aims to address.

Scientific documents exhibit rich structures through both citation-based linkages and textual content, offering multiple complementary perspectives for assessing similarity and forming clusters; see [6, 7]. Citation-based similarities, such as direct citations, co-citations, and bibliographic coupling, capture intellectual relationships, while textual features reflect semantic closeness. These diverse features provide a foundation for clustering but also raise the question of which fuzzy clustering techniques are best suited to integrate such heterogeneous data.

A commonly used approach in fuzzy clustering is the Fuzzy C-Means (FCM) algorithm [8], which has demonstrated success in various domains, including medical image segmentation [9] and pattern recognition [10]. However, FCM requires the vectorization of documents – transforming textual or citation data into numerical vectors – prior to clustering. This preprocessing step can be computationally intensive when dealing with large-scale datasets and also tightly couples representation and clustering.

To overcome these limitations, we adopt the similarity-based fuzzy clustering method proposed by Nepusz et al. [11], which models clustering as a constrained optimization problem minimizing the discrepancy between observed and predicted similarities. Unlike FCM, this approach can operate directly on pairwise similarity matrices derived from citation data, eliminating the need for vectorization. It also allows for the integration of additional features, such as semantic similarities from text, into the similarity matrix without altering the core clustering mechanism.

Nevertheless, adapting the method to large publication databases, such as OpenAlex¹ and Web of Science², which comprise millions of articles and billions of citations, poses substantial challenges. Efficient and scalable algorithms

¹<https://openalex.org>

²<https://www.webofscience.com>

are essential for handling data of this magnitude. In this paper, we address similarity-based fuzzy clustering from both mathematical and computational perspectives. We introduce parallel implementations of the gradient projection algorithm (parallel GPA) and the fast iterative shrinkage-thresholding algorithm (parallel FISTA) for large-scale processing. Both methods yield significant reductions in the clustering objective, with parallel FISTA exhibiting strong heuristic acceleration on graphs with several million nodes. On the theoretical side, we prove the convergence of parallel GPA to critical points and, for the first time, establish second-order optimality conditions, offering insights regarding the quality of the solutions.

A central contribution of our work is a scalable parallelization strategy that leverages the mathematical structure of the fuzzy clustering problem. By decoupling computations across columns and sharing compact intermediate results, our algorithms avoid costly operations on large matrices. Our CUDA-based implementation achieves high throughput, supporting clustering on graphs with millions of nodes.

The remainder of the paper is organized as follows. Section 2 introduces the fuzzy clustering model and the gradient projection algorithm, highlighting the computational challenges. Section 3 details our parallelization techniques, emphasizing efficient gradient computations that enable scalability to millions of articles. Section 4 presents theoretical refinements via second-order optimality conditions. Section 5 reports on experiments across synthetic and real-world datasets, demonstrating the scalability and effectiveness of parallel GPA and parallel FISTA. We summarize our contribution and outline directions for future work in Section 6.

2 Fuzzy clustering with gradient projection

Given a set of N scientific articles, we are interested in the problem of partitioning the set into C clusters such that articles within the same cluster are more similar to each other than to articles in different clusters. To reflect the fact that an article can span multiple research domains, we allow each article to belong to multiple clusters with varying degrees of membership.

Such a partition can be described by a matrix $X = (x_{ki}) \in [0, 1]^{C \times N}$ with each x_{ki} representing the membership degree of the article i w.r.t. the cluster k . When $x_{ki} = 1$, then article i sole belongs to cluster k ; otherwise it participates in every cluster k where $x_{ki} > 0$. For each article, we assume that the total membership degrees over C clusters is 1. Thus, a valid partition is a matrix from the set

$$\Omega := \left\{ X = (x_{ki}) \in [0, 1]^{C \times N} \mid \sum_{k=1}^C x_{ki} = 1, \forall i \in \{1, \dots, N\} \right\}. \quad (1)$$

A partition can reveal a similarity measure among articles. Consider two articles $i, j \in \{1, \dots, N\}$ from a partition matrix $X \in \mathbb{R}^{C \times N}$. For each $k \in$

$\{1, \dots, C\}$, x_{ki} and x_{kj} are membership degrees of the articles i and j , respectively. How much property of the cluster k the two articles share can be quantified by $x_{ki}x_{kj}$, their similarity w.r.t. cluster k . Summing up over all C clusters, we get a similarity between i and j . This means that each partition $X \in \mathbb{R}^{C \times N}$ defines a similarity matrix $Y = (y_{ij}) \in [0, 1]^{N \times N} = X^T X$ with element y_{ij} representing the similarity between two articles i, j .

As our aim is to group articles based on their similarity, we suppose that an observed similarity among articles, $S = (s_{ij}) \in \mathbb{R}^{N \times N}$ is given by the data, for example, through citation- or text-based measures. A good partition should group similar articles in the same cluster. To measure how good a partition X is, we quantify how well the similarity Y derived from partition X approximates the observed similarity S by

$$f(X) = \sum_{i=1}^N \sum_{j=1}^N (s_{ij} - y_{ij})^2 = \sum_{i=1}^N \sum_{j=1}^N \left(s_{ij} - \sum_{k=1}^C x_{ki}x_{kj} \right)^2 = \|S - X^T X\|_F^2,$$

where $\|\cdot\|_F$ stands for the matrix Frobenius norm. The problem of fuzzy clustering the set of N scientific articles into C clusters based on an observed similarity $S = (s_{ij}) \in \mathbb{R}^{N \times N}$ now can be formulated, as in [11], as follows

$$\min_{X \in \Omega} f(X). \quad (\mathcal{P})$$

The gradient projection algorithm (GPA). To design a solution method for (\mathcal{P}) , we first observe that the constraint set Ω is nonempty and compact while the objective function f is continuous and bounded from below (by 0) on $\mathbb{R}^{C \times N}$. Thus, the problem (\mathcal{P}) has a *global optimal solution*, i.e., the (global) optimal solution set

$$\text{Sol}(\mathcal{P}) := \{\bar{X} \in \Omega \mid f(\bar{X}) \leq f(X), \forall X \in \Omega\}$$

is nonempty. While it is desirable to find a global optimum solution, this can be very costly, as general nonconvex optimization is NP -hard. Thus, it is more practical to find a *local optimal solution* of (\mathcal{P}) , i.e., a candidate $\bar{X} \in \Omega$ such that $f(\bar{X}) \leq f(X)$ for any $X \in \Omega$ and “close enough” to \bar{X} . Since the constraint set is convex and the objective function is differentiable on the whole space, a local solution \bar{X} has to satisfy (see, e.g., [12, Theorem 3.24]) the *first-order necessary optimality condition*

$$\bar{X} \in \Omega \quad \text{and} \quad \langle \nabla f(\bar{X}), X - \bar{X} \rangle \geq 0, \quad \forall X \in \Omega, \quad (2)$$

which shows a special interaction between the constraint set Ω and the gradient $\nabla f(\bar{X})$ of the objective function at \bar{X} .

One calls a matrix \bar{X} satisfying (2) a *critical point* of (\mathcal{P}) . Such a matrix can be found by the *gradient projection algorithm* (GPA); see, e.g., [12, Theorem 6.1]. The algorithm starts with initializing a membership matrix $X^0 \in \Omega$, then iterates over

$$X^{n+1} := P_{\Omega}(X^n - \tau_n \nabla f(X^n)), \quad \forall n \geq 1$$

until a stopping condition is met. Each step of the algorithm consists of two ingredients: moving the current state X^n toward the direction of the negative gradient $\nabla f(X^n)$ with a *step size* $\tau_n > 0$, and then projecting onto the constraint set P_Ω to ensure the requirements in (1) are not broken.

Projection onto Ω . The set Ω is nonempty, closed, and convex in $\mathbb{R}^{C \times N}$. Thus, for any matrix $X \in \mathbb{R}^{C \times N}$, the *projection* $P_\Omega(X)$ of X onto the set Ω exists uniquely as

$$P_\Omega(X) := \operatorname{argmin} \{ \|Y - X\|_F^2 \mid Y \in \Omega \},$$

i.e., as the minimizer of a quadratic function over the set Ω . Solving this constrained optimization problem can be avoided by exploiting the structure of Ω .

Observe that the constraint $X \in \Omega$ means that each element x_{ki} of the matrix X must lie in the interval $[0, 1]$, and for each column $\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{Ci})^\top$, its elements must sum to 1. Therefore, Ω can be rewritten as

$$\Omega = \underbrace{\Delta^C \times \Delta^C \times \dots \times \Delta^C}_{N \text{ times}},$$

where

$$\Delta^C := \left\{ \mathbf{y} = (y_1, y_2, \dots, y_C)^\top \in \mathbb{R}^C \mid y_k \geq 0, \forall k, \text{ and } \sum_{k=1}^C y_k = 1 \right\}$$

is the *unit simplex* in \mathbb{R}^C . Consequently, for any $X \in \mathbb{R}^{C \times N}$, the projection of X onto Ω is performed independently column-wise

$$P_\Omega(X) = P_{\Delta^C}(\mathbf{x}_1) \times P_{\Delta^C}(\mathbf{x}_2) \times \dots \times P_{\Delta^C}(\mathbf{x}_N);$$

see, e.g., [13, Proposition 29.3].

The unit simplex Δ^C is a nonempty, closed, and convex subset of the Euclidean space \mathbb{R}^C . For a given vector $\mathbf{x} \in \mathbb{R}^C$, the projection $P_{\Delta^C}(\mathbf{x})$ of \mathbf{x} onto the unit simplex is the unique solution of the problem

$$\min \|\mathbf{y} - \mathbf{x}\|_2^2 \quad \text{subject to} \quad \mathbf{y} \in \Delta^C.$$

Note that if $\mathbf{x} \in \Delta^C$, then $P_{\Delta^C}(\mathbf{x}) = \mathbf{x}$ which means that the projection of \mathbf{x} onto the unit simplex is itself. In general, by using Lagrange multipliers to deal with the constraints of the above minimization problem, one can find the projection by, for example, the Duchi algorithm (Algorithm 1; see [14]).

To find the projection of a given vector \mathbf{x} onto the unit simplex using the Duchi algorithm, we first sort the components of the vector \mathbf{x} in descending order. This allows the algorithm to operate on the largest components first, which simplifies the problem. Then, we find a threshold value τ that ensures

Algorithm 1 (Duchi algorithm – projection onto the unit simplex – see [14])

- 1: **Input:** A vector $\mathbf{x} = [x_1, x_2, \dots, x_C]$
 - 2: Sort the components of \mathbf{x} in descending order: $x_{\text{sorted}} = [x_{(1)}, x_{(2)}, \dots, x_{(C)}]$;
 - 3: Compute the cumulative sum of the sorted components: $S_k = \sum_{i=1}^k x_{(i)}$ for $k = 1, 2, \dots, C$;
 - 4: Find the largest k such that: $x_{(k)} - \frac{S_k - 1}{k} \geq 0$;
 - 5: Set the threshold $\tau = \frac{S_k - 1}{k}$;
 - 6: For each component i , compute the projected value: $y_i = \max(x_i - \tau, 0)$.
 - 7: **Output:** The projected vector $\mathbf{y} = [y_1, y_2, \dots, y_C]$
-

the resulting vector sums up to 1 while maintaining nonnegative components. Finally, after determining the threshold, we subtract it from the components of \mathbf{x} and set any values that become negative to zero.

Gradient update.

For each $X \in [0, 1]^{C \times N}$, the gradient is a matrix of size $C \times N$ given by

$$\nabla f(X) = -4X(S - X^T X), \quad \forall X \in \mathbb{R}^{C \times N}. \quad (3)$$

Updating the gradient using (3) involves:

- matrix multiplication $X^T X$ (size $N \times N$) with $\mathcal{O}(CN^2)$ operations;
- subtraction $S - X^T X$ (size $N \times N$) with $\mathcal{O}(N^2)$ operations;
- matrix multiplication $X(S - X^T X)$ (size $C \times N$) with $\mathcal{O}(CN^2)$ operations;
- scalar multiplication by -4 .

Therefore, each gradient update costs $\mathcal{O}(CN^2)$. Combining this with the cost $\mathcal{O}(NC \log C)$ for computing the projection (N columns, each column involves $C \log C$ operations for the sorting algorithm) we can estimate the *per-iteration cost for the gradient projection method* as

$$\mathcal{O}(CN^2 + NC \log C).$$

This is extremely expensive for the target data with N from 50 to 250 million articles, leading us to the question of *how to implement the gradient projection method efficiently*.

3 Parallelization

The fact that the projection onto Ω is done column-wise suggests that we access columns of the gradient matrix $\nabla f(X)$ at each step. Updating $\nabla f(X)$ using (3) and then extracting its N columns to perform N projections onto the unit simplex is prohibitive for large N .

Let $X \in \mathbb{R}^{C \times N}$ and let $\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{Ci})^\top$ stand for its i -th column. Then gradient of f at X can be written element-wise as

$$\frac{\partial f}{\partial x_{ki}} = \sum_{j=1}^N -4(s_{ij} - \mathbf{x}_i^\top \mathbf{x}_j)x_{kj}, \quad \forall k = 1, 2, \dots, C, \quad i = 1, 2, \dots, N.$$

This means that the element at row k and column i of $\nabla f(X)$ is

$$-4 \times [\text{row } k \text{ of } X] \times [\text{column } i \text{ of } S - \text{column } i \text{ of } X^\top X].$$

Thus, column i of $\nabla f(X)$ is

$$-4X \times [\text{column } i \text{ of } S - \text{column } i \text{ of } X^\top X].$$

Since column i of $X^\top X$ is $X^\top \mathbf{x}_i$, we derive that

$$\text{column } i \text{ of } \nabla f(X) = -4(X\mathbf{s}_i - XX^\top \mathbf{x}_i), \quad (4)$$

where \mathbf{s}_i stands for the column i of the matrix S .

Algorithm 2 (Parallel GPA for fuzzy clustering)

- 1: **Input:** similarity matrix $S \in [0, 1]^{N \times N}$, cluster-num $C > 0$, step size $\tau_n > 0$
 - 2: **Initialize:** membership matrix $X^0 = [\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_N^0] \in \Omega$;
(\mathbf{x}_i^0 is the i -th column of X^0)
 - 3: for each iteration $n := 0, \dots$, max-iter & a stopping criterion is not met:
 - 4: share ^{n} := $X^n(X^n)^\top$;
 - 5: for $i := 1, \dots, N$ (in parallel) do:
 - 6: **grad** _{i} ^{n} := $-4(X^n \mathbf{s}_i - \text{share}^n \mathbf{x}_i^n)$; [gradient update]
 - 7: \mathbf{x}_i^{n+1} := $P_{\Delta C}(\mathbf{x}_i^n - \text{step size} \times \mathbf{grad}_i^n)$; [Duchi algorithm]
 - 8: **Output:** membership matrix $X^{n+1} = [\mathbf{x}_1^{n+1}, \mathbf{x}_2^{n+1}, \dots, \mathbf{x}_N^{n+1}]$
-

Note that S is typically a sparse matrix where each column \mathbf{s}_i of S contains only a few nonzeros, so the computation for $X\mathbf{s}_i$ can be done fast. The cost for XX^\top is $\mathcal{O}(NC^2)$ operations, which is linear w.r.t. to the data size N , as C is much smaller than N . The *overall cost for updating each column of the gradient is $\mathcal{O}(N)$* , which is more manageable than $\mathcal{O}(N^2)$ of updating the whole gradient at once. More importantly, formula (4) tells us that the gradient update can be implemented column-wise in parallel, matching with the parallelization of the projection. Note also that the computation for XX^\top needs to be performed once and is shared among the column updates. Therefore, we propose Algorithm 2, a parallel gradient projection algorithm for solving (\mathcal{P}) .

The following theorem guarantees the non-increasing property of the objective function and the convergence of the iterative matrices generated by Algorithm 2 when the step size is chosen to be sufficiently small.

Theorem 3.1. *There exists $\bar{\tau} > 0$ such that, for any $\tau \in (0, \bar{\tau})$, the sequence $\{X^n\}$ generated by Algorithm 2 from any starting point $X^0 \in \Omega$ with constant step size $\{\tau_n\} \equiv \tau$ has the following properties:*

- (i) *The sequence $\{f(X^n)\}$ is non-increasing. In addition, it holds that*

$$f(X^{n+1}) < f(X^n), \quad \forall n \geq 0,$$

unless X^n is a critical point of (\mathcal{P}) .

- (ii) *The sequence $\{\|X^{n+1} - X^n\|_F\}$ converges to 0; and thus, any accumulation point of $\{X^n\}$ is a critical point of (\mathcal{P}) .*

Proof. Let $\bar{\tau} = 1/L$ with $L := 4\|S\|_2 + 12N$ and $\|S\|_2$ the spectral norm of S (the largest singular value of S). We claim that the operator $X \mapsto \nabla f(X)$ is L -Lipschitz over the bounded set Ω , i.e.,

$$\|\nabla f(X) - \nabla f(Y)\|_F \leq L\|X - Y\|_F, \quad \forall X, Y \in \Omega. \quad (5)$$

The assertions of the theorem then follow from [15, Theorem 9.14] and [12, Theorem 6.1].

To justify (5), let X and Y be arbitrary in Ω . Using the formula (3), the submultiplicative property of the Frobenius norm, and the relation between the Frobenius norm and the spectral norm, we have

$$\begin{aligned} \|\nabla f(X) - \nabla f(Y)\|_F &= 4\|X(S - X^\top X) - Y(S - Y^\top Y)\|_F \\ &\leq 4\|(X - Y)S\|_F + 4\|XX^\top X - YY^\top Y\|_F \\ &\leq 4\|S\|_2\|X - Y\|_F + 4\|XX^\top X - YY^\top Y\|_F. \end{aligned} \quad (6)$$

We estimate the second term on the right-hand side of the last inequality as

$$\begin{aligned} \|XX^\top X - YY^\top Y\|_F &= \|XX^\top X - XX^\top Y + XX^\top Y - YY^\top Y\|_F \\ &\leq \|XX^\top X - XX^\top Y\|_F + \|XX^\top Y - YY^\top Y\|_F \\ &\leq \|XX^\top(X - Y)\|_F + \|(XX^\top - YY^\top)Y\|_F. \end{aligned} \quad (7)$$

Thus, as before, we get

$$\begin{aligned} \|XX^\top(X - Y)\|_F &\leq \|XX^\top\|_F\|X - Y\|_F \\ &\leq \|X\|_F^2\|X - Y\|_F \end{aligned} \quad (8)$$

and

$$\begin{aligned} \|(XX^\top - YY^\top)Y\|_F &\leq \|XX^\top - YY^\top\|_F\|Y\|_F \\ &\leq (\|XX^\top - XY^\top\|_F + \|XY^\top - YY^\top\|_F)\|Y\|_F \\ &\leq (\|X\|_F\|(X - Y)^\top\|_F + \|X - Y\|_F\|Y^\top\|_F)\|Y\|_F \\ &= (\|X\|_F\|Y\|_F + \|Y\|_F^2)\|X - Y\|_F. \end{aligned} \quad (9)$$

It follows from (6)–(9) that

$$\|\nabla f(X) - \nabla f(Y)\|_F \leq 4 (\|S\|_2 + \|X\|_F^2 + \|X\|_F \|Y\|_F + \|Y\|_F^2) \|X - Y\|_F.$$

Thus, by noticing that $\|X\|_F \leq \sqrt{N}$, $\|Y\|_F \leq \sqrt{N}$ as $X, Y \in \Omega$ we get (5). \square

Loss Update. The first assertion of Theorem 3.1 suggests that a sufficient decrease of the loss $f(X)$ can be used as a stopping criterion for Algorithm 2. This requires tracking the loss value at each iteration. Directly computing the loss via the formula $f(X) = \|S - X^\top X\|_F^2$ is expensive, since, as before, it needs to access the dense matrix $X^\top X \in \mathbb{R}^{N \times N}$. To overcome this, we will exploit the shared matrix $XX^\top \in \mathbb{R}^{C \times C}$ and the vectors $X\mathbf{s}_1, X\mathbf{s}_2, \dots, X\mathbf{s}_N$ already computed when updating the gradient in parallel.

Let $\langle \cdot, \cdot \rangle_F$ stand for the *inner product* corresponding to the Frobenius norm, i.e., $\langle X, Y \rangle_F := \text{trace}(X^\top Y)$ for any two same-size real matrices X, Y and $\|X\|_F = \sqrt{\langle X, X \rangle_F}$ for any real matrix X . By the property of the inner product, we have $f(X) = \|S - X^\top X\|_F^2 = \|S\|_F^2 + \|X^\top X\|_F^2 - 2\langle X^\top X, S \rangle_F$. Thus, to update $f(X)$, we need to compute two values $\|X^\top X\|_F^2$ and $\langle X^\top X, S \rangle_F$. Given that $\|X^\top X\|_F^2 = \|XX^\top\|_F^2$ we can compute the first value using XX^\top , which is of size $C \times C$ and hence its Frobenius norm can be obtained directly by definition. Similarly, the second value can be computed by noticing that

Algorithm 3 (Parallel GPA for fuzzy clustering with loss update)

- 1: **Input:** similarity matrix $S \in \mathbb{R}^{N \times N}$, cluster-num $C > 0$, step size $\tau > 0$;
 - 2: **Initialize:** membership matrix $X^0 = [\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_N^0] \in \Omega$;
 $\text{loss}^{-1} := N^2$ (largest value possible for the loss);
 - 3: for each iteration $n := 0, \dots, \text{max-iter}$:
 - 4: $\text{share}^n := X^n (X^n)^\top$;
 - 5: for $i := 1, \dots, N$ (in parallel) do:
 - 6: $\text{prod}_i^n := \langle X^n \mathbf{s}_i, \mathbf{x}_i^n \rangle$; [vector inner product]
 - 7: $\text{merge}^n := \text{prod}_1^n + \text{prod}_2^n + \dots + \text{prod}_N^n$; [scalar addition]
 - 8: $\text{loss}^n := \|S\|_F^2 + \|\text{share}^n\|_F^2 - 2 \times \text{merge}^n$; [loss calculation]
 - 9: if $\text{loss}^{n-1} - \text{loss}^n \leq \text{tol}$:
 - 10: for $i := 1, \dots, N$ (in parallel) do:
 - 11: $\mathbf{x}_i^{n+1} := \mathbf{x}_i^n$;
 - 12: break;
 - 13: else:
 - 14: for $i := 1, \dots, N$ (in parallel) do:
 - 15: $\mathbf{grad}_i^n := -4(X^n \mathbf{s}_i - \text{share}^n \mathbf{x}_i^n)$; [gradient update]
 - 16: $\mathbf{x}_i^{n+1} := P_{\Delta^C}(\mathbf{x}_i^n - \text{step size} \times \mathbf{grad}_i^n)$; [Duchi algorithm]
 - 17: **Output:** membership matrix $X^{n+1} = [\mathbf{x}_1^{n+1}, \mathbf{x}_2^{n+1}, \dots, \mathbf{x}_N^{n+1}]$
-

$\langle X^\top X, S \rangle_F = \text{trace}(X^\top (XS))$. The element i , $i = 1, \dots, N$, on the diagonal

of the matrix $X^\top(XS)$ is

$$[\text{row}(i) \text{ of } X^\top] \times [\text{column}(i) \text{ of } XS] = \langle \mathbf{x}_i, X\mathbf{s}_i \rangle.$$

Thus, the trace of matrix $(X^\top(XS))$ is the sum of $\langle \mathbf{x}_i, X\mathbf{s}_i \rangle$ over all index $i = 1, 2, \dots, N$. Consequently, we have

$$\langle X^\top X, S \rangle_F = \sum_{i=1}^N \langle \mathbf{x}_i, X\mathbf{s}_i \rangle.$$

It follows that $f(X) = \|S\|_F^2 + \|XX^\top\|_F^2 - 2 \sum_{i=1}^N \langle \mathbf{x}_i, X\mathbf{s}_i \rangle$. Updating the loss value at each iteration can now be done in parallel, as shown in Algorithm 3. Compared to the base algorithm, Algorithm 3 makes use of vectors $\mathbf{x}_i, X\mathbf{s}_i$ needed to update the gradient to compute the inner product prod_i in line 6. These inner product values are then merged in line 8 to compute the loss and evaluate it against the tolerance.

Nesterov Accelerated Gradient (FISTA). Regarding the convergence rate of the objective function values $f(X_n)$, the *fast iterative shrinkage-thresholding*

Algorithm 4 (Parallel FISTA for fuzzy clustering)

- 1: **Input:** similarity matrix $S \in \mathbb{R}^{N \times N}$, cluster-num $C > 0$, step size $\tau > 0$;
 - 2: initialize $\bar{X}^0 = [\bar{\mathbf{x}}_1^0, \bar{\mathbf{x}}_2^0, \dots, \bar{\mathbf{x}}_N^0] \in \Omega$, $X^1 := \bar{X}^0$, $\overline{\text{loss}}^0 := f(\bar{X}^0)$;
 - 3: for each iteration $n = 1, \dots$, max-iter do:
 - 4: *//—basic GPA—*
shareⁿ := $X^n(X^n)^\top$;
 - 5: for $i := 1, \dots, N$ (in parallel) do:
 - 6: **grad**_{*i*}ⁿ := $-4(X^n\mathbf{s}_i - \text{share}^n\mathbf{x}_i^n)$;
 - 7: $\bar{\mathbf{x}}_i^n := P_{\Delta^C}(\mathbf{x}_i^n - \text{step size} \times \mathbf{grad}_i^n)$;
 - 8: *//—check loss quality—*
shareⁿ := $\bar{X}^n(\bar{X}^n)^\top$;
 - 9: for $i := 1, \dots, N$ (in parallel) do:
 - 10: prod_{*i*}ⁿ := $\langle \bar{X}^n\mathbf{s}_i, \bar{\mathbf{x}}_i^n \rangle$;
 - 11: mergeⁿ := $\overline{\text{prod}}_1^n + \overline{\text{prod}}_2^n + \dots + \overline{\text{prod}}_N^n$;
 - 12: lossⁿ := $\|S\|_F^2 + \|\text{share}^n\|_F^2 - 2 \times \text{merge}^n$;
 - 13: if $\overline{\text{loss}}^n - \overline{\text{loss}}^{n-1} \leq \text{tol}$:
 - 14: **break**;
 - 15: *//—update inertial parameter and accelerated term—*
 $t_{n+1} := \frac{1 + \sqrt{1 + 4t_n^2}}{2}$;
 - 16: for $i := 1, \dots, N$ (in parallel) do:
 - 17: $\mathbf{x}_i^{n+1} := \bar{\mathbf{x}}_i^n + \left(\frac{t_n - 1}{t_{n+1}}\right) (\bar{\mathbf{x}}_i^n - \bar{\mathbf{x}}_i^{n-1})$;
 - 18: **Output:** membership matrix $\bar{X}^n = [\bar{\mathbf{x}}_1^n, \bar{\mathbf{x}}_2^n, \dots, \bar{\mathbf{x}}_N^n]$
-

algorithm (FISTA, [16]) – a generalized version of the Nesterov accelerated

gradient scheme [17] for constrained convex optimization problems – is well known to speed up the basic gradient projection method. Although there is no theoretical guarantee for nonconvex problems, we would like to examine the performance of the method for fuzzy clustering through numerical experiments.

In FISTA, apart from the basic gradient projection step, we need to use not only the information from the current iteration but also from the previous iteration to define the next one using a special inertial term. This increases the computation cost and memory requirements per iteration. To initialize FISTA, we need points $\bar{X}^0 \in \Omega$, $X^1 := \bar{X}^0$, value $f(\bar{X}^0)$, step size $\tau > 0$, tolerance $\text{tol} > 0$, and set the inertial parameter $t_1 = 1$. Each iteration $n \geq 1$ starts with computing the gradient $\nabla f(X^n)$ and updating the basic GPA

$$\bar{X}^n = P_\Omega [X^n - \tau \nabla f(X^n)]$$

If the loss per iteration is not high enough, i.e., $f(\bar{X}^{n-1}) - f(\bar{X}^n) \geq \text{tol}$, the inertial parameter and the accelerated term are updated with

$$t_{n+1} = \frac{1 + \sqrt{1 + 4t_n^2}}{2}$$

$$X^{n+1} = \bar{X}^n + \left(\frac{t_n - 1}{t_{n+1}} \right) (\bar{X}^n - \bar{X}^{n-1}).$$

A parallel version of FISTA is described in Algorithm 4 where we follow the logic of parallelization of the basic GPA.

4 Refining the set of critical points

Recall that due to the non-convexity of the problem (\mathcal{P}), GPA can guarantee at most a critical point, an $\bar{X} \in \Omega$ satisfying the first-order optimality condition (2). There could be many such points, and thus it is ideal to refine the set of critical points into a smaller set containing local solutions. Aiming at this, we will provide second-order optimality conditions for (\mathcal{P}).

As the constraint set Ω is convex, the first-order optimality condition (2) can be rewritten ([12, Lemma 3.13 and Theorem 3.24]) as

$$\langle \nabla f(\bar{X}), V \rangle_F \geq 0, \quad \forall V \in T(\bar{X}; \Omega) \quad (10)$$

where $T(\bar{X}; \Omega)$ stands for the *Bouligand-Severi tangent cone* to Ω at $\bar{X} \in \Omega$. A direction $V \in \mathbb{R}^{C \times N}$ is called ([12, Definition 3.11]) a *tangent direction* to the set Ω at $\bar{X} \in \Omega$, $V \in T(\bar{X}; \Omega)$, if there exist sequences of points $X^n \in \Omega$ and scalars $\tau_n > 0$, $n = 1, 2, \dots$, such that $\tau_n \downarrow 0$ and

$$V = \lim_{n \rightarrow \infty} \frac{X^n - \bar{X}}{\tau_n}.$$

The concept of tangent direction is fundamental in analyzing perturbations around \bar{X} – a candidate for minimizers of the optimization problem – via points in Ω converging to \bar{X} .

The gradient projection method updates by stepping in the negative gradient direction and projecting onto the constraint set Ω . In (10), if a direction $V \in T(\bar{X}; \Omega)$ satisfies $\langle \nabla f(\bar{X}), V \rangle_F = 0$, the directional derivative of f at \bar{X} along V vanishes, indicating that f does not decrease in that direction. As a result, the method may get stuck at \bar{X} . In such cases, we need to exploit further second-order approximations, the Hessian of the objective function and the second-order tangent cone of the constraint set, to confirm if \bar{X} is a local minimizer.

Let $\bar{X} \in \Omega$ and $\bar{V} \in T(\bar{X}; \Omega)$. One calls ([12, Definition 3.41]) $W \in \mathbb{R}^{C \times N}$ a *second-order tangent direction* to the set Ω at the point \bar{X} in direction \bar{V} if there exist sequences of points $X^n \in \Omega$ and scalars $\tau_n > 0$, $n = 1, 2, \dots$, such that $\tau_n \downarrow 0$ and

$$W = \lim_{n \rightarrow \infty} \frac{X^n - \bar{X} - \tau_n \bar{V}}{\frac{1}{2} \tau_n^2}.$$

The set of second-order tangent directions to Ω at the point \bar{X} in direction \bar{V} is denoted by $T^2(\bar{X}, \bar{V}; \Omega)$.

The next theorem provides us with second-order necessary optimality conditions for (\mathcal{P}) . These conditions pair second-order information of the objective function (Hessian) with first-order information of the constraint set (tangent cone) and vice versa, first-order information of the objective function (gradient) with second-order information of the constraint set (second-order tangent cone).

Theorem 4.1. *Let \bar{X} be a local minimizer of (\mathcal{P}) . In addition, let $\bar{V} \in \mathbb{R}^{C \times N}$ be a direction such that $\langle \nabla f(\bar{X}), \bar{V} \rangle_F = 0$ and $\bar{V} \in T(\bar{X}; \Omega)$. Then*

$$\langle \nabla^2 f(\bar{X}) \bar{V}, \bar{V} \rangle_F \geq 0 \quad (11)$$

and

$$\langle \nabla f(\bar{X}), W \rangle_F \geq 0, \quad \forall W \in T^2(\bar{X}, \bar{V}; \Omega), \quad (12)$$

where the Hessian $\nabla^2 f(\bar{X})$ maps V to

$$\nabla^2 f(\bar{X})V = -4V(S - \bar{X}^T \bar{X}) + 4\bar{X}(V^T \bar{X} + \bar{X}^T V) \in \mathbb{R}^{C \times N} \quad (13)$$

and the second-order tangent cone to Ω at \bar{X} in direction \bar{V} is given by

$$T^2(\bar{X}, \bar{V}; \Omega) = \left\{ W = (w_{ki}) \in \mathbb{R}^{C \times N} \left| \sum_{k=1}^C w_{ki} = 0, \forall i; w_{ki} \geq 0 \text{ if } \begin{cases} \bar{x}_{ki} = 0 \\ \bar{v}_{ki} = 0 \end{cases} \right. \right\}. \quad (14)$$

Proof. Since the objective function is C^2 -smooth and the constraint set is a polyhedral, applying [18, Theorem 3] we get (11) and (12). It remains to show formulas (13) and (14).

Given $\bar{X} \in \mathbb{R}^{C \times N}$, the Hessian $\nabla^2 f(\bar{X})$ is the Fréchet derivative of the gradient operator $X \mapsto \nabla f(X) = -4X(S - X^T X)$ at \bar{X} . First, it is not difficult to verify that the operator

$$V \mapsto \nabla^2 f(\bar{X})V := -4V(S - \bar{X}^T \bar{X}) + 4\bar{X}(V^T \bar{X} + \bar{X}^T V)$$

is a continuous linear operator from $\mathbb{R}^{C \times N}$ to $\mathbb{R}^{C \times N}$. We next justify that

$$\lim_{\|V\|_F \rightarrow 0} \frac{\|\nabla f(\bar{X} + V) - \nabla f(\bar{X}) - \nabla^2 f(\bar{X})V\|_F}{\|V\|_F} = 0 \quad (15)$$

with $\nabla^2 f(\bar{X})V$ given in (13).

From the formula for the gradient, we have

$$\nabla f(\bar{X} + V) = -4(\bar{X} + V)(S - \bar{X}^T \bar{X}) + 4(\bar{X} + V)(V^T \bar{X} + \bar{X}^T V + V^T V).$$

Thus,

$$\begin{aligned} \nabla f(\bar{X} + V) - \nabla f(\bar{X}) &= -4V(S - \bar{X}^T \bar{X}) + 4\bar{X}(V^T \bar{X} + \bar{X}^T V) \\ &\quad + 4V(V^T \bar{X} + \bar{X}^T V) + 4(\bar{X} + V)V^T V. \end{aligned}$$

Now, using (13) we have

$$\nabla f(\bar{X} + V) - \nabla f(\bar{X}) - \nabla^2 f(\bar{X})V = 4V(V^T \bar{X} + \bar{X}^T V) + 4(\bar{X} + V)V^T V.$$

Hence, bounding the norm yields

$$\begin{aligned} \|\nabla f(\bar{X} + V) - \nabla f(\bar{X}) - \nabla^2 f(\bar{X})V\|_F &= \|4V(V^T \bar{X} + \bar{X}^T V) + 4(\bar{X} + V)V^T V\|_F \\ &\leq 4(\|V(V^T \bar{X} + \bar{X}^T V)\|_F + \|(\bar{X} + V)V^T V\|_F) \\ &\leq 4(2\|V\|_F^2 \|\bar{X}\|_F + (\|\bar{X}\|_F + \|V\|_F)\|V\|_F^2). \end{aligned}$$

It follows that

$$\begin{aligned} &\lim_{\|V\|_F \rightarrow 0} \frac{\|\nabla f(\bar{X} + V) - \nabla f(\bar{X}) - \nabla^2 f(\bar{X})V\|_F}{\|V\|_F} \\ &= \lim_{\|V\|_F \rightarrow 0} 4(2\|V\|_F \|\bar{X}\|_F + (\|\bar{X}\|_F + \|V\|_F)\|V\|_F) = 0, \end{aligned}$$

justifying (15).

We now prove (14). By the polyhedral structure,

$$\Omega = \left\{ X = (x_{ki}) \in \mathbb{R}^{C \times N} \left| \sum_{k=1}^C x_{ki} = 1, \forall i, \text{ and } x_{ki} \geq 0, \forall k, i \right. \right\},$$

the tangent cone to Ω at $\bar{X} \in \Omega$ can be given explicitly as

$$T(\bar{X}; \Omega) = \left\{ V = (v_{ki}) \in \mathbb{R}^{C \times N} \left| \sum_{k=1}^C v_{ki} = 0, \forall i, \text{ and } v_{ki} \geq 0 \text{ if } \bar{x}_{ki} = 0 \right. \right\}; \quad (16)$$

see, e.g., [19, formula (3.13)]. Since the latter is also a polyhedral, the second-order tangent cone $T^2(\bar{X}, \bar{V}; \Omega)$ can be computed ([12, Lemma 3.43]) via the (first-order) tangent cone as $T^2(\bar{X}, \bar{V}; \Omega) = T(\bar{V}; T(\bar{X}; \Omega))$. Thus, invoking (16) we obtain (14), which completes the proof. \square

Remark 4.1. As pointed out in [18], conditions (11) and (12) together form a stronger version of the second-order necessary optimality condition stated in [12, Theorem 3.45]: *If $\bar{X} \in \Omega$ is a local minimizer, then for every $\bar{V} \in T(\bar{X}; \Omega)$ with $\langle \nabla f(\bar{X}), \bar{V} \rangle_F = 0$ one has*

$$\langle \nabla^2 f(\bar{X})\bar{V}, \bar{V} \rangle_F + \langle \nabla f(\bar{X}), W \rangle_F \geq 0, \quad \forall W \in T^2(\bar{X}, \bar{V}; \Omega).$$

This, in particular, makes the procedure of refining the set of critical points easier by showing separately that if either (11) or (12) is violated at \bar{X} , then \bar{X} is *not* a local solution of (\mathcal{P}) .

Remark 4.2. Condition (11) is satisfied at $\bar{X} \in \Omega$ for all direction $\bar{V} \in T(\bar{X}; \Omega)$ and $\langle \nabla f(\bar{X}), \bar{V} \rangle_F = 0$ if the optimal value μ_1 of the minimization problem

$$\min_{V \in \mathbb{R}^{C \times N}} \{ \langle \nabla^2 f(\bar{X})V, V \rangle_F \mid V \in T(\bar{X}; \Omega) \ \& \ \langle \nabla f(\bar{X}), V \rangle_F = 0 \} \quad (\mathcal{P}_1)$$

is non-negative. For a fixed $\bar{X} \in \Omega$, the constraint set of (\mathcal{P}_1) is a polyhedron, while the objective function is of quadratic form. Solving nonconvex quadratic programs is NP-hard in general, which requires employing global optimization techniques like branch-and-cut using spatial-branching, as it is available, for example, in solvers such as SCIP³ [20], BARON⁴, Gurobi⁵, or FICO Xpress⁶.

Similarly, condition (12) is fulfilled at a $\bar{X} \in \Omega$ if the optimal value μ_2 of the linear program

$$\min_{W, V \in \mathbb{R}^{C \times N}} \{ \langle \nabla f(\bar{X}), W \rangle_F \mid W \in T^2(\bar{X}, V; \Omega), V \in T(\bar{X}; \Omega), \langle \nabla f(\bar{X}), V \rangle_F = 0 \} \quad (\mathcal{P}_2)$$

is non-negative. Compared to (\mathcal{P}_1) , solvers for (\mathcal{P}_2) are more available even at large scales, for example, additionally to the above, also HiGHS⁷, COPT⁸, and CPLEX⁹. Note also from the formulas of the tangent cones (14) and (16) that

$$T^2(\bar{X}, V; \Omega) = T(\bar{X}; \Omega), \quad \forall V \in T(\bar{X}; \Omega)$$

when $\bar{X} = (x_{ki})$ is a critical point in the interior of Ω , $x_{ki} > 0$ for all k, i . Thus, condition (12) is already satisfied at \bar{X} by the first-order optimality condition (10) and solving (\mathcal{P}_2) is not needed.

We conclude the section with a toy example where GPA identifies multiple critical points, highlighting its strengths and the application of Theorem 4.1 in refining them.

³<https://scipopt.org>

⁴<https://minlp.com>

⁵<https://www.gurobi.com>

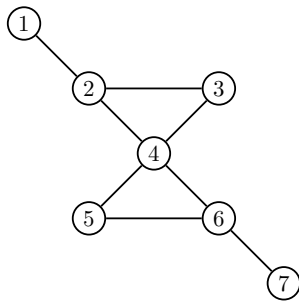
⁶<https://www.fico.com>

⁷<https://highs.dev>

⁸<https://www.copt.de>

⁹<https://www.ibm.com>

Example 4.1. Consider the fuzzy clustering problem (\mathcal{P}) for a 7-node graph, illustrated in Fig. 1. The graph is designed to mimic a small citation network, where each node represents a paper. The nodes are organized into two groups: Group 1, consisting of papers 1, 2, 3, 4, and Group 2, consisting of papers 4, 5, 6, 7. Paper 4 serves as a bridge between the two groups – it cites papers 2 and 3 (from Group 1), and is in turn cited by papers 5 and 6 (from Group 2). The similarity matrix S is constructed from the graph’s adjacency matrix by adding 1s along the diagonal. This reflects the assumption that papers are “similar” if one cites the other, i.e., $s_{ij} = 1$ if paper i cites or is cited by paper j , and each paper is considered “most similar” to itself with $s_{ii} = 1$.



(a) Graph with 7 nodes

$$S = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

(b) Similarity matrix S

Figure 1: A 7-nodes graph and its similarity matrix S

Given that the graph naturally forms two loosely connected clusters with node 4 in common, we set the number of clusters to $C = 2$. We expect GPA to produce a membership matrix $\bar{X} \in \mathbb{R}^{2 \times 7}$ that accurately captures this structure. Algorithm 2 was implemented using Jupyter Notebook on a MacBook with a 1.4 GHz Quad-Core Intel Core i5 processor, a step size of 0.1, and was terminated when the objective function values ceased to decrease. We obtained three different outputs depending on the choice of starting points; see Table 1.

In Scenario 1, where the starting point was initialized randomly, the objective function decreased to a value of 6.49. The resulting membership matrix $\bar{X}_1 = (\bar{x}_{ki})$ closely aligns with the expected structure: papers 2 and 6 strongly define the two clusters ($\bar{x}_{12} = 1, \bar{x}_{22} = 0; \bar{x}_{16} = 0, \bar{x}_{26} = 1$), paper 4 participates in both clusters ($\bar{x}_{14}, \bar{x}_{24} \approx 0.5$), papers 1 and 3 are mainly associated with Cluster 1 ($\bar{x}_{11}, \bar{x}_{13} \approx 1$), and papers 5 and 7 are primarily associated with Cluster 2 ($\bar{x}_{25}, \bar{x}_{27} \approx 1$). We repeated the experiment multiple times using different randomly generated initial points and found the results to be stable. We conclude that GPA successfully recovers the two underlying clusters in this graph instance.

In Scenario 2, the starting point was biased: all seven nodes were assigned to Cluster 1, and none to Cluster 2 (i.e., row 1 was filled with 1s, and row 2 with 0s). The objective function decreased to a value of 8.84. The resulting membership

matrix $\bar{X}_2 = (\bar{x}_{ki})$ exhibits a different structure: paper 4 anchors Cluster 1, grouping together papers 2, 3, 5, and 6, all of which are directly connected via citations. Meanwhile, papers 1 and 7 comprise Cluster 2. Although this solution yields a higher objective function value than in Scenario 1, it remains plausible – effectively grouping strongly connected papers while separating those with weaker connections.

Outputs by GPA for the 7-node graph	Illustration
<p>Scenario 1. The starting point is initialized randomly, the loss is at 6.49 with \bar{X}_1 :</p>	
$\begin{bmatrix} 0.8835 & \mathbf{1.} & 0.9096 & \mathbf{0.5202} & 0.1163 & \mathbf{0.} & 0.0906 \\ 0.1165 & \mathbf{0.} & 0.0904 & \mathbf{0.4798} & 0.8837 & \mathbf{1.} & 0.9094 \end{bmatrix}$	
<p>Scenario 2. The starting point has 1s in row 1 and 0s in row 2; the loss is at 8.84 with \bar{X}_2 :</p>	
$\begin{bmatrix} 0.1308 & 0.6435 & 0.8692 & \mathbf{1.} & 0.8692 & 0.6435 & 0.1308 \\ 0.8692 & 0.3565 & 0.1308 & \mathbf{0.} & 0.1308 & 0.3565 & 0.8692 \end{bmatrix}$	
<p>Scenario 3. The starting point is the matrix with 0.5s in each element; the loss is at 12.25 with \bar{X}_3 :</p>	
$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}$	

Table 1: Outputs by GPA with different starting points for the 7-node graph

In Scenario 3, the starting point was uniformly initialized with 0.5 in all entries, implying that each paper belongs equally to both clusters. The algorithm

terminated after the first iteration, yielding a membership matrix \bar{X}_3 identical to the initial point, with an objective function value of 12.25. Though being a critical point, \bar{X}_3 provides no meaningful insight into the graph’s structure.

It turns out that \bar{X}_3 is *not* a local solution to (\mathcal{P}) , as it violates the second-order necessary optimality condition (11):

$$\langle \nabla^2 f(\bar{X}_3) \bar{V}, \bar{V} \rangle_F = -4 < 0$$

with

$$\bar{V} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in T(\bar{X}_3; \Omega) \text{ satisfying } \langle \nabla f(\bar{X}_3), \bar{V} \rangle_F = 0.$$

The remaining candidates for local solutions are \bar{X}_1 and \bar{X}_2 . Since \bar{X}_2 yields a higher objective function value than \bar{X}_1 , it cannot be a global solution. Numerical results from the Gurobi solver confirm that \bar{X}_1 is a global optimum.

As a side note, the Louvain and Leiden algorithms (implemented using the `igraph` and `leidenalg` libraries in Python) returned two disjoint clusters, $\{1, 2, 3\}$ and $\{4, 5, 6, 7\}$, which capture less structural information from the graph than the GPA solution \bar{X}_1 .

5 Experiments with medium and large datasets

In this section, we examine the performance of the parallel GPA and parallel FISTA algorithms on medium- to large-scale instances. We consider two datasets built from real citation networks:

- a medium-sized instance consisting of approx. 700k articles linked by 4.6 million citations, derived from the Web of Science;
- a large-sized instance comprising about 60 million articles linked by 1.2 billion citations, based on the OpenAlex dataset.

Due to their real and complex graph structures, these instances provide a suitable setting to observe how the parallel GPA and parallel FISTA improve objective function values over iterations. Additionally, we include a synthetic medium-sized instance with two known clusters, enabling us to evaluate the quality of solutions produced by the two algorithms. Details on data preparation are provided below.

WoS citation subgraph. We utilized the Web of Science (WoS) database [21], supported by the German Competence Network for Bibliometrics, and extracted a subgraph of papers in two subjects: *Mathematics* (M) and *Operations Research & Management Science* (OR&MS) after cleaning. Nodes correspond to scholarly works and directed edges to citation relationships. Cleaning excluded non-English papers, those missing key metadata (year, author, WoS unique

identifier, title, journal), papers beyond the range 2000–2024, and invalid references, resulting in 964,811 nodes and 5,087,058 edges. After extracting the largest connected component (796,467 nodes, 4,930,134 edges) and iteratively removing degree-1 nodes, we refined the graph to 722,623 nodes and 4,856,290 edges. For our future embedding tasks, we removed papers lacking abstracts and reprocessed the graph, yielding a final version with 698,135 nodes and 4,590,190 edges.

OpenAlex connected citation graph. The OpenAlex connected citation graph represents the largest connected component of the OpenAlex citation dataset [22] after cleaning and preprocessing. The dataset, based on the OpenAlex snapshot (July 31, 2024), was cleaned by excluding works without journal, authors, or title, with years outside 1901–2024, non-English language, non-article/chapter types, or marked as retracted. Invalid references were also removed. After building the graph and extracting its largest connected component using a BFS-like algorithm, the final graph after degree-1 removal contains 59,343,462 nodes, 1,176,978,458 edges.

Artificial graph. An artificial graph was constructed to mirror the scale of the WoS citation subgraph using a two-cluster Erdős–Rényi model. Two random graphs were generated separately: Cluster 1 (500,000 nodes, edge probability $p_{c1} = 1200/499999$) and Cluster 2 (250,000 nodes, edge probability $p_{c2} = 1600/249999$), with Cluster 2 nodes re-indexed to avoid ID overlaps before merging. A fixed number 200 of inter-cluster edges was then added by randomly connecting node pairs between the two clusters to control the level of inter-connectivity. After merging, nodes with degree smaller than 1 were iteratively pruned, and the final graph was verified to form a single connected component.

Hardware. All of the experiments in this section were conducted on a high-performance computing system equipped with two Intel® Xeon® Gold 6132 CPUs, providing a total of 28 cores and 56 threads at 2.60 GHz base clock speed (3.7 GHz max). The system includes 376 GiB of RAM and four NVIDIA Tesla V100-SXM2-16GB GPUs with 5.120 CUDA cores and 16GB HBM2 memory. The setup ran on CUDA 12.2 with NVIDIA driver 535.183.01.

Hyperparameters. For every graph, the number of clusters C was set to 2 and the similarity matrix S was formed by adding ones on the diagonal of the adjacency matrix, as done with the 7-node tested graph in Example 4.1.

To get the algorithms running, it remains to initialize a membership matrix $X^0 \in \Omega$ and to choose a step size τ . Theorem 3.1 states that the objective function values are non-increasing over iterations of the parallel GPA if the constant step size τ is small enough, regardless of the choice of $X^0 \in \Omega$. Thus, unless otherwise stated, we use a randomly generated $X^0 \in \Omega$. Shown in the proof of the theorem, a viable step size is

$$\tau \leq \bar{\tau} = \frac{1}{4\|S\|_2 + 12N}$$

with $\|S\|_2$ the spectral norm of S . As computing the spectral norm of S can be expensive with large datasets, we replace $\|S\|_2$ with $\|S\|_F$. The latter is easy to compute in our situation, by counting the number of non-zeros in S , as S is obtained from the adjacency matrix of the graph with 1s on the diagonal. A sufficiently small step size for the medium-size instance is 10^{-8} , and for the large-size instance 10^{-10} . We stopped the computation when there was no further decrease in the loss.

Experiment 1: How parallel GPA and FISTA can reduce the loss?

In this experiment, we evaluate the effectiveness of parallel GPA and FISTA

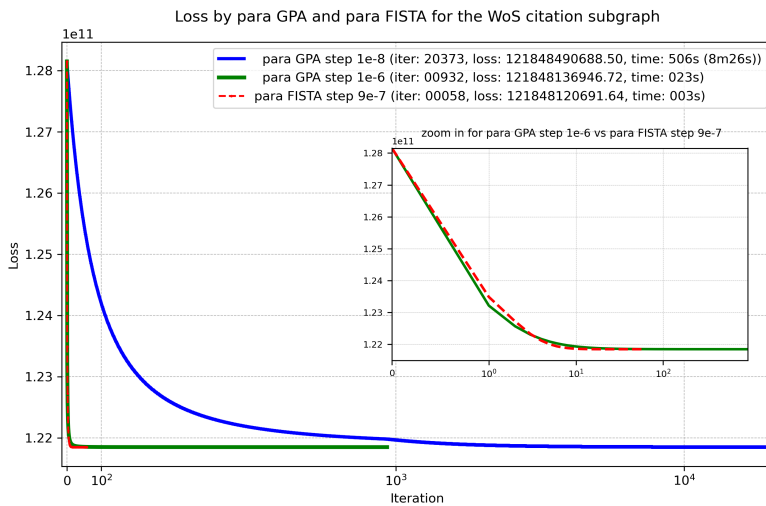


Figure 2: Loss over iterations by parallel GPA and FISTA for the WoS citation subgraph with different step sizes

in reducing the objective loss.

For the WoS medium-sized instance, we first test the theoretically viable step size 10^{-8} . As expected from Theorem 3.1, parallel GPA with this step size results in a monotonically decreasing loss over 20,000 iterations, taking about 8.5 minutes. However, empirical tuning reveals that the algorithm can tolerate much larger step sizes. For example, with a step size of 10^{-6} , a factor of 100 larger, parallel GPA converges to a comparable (and actually lower) loss in fewer than 1,000 iterations and just 23 seconds.

This shows that while the theoretical step size ensures convergence, practical performance can be significantly improved by using a well-chosen, larger step size. Nevertheless, even with the tuned step size, GPA may still require many iterations with diminishing returns in loss reduction. This motivates us to consider parallel FISTA as a heuristic acceleration technique – despite its lack of

convergence guarantees in our non-convex setting. With a slightly smaller step size of 9×10^{-7} , parallel FISTA reaches the best loss achieved by parallel GPA in only 58 iterations and 3 seconds – demonstrating a substantial speedup. A detailed comparison of the performance of parallel GPA and FISTA is shown in Fig. 2.

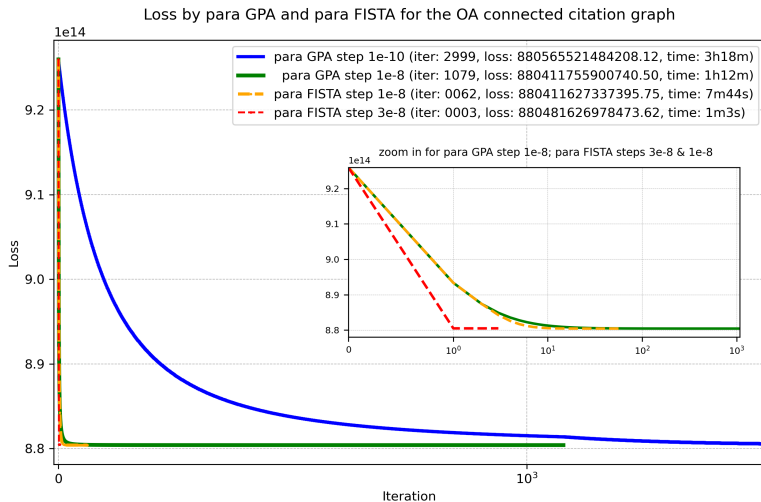


Figure 3: Loss over iterations by parallel GPA and FISTA for the OpenAlex connected citation graph with different step sizes

We now evaluate the performance of parallel GPA and FISTA on the large-scale OpenAlex connected citation graph. As with the WoS instance, we begin by applying a conservative step size of 10^{-10} , which is well below the theoretical threshold established in Theorem 3.1. As guaranteed by the theorem, this ensures that parallel GPA achieves a monotonically decreasing loss over 3,000 iterations – though at the cost of more than 3.3 hours of runtime. In contrast, parallel FISTA with a larger step size of 3×10^{-8} achieves a significantly better loss value with just 3 iterations and 1 minute.

With the step size 10^{-8} , both methods yield further reductions in loss within reasonable iterations and times: parallel GPA with 1,079 iterations and more than an hour, and parallel FISTA with 62 iterations and less than 8 minutes. Notably, the loss initially achieved by parallel GPA with step size 10^{-10} can now be matched by parallel GPA and FISTA after only 30 and 10 iterations, respectively. A visual comparison of the loss trajectories across these configurations is provided in Fig. 3.

The experiments on WoS and OpenAlex instances demonstrate that parallel GPA and parallel FISTA are effective in reducing the objective loss, especially

when step sizes are appropriately tuned. Interestingly, parallel FISTA can serve as a powerful heuristic acceleration method, particularly on large-scale datasets – even without theoretical guarantees in the non-convex setting.

Experiment 2: Do the solutions look reasonable?

Assessing whether a clustering solution “looks reasonable” is challenging for large graphs, as they cannot be easily visualized. Example 4.1 also demonstrated how GPA can yield a variety of outcomes depending on initialization, reflecting the non-convex nature of the objective and the sensitivity of the method to the starting point. To better assess the quality of the clustering solution, we focus here on the artificial graph, which has a comparable size to the mid-sized WoS graph but has two known ground-truth clusters (labeled 1 for Node IDs 1–500000 and labeled 2 for the others).

With the same setting, step size 5×10^{-7} and X^0 with ones in the second row, both parallel GPA and parallel FISTA successfully recovered the expected structure in less than 10 iterations and about 15 seconds. This demonstrates the potential of the two algorithms when the underlying graph is well-formed. Fig. 4 visualizes the clustering result (the output membership matrix \bar{X}) by

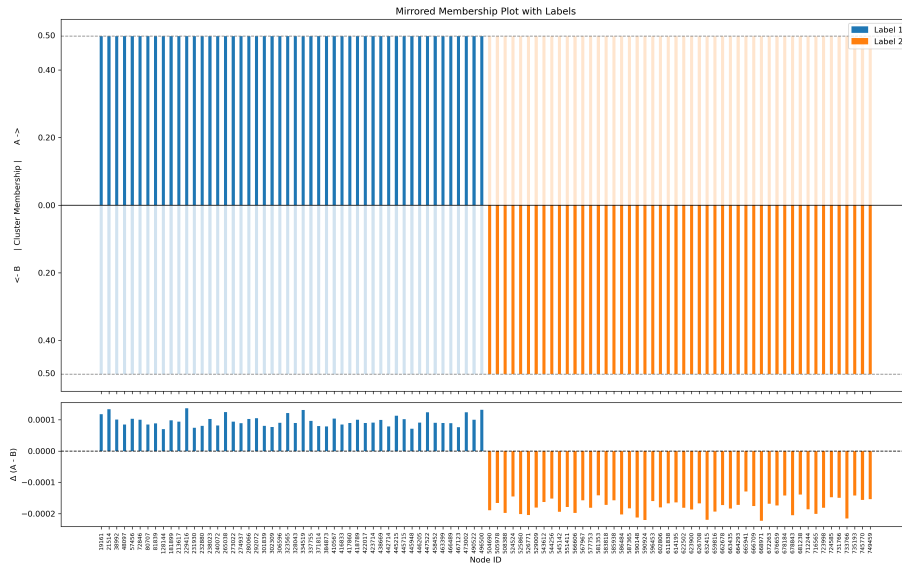


Figure 4: Clustering solution by parallel GPA on the artificial graph.

parallel GPA for the artificial graph. Due to the graph’s size, we randomly selected 50 nodes from each of the two labeled groups for plotting. The top plot shows the membership values of each selected node with respect to Clusters A and B, where nondominant values are rendered in a fainter color. The bottom plot displays the differences $\Delta (A - B)$ in membership value between Cluster A and Cluster B. These plots indicate strong agreement between the dominant memberships and the ground-truth labels.

While results on WoS and OpenAlex graphs are less conclusive – likely due to the complex graph topology or sensitivity to initial points – parallel GPA and FISTA remain promising for large-scale fuzzy clustering.

6 Conclusion and future work

We introduced parallel GPA and parallel FISTA, two algorithms for fuzzy clustering tailored to large-scale scientific article datasets. Both methods achieve substantial reductions in the clustering objective, with parallel FISTA notably delivering strong heuristic acceleration on real citation graphs containing hundreds of thousands to millions of nodes. For parallel GPA, we prove convergence to critical points and establish second-order optimality conditions that offer novel theoretical insights into solution quality. On a synthetic graph with a well-defined cluster structure, two methods successfully recover the expected clusters, confirming their effectiveness under ideal conditions.

A key innovation of our approach lies in the parallelization strategy: By taking advantage of the mathematical structure of the problem, projections are calculated independently on each column, while gradient updates are based on a small shared matrix XX^T among columns, avoiding explicit computation of the large $X^T X$ matrix. This small matrix is also reused to evaluate the loss function $f(X) = \|S - X^T X\|_F^2$, significantly reducing computational cost. These algorithmic improvements are implemented with CUDA on GPUs, exploiting their massive parallelism and high memory bandwidth to scale efficiently to very large datasets.

Future work will focus on improving initialization strategies, adaptive step size tuning, and exploiting hard clustering solutions to enhance robustness and interpretability. We also plan to use weighted edges, incorporate structured sparsity, metadata, and text-based similarity derived from titles and abstracts using large language models, to better capture complex real-world graphs. Finally, we plan to extensively compare the results with existing clusterings.

Acknowledgements. This work is co-funded by the European Union (European Regional Development Fund EFRE, Fund No. STIIV-001) and supported by the German Competence Network for Bibliometrics (Grant No. 16WIK2101A). The research for this article was conducted at the Research Campus MODAL, funded by the German Federal Ministry of Education and Research (BMBF) (Grant No. 05M14ZAM, 05M20ZBM, 05M2025).

References

- [1] S. Fortunato, C. T. Bergstrom, K. Börner, et al., Science of science, *Science* **359**, eaao0185, 2018.
- [2] J. Beel, B. Gipp, S. Langer, and C. Breitingner, Research-paper recommender systems: a literature survey, *Int. J. Digit. Libr.* **17**, 305–338, 2016.

- [3] R. Klavans and K. W. Boyack, Research portfolio analysis and topic prominence, *J. Informetrics* **10**, 441–457, 2016.
- [4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, Fast unfolding of communities in large networks, *J. Stat. Mech.* **2008**, P10008, 2008.
- [5] V. A. Traag, L. Waltman, and N. J. van Eck, From Louvain to Leiden: guaranteeing well-connected communities, *Sci. Rep.* **9**, 5233, 2019.
- [6] Q. Xie and L. Waltman, Citation-based clustering of publications using CitNetExplorer and VOSviewer, *Scientometrics* **111**, 1053–1070, 2017.
- [7] L. Waltman, K. W. Boyack, G. Colavizza, and N. J. van Eck, A principled methodology for comparing relatedness measures for clustering publications, *Quant. Sci. Stud.* **1**, 691–713, 2020.
- [8] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Springer, 1981.
- [9] D. L. Pham, C. Xu, and J. L. Prince, Current methods in medical image segmentation, *Annu. Rev. Biomed. Eng.* **2**, 315–337, 2000.
- [10] N. R. Pal and J. C. Bezdek, On cluster validity for the fuzzy c-means model, *IEEE Trans. Fuzzy Syst.* **3**, 370–379, 1995.
- [11] T. Nepusz, A. Petróczy, L. Négyessy, and F. Bazsó, Fuzzy communities and the concept of bridgeness in complex networks, *Phys. Rev. E* **77**, 016107, 2008.
- [12] A. Ruszczyński, *Nonlinear optimization*, Princeton Univ. Press, Princeton, NJ, 2006.
- [13] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, New York, NY, 2010.
- [14] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, Efficient projections onto the ℓ_1 -ball for learning in high dimensions, in *Proc. 25th Int. Conf. Mach. Learn.*, 2008.
- [15] A. Beck, *Introduction to nonlinear optimization: theory, algorithms, and applications with Python and MATLAB*, 2014.
- [16] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.* **2**, 183–202, 2009.
- [17] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, *Soviet Math. Dokl.* **27**, 372–376, 1983.
- [18] D. T. V. An and N. D. Yen, Optimality conditions based on the Fréchet second-order subdifferential, *J. Glob. Optim.* **81**, 351–365, 2021.

- [19] L. Ban, B. S. Mordukhovich, and W. Song, Lipschitzian stability of parametric variational inequalities over generalized polyhedra in Banach spaces, *Nonlinear Anal.* **74**, 441–461, 2011.
- [20] K. Bestuzheva, M. Besançon, W-K. Chen, A. Chmiela, T. Donkiewicz, J. van Doornmalen, L. Eifler, O. Gaul, G. Gamrath, A. Gleixner, L. Gottwald, C. Graczyk, K. Halbig, A. Hoen, C. Hojny, R. van der Hulst, T. Koch, M. Lübbecke, S.J. Maher, F. Matter, E. Mühmer, B. Müller, M.E. Pfetsch, D. Rehfeldt, S. Schlein, F. Schlösser, F. Serrano, Y. Shinano, B. Sofranac, M. Turner, S. Vigerske, F. Wegscheider, P. Wellner, D. Weninger, and J. Witzig, Enabling Research through the SCIP Optimization Suite 8.0, *ACM Trans. Math. Softw.* **49**, 1–21, 2022.
- [21] Clarivate Analytics, Web of Science, 2024.
- [22] J. Priem, H. Piwowar, and R. Orr, OpenAlex: a fully-open index of scholarly works, authors, venues, institutions, and concepts, arXiv preprint arXiv:2205.01833, 2022.