Learning doubly stochastic and nearly idempotent affinity matrix for graph-based clustering

Julien Ah-Pine

▶ **To cite this version:**

Julien Ah-Pine. Learning doubly stochastic and nearly idempotent affinity matrix for graph-based clustering. European Journal of Operational Research, 2021, 10.1016/j.ejor.2021.12.034 . hal-03539972

HAL Id: hal-03539972
https://hal.science/hal-03539972

Submitted on 28 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Graphical Abstract

Learning Doubly Stochastic and Nearly Idempotent Affinity Matrix for Graph-Based Clustering

Julien Ah-Pine

Highlights

Learning Doubly Stochastic and Nearly Idempotent Affinity Matrix for Graph-Based Clustering

Julien Ah-Pine

- Using doubly stochastic and idempotent matrices for representing partitions in graph-based clustering.
- New continuous relaxation of the NP-hard graph partitioning problem.
- Clustering and Laplacian matrices as complementary orthogonal projection matrices.
- Optimization by Alternating Direction Method of Multipliers and Projections On Convex Sets.

Learning Doubly Stochastic and Nearly Idempotent Affinity Matrix for Graph-Based Clustering

Julien Ah-Pine[a,b,c]

[a] *University of Lyon, Lyon 2 and ERIC EA 3083, 5 Avenue Pierre Mendès France, 69500, Bron, France*
[b] *University of Clermont Auvergne, LMBP UMR 6620, 3 place Vasarely, 63170, Aubière, France*
[c] *Université Clermont Auvergne, CNRS, IRD, CERDI, F-63000, Clermont-Ferrand, France*

Abstract

In graph-based clustering, a relevant affinity matrix is crucial for good results. Double stochasticity of the affinity matrix has been shown to be an important condition, both in theory and in practice. In this paper, we emphasize idempotency as another key condition. In fact, a theorem from Sinkhorn, R. (1968) allows us to exhibit the bijective relationship between the set of doubly stochastic and idempotent matrices of order n (modulo permutation of rows and columns) on the one hand, and the set of possible partitions of a set of n objects on the other hand. Thereby, both properties are necessary and sufficient conditions for properly modeling the clustering or graph partitioning tasks using matrices. Yet, this leads to a NP-hard discrete optimization problem. In this context, our main contribution is the introduction of a new relaxed model that efficiently learns a double stochastic and nearly idempotent affinity matrix for graph-based clustering. Our approach leverages existing properties between doubly stochastic and idempotent matrices on the one hand, and their associated Laplacian matrices on the other hand. The resulting optimization problem is bi-convex and can be addressed by an Alternating Direction Method of Multipliers scheme. Furthermore, our model requires less parameters to set in contrast to most of recent works. The experimental results we obtained using several real-world benchmarks, exhibit the interest of our method and the importance of taking into account idempotency in graph-based clustering.

Keywords: Machine learning, Graph-based clustering, Doubly stochastic

Preprint submitted to EJOR

affinity matrix, Idempotent matrix, ADMM.

1. Introduction

Given a set of objects, clustering aims to automatically find groups of similar elements. The discovered partition allows one to have a synthesized viewpoint of the different profiles among the objects, to gain knowledge from these patterns and to take better decisions. Clustering is a core topic investigated in the data analysis and machine learning communities and related textbooks are, for example, [20, 15, 17]. Clustering can also be apprehended from a graph point of view. In this context, the goal is to separate the set of nodes into several groups such that within each subset the nodes are densely connected. In this case, the clustering task is related to the following problems: graph or clique partitioning (see for example [8]) and graph max-cut or min-cut (see for example [13]). These problems have been studied by scholars from the discrete mathematics and operational research communities as well (see for example [14, 5]).

From a discrete point of view, the brute force approach for solving the clustering problem is to enumerate all possible partitions, measure their respective quality according to a given criterion and keep the best one(s). But the number of possible partitions of a set of n objects is given by the Bell number which grows exponentially with respect to n. For instance if $n = 10$ the number of possible cases is 115,975 while if $n = 100$ the number of solutions increases to $\approx 4 \times 10^{115}$. Therefore, this naïve approach is not possible in practice from a general perspective.
The clustering problem can also be expressed as a binary integer linear program [23, 18]. However the number of constraints is of order $O(n^3)$ and thus, the application of this exact procedure is not scalable either.

The complexity of the clustering problem depends on the criterion used to assess the quality of the partitions [19]. In this paper, we focus on the conventional Sum of Squared Errors (SSE) criterion (that we formally introduce shortly after). In this instance, the clustering problem is NP-hard (see [2] and references therein). As a consequence, when minimizing SSE, heuristics have been developed in order to cope with the combinatorial nature of the task. Moreover, the research activities in this domain have been stimulated by the wide range of applications of cluster analysis. Indeed, clustering algorithms are employed in many applied domains such as for image segmentation in signal processing, for gene functional analysis in bioinformatics, for

community detection in social network analysis or for pattern discovery in computational social science, ... (see for example [32] and references therein). Regarding fields closer to operational research applications, cluster analysis is used in marketing for customer segmentation and relationship management (see for example [26] and references therein). Another example concerns the design of manufacturing systems, where cluster analysis has been applied for group technology and cell formation (see for example [9] and references therein). Yet another application is in VLSI (Very Large Scale Integration) design (see for example [1, 14]), where clustering methods can help optimize efficiency.

There are different clustering heuristics and one way to categorize them is by the type of input they deal with.
The well-known k-means algorithm is a feature-based approach. It takes as input a feature matrix which represents the set of objects as vectors in an Euclidean space. Then, it seeks a partition with k clusters that minimizes the SSE criterion.
In contrast, graph-based clustering takes as input a pairwise affinity matrix. It is the weighted adjacency matrix of an undirected graph whose vertices are the objects and the edges weights are the similarity (or affinity) values between the connected pairs of objects. In this context, the clustering problem becomes a graph partitioning one and the SSE objective function employed in the k-means algorithm has strong relationships with the normalized cut criterion used in balanced graph cuts [10, 35].

In this paper, we are interested in graph-based clustering and we particularly focus on the *spectral clustering* framework which is a modern and elegant cluster analysis methodology which outperforms conventional clustering techniques very often, as outlined in [35, 24]. This approach borrows concepts and tools from spectral graph theory to solve graph partitioning tasks approximately as exposed in [35, 24].
Spectral clustering proceeds in three steps: (ι) determine a suitable affinity matrix of the objects; ($\iota\iota$) compute the spectral decomposition of the Laplacian matrix and; ($\iota\iota\iota$) apply a feature-based clustering technique to a limited number of leading eigenvectors. Step ($\iota\iota$) can be interpreted as a spectral embedding of the vertices into a low-dimensional Euclidean space. In step ($\iota\iota\iota$), the regular k-means algorithm is usually applied. Our work is mainly concerned with step (ι) which is about learning a significant affinity matrix

for clustering purposes. This step is crucial in order to obtain good results[1].

It is interesting to interpret the spectral clustering algorithm from an optimization standpoint. In fact, the three steps (ι), ($\iota\iota$), ($\iota\iota\iota$), form a procedure that is similar in spirit to the following common strategy used to approximately solve a NP-complete discrete optimization problem: (i) define a continuous relaxation of the initial optimization problem; (ii) solve the relaxed problem in polynomial time; (iii) apply a rounding procedure in order to obtain a feasible solution to the initial problem. Accordingly, from an optimization perspective, our work is related to step (i) and aims to design a new continuous relaxation model of the clustering problem.

When attempting to bridge the gap between the affinity matrix in (ι) and the partition one ultimately targets in ($\iota\iota\iota$), a first important property arises [39]: *double stochasticity*. A square matrix $\mathbf{X} = (x_{ii'})$ in $\mathbb{R}^{n\times n}$, is said to be doubly stochastic if $\mathbf{X} \geq \mathbf{0}_n$ and $\mathbf{X}\mathbf{e}_n = \mathbf{X}^\top \mathbf{e}_n = \mathbf{e}_n$, where $\mathbf{0}_n$ is the null square matrix of order n, $\mathbf{X} \geq \mathbf{0}_n$ is a shortcut for $x_{ii'} \geq 0, \forall i, i' = 1, \ldots, n$, $\mathbf{e}_n$ is the n dimensional vector with 1 in all entries and $\mathbf{X}^\top$ is the transpose of $\mathbf{X}$.
In fact, the graph-based formulation of SSE shows that the solution to the clustering problem that minimizes this criterion has to be doubly stochastic. This observation have conducted several researchers to define in step (ι), unsupervised learning algorithms that output a doubly stochastic matrix from a given affinity matrix [40, 36]. They showed that learning a doubly stochastic affinity matrix indeed improves the spectral clustering results. Some recent works in this area, such as [37, 28], have exploited additional properties in order to make the affinity matrix learnt in (ι) closer and closer to a matrix which properly encodes a partition that is expected at the final stage ($\iota\iota\iota$). Nonetheless, these approaches lead to complex models with numerous parameters to tune.

In this contribution, we introduce a new framework for learning a doubly stochastic affinity matrix. A core characteristic of our model is that the affinity matrix we search for, should be *idempotent* "as much as possible" in addition to be doubly stochastic. Our framework can be briefly summarized as follows. After having exhibited the relationships between SSE and the Frobenius distance between the given affinity matrix and our variable, we propose to apply this latter criterion as an objective function of our opti-

[1]Not only to spectral clustering but to all graph-based clustering techniques as well.

mization problem. Then, by exploiting the fact that both a doubly stochastic and idempotent affinity matrix and its related Laplacian matrix are orthogonal projection matrices, we design a new bi-convex optimization problem that can be addressed by an Alternating Direction Method of Multipliers (ADMM) scheme.

The rest of the paper is organized as follows. In section 2, we recall important concepts and in particular, we stress a result from Sinkhorn [33] which is central to our work. In section 3, we present our framework which takes a distinct path from previous works. Our approach aims to learn a doubly stochastic and *nearly idempotent* affinity matrix. In the goal of demonstrating the interest of our approach we conducted experiments using nine real-world clustering benchmarks. The results we obtained are depicted in section 4 and show the superior performances of our method over two existing doubly stochastic affinity matrix learning techniques. Finally in section 5, we give a summary of our contributions and provide some future research lines as concluding remarks.

2. Background and motivations

2.1. Doubly stochastic and idempotent matrices as clustering matrices

As stated in the introduction, double stochasticity is a property that has been addressed in several papers for clustering purposes. *Idempotency* is another property that we promote in this paper. $\mathbf{X} \in \mathbb{R}^{n \times n}$ is said to be idempotent if $\mathbf{X}^2 = \mathbf{X}$. Let $\mathbf{J}_n$ and $\mathbf{I}_n$ be two special square matrices of order n. The former one has the value $1/n$ in all entries and the latter one is the identity matrix.

For example, in the case $n = 4$, one has:

$$\mathbf{J}_4 = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \text{ and } \mathbf{I}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Note that $\mathbf{J}_n$ and $\mathbf{I}_n$ are idempotent. The following result from Sinkhorn [33] is at the core of our developments.

Theorem 1 ([33])**.** *$\mathbf{X} \in \mathbb{R}^{n \times n}$ is doubly stochastic and idempotent if and only if there are k positive numbers $n_1, n_2, \ldots, n_k$ with $n_1 + n_2 + \ldots + n_k = n$,*

and a permutation matrix $\mathbf{P}$ *such that:*

$$\mathbf{X} = \mathbf{P} \begin{pmatrix} \mathbf{J}_{n_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{n_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_{n_k} \end{pmatrix} \mathbf{P}^\top \tag{1}$$

Accordingly, any doubly stochastic and idempotent matrix $\mathbf{X}$ of order n (modulo permutation of rows and columns) can be associated with a partition of the integer n and *vice-versa*. In the context of graph partitioning, if $\mathbf{X}$ is the weighted adjacency matrix of the affinity relationships among a set of n objects, then each one of the k blocks $\mathbf{J}_{n_1}, \dots, \mathbf{J}_{n_k}$ in Theorem 1, corresponds to a complete sub-graph (or a clique or an equivalence class). Clearly, these blocks represent subsets that are mutually disjoint and, all together, they form a partition of the set of objects in k clusters.

Note that, given any square matrix $\mathbf{A}$ and permutation matrix $\mathbf{P}$, $\mathbf{PAP}^\top$ rearranges the rows and columns of $\mathbf{A}$ in exactly the same manner. Consequently, $\mathbf{PAP}^\top$ is symmetric if and only if $\mathbf{A}$ is symmetric. Applying this reasoning to (1), we can state the following necessary condition.

Corollary 1. *If* $\mathbf{X} \in \mathbb{R}^{n \times n}$ *is doubly stochastic and idempotent then* $\mathbf{X}$ *is symmetric.*

Furthermore, (1) indicates that $\mathrm{Tr}(\mathbf{X}) = \mathrm{Tr}(\mathbf{J}_{n_1}) + \ldots + \mathrm{Tr}(\mathbf{J}_{n_k})$ because for any square matrix $\mathbf{A}$ and permutation matrix $\mathbf{P}$, $\mathrm{Tr}(\mathbf{PAP}^\top) = \mathrm{Tr}(\mathbf{P}^\top\mathbf{PA}) = \mathrm{Tr}(\mathbf{A})$. Moreover, it is clear that $\mathrm{Tr}(\mathbf{J}_{n_j}) = 1, \forall j = 1, \dots, k$. Hence, we have the following property.

Corollary 2. *If* $\mathbf{X} \in \mathbb{R}^{n \times n}$ *is doubly stochastic and idempotent then* $\mathrm{Tr}(\mathbf{X})$ *equals* k *the number of positive numbers that sum to* n *in the relationship* $n_1 + n_2 + \ldots + n_k = n$ *in Theorem 1.*

In other words, if $\mathbf{X}$ is doubly stochastic and idempotent then $\mathrm{Tr}(\mathbf{X})$ equals the number of clusters of the associated partition of its rows (and columns). Given these properties, any $\mathbf{X} \in \mathbb{R}^{n \times n}$ which is doubly stochastic and idempotent is called a *clustering matrix* hereafter.

For illustration purposes, we give the following example: the partition $\{\{a, b, d\}, \{c\}\}$ of the set $\{a, b, c, d\}$ in the two clusters $\{a, b, d\}$ and $\{c\}$, is

an instance of the integer partition[2] $3+1=4$ and can be represented by:

$$\mathbf{X} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \end{pmatrix} \end{array}$$

This matrix is clearly doubly stochastic and idempotent. Moreover, if one permutes the rows and columns of c and d using the permutation matrix $\mathbf{P}$ below, it becomes block diagonal with each block $\mathbf{J}_3$ and $\mathbf{J}_1$ representing a cluster:

$$\underbrace{\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}}_{\mathbf{P}} \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \\ 1/3 & 1/3 & 0 & 1/3 \end{pmatrix} \end{array} \underbrace{\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}}_{\mathbf{P}^\top} =$$

$$\begin{array}{c} \\ a \\ b \\ d \\ c \end{array} \begin{array}{c} \begin{array}{cccc} a & b & d & c \end{array} \\ \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Besides, in this example, $\mathrm{Tr}(\mathbf{X}) = 2$, the number of clusters.

Note that $\mathbf{J}_4$ and $\mathbf{I}_4$ given as examples at the beginning of this paragraph, can be interpreted as two clustering matrices of the set $\{a, b, c, d\}$ respectively representing the two opposite partitions $\{\{a, b, c, d\}\}$[3] and $\{\{a\}, \{b\}, \{c\}, \{d\}\}$[4].

[2]This partition corresponds to the case $n = 4$, $k = 2$, $n_1 = 3$, $n_2 = 1$, and $n_1 + n_2 = n$ in Theorem 1 and Corollary 2.

[3]This partition corresponds to the case $n = 4$, $k = 1$, $n_1 = 4$ and $n_1 = n$ in Theorem 1 and Corollary 2.

[4]This partition corresponds to the case $n = 4$, $k = 4$, $n_1 = n_2 = n_3 = n_4 = 1$ and $n_1 + n_2 + n_3 + n_4 = n$ in Theorem 1 and Corollary 2.

2.2. Graph-based formulation of SSE and the Frobenius distance criterion

Doubly stochastic and idempotent matrices naturally arise when one recasts the SSE criterion using inner-product matrices. Let $\{\mathbf{x}_i\}_{i=1,\dots,n}$ be n vectors in an Euclidean space, representing the objects to cluster. Suppose there is a mapping ϕ from the latter space to a higher-dimensional Reproducing Kernel Hilbert Space (RKHS). In the general kernel k-means framework [10], one seeks a partition $C = \{C_1, \dots, C_k\}$ that minimizes:

$$\mathrm{SSE}(C) = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} \|\phi(\mathbf{x}_i) - \mathbf{c}_j\|^2 \tag{2}$$

where $\|.\|$ is the distance in the RKHS and $\mathbf{c}_j$ is the mean vector of elements in C_j of cardinal n_j in the RKHS: $\mathbf{c}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in C_j} \phi(\mathbf{x}_i)$.

In the conventional general k-means method, one represents a partition $C = \{C_1, \dots, C_k\}$ by a $n \times k$ binary *assignment matrix* $\mathbf{Y} = (y_{ij})$ of general term $y_{ij} = 1$ if $\mathbf{x}_i \in C_j$ and $y_{ij} = 0$ otherwise. Using $\mathbf{Y}$, one has $n_j = \sum_{i=1}^{n} y_{ij}$ and $\mathbf{c}_j = \frac{1}{\sum_{i'=1}^{n} y_{i'j}} \sum_{i''=1}^{n} \phi(\mathbf{x}_{i''}) y_{i''j}$.The clustering model based on the minimization of SSE can then be expressed as:

$$\min_{\mathbf{Y} \in \{0,1\}^{n \times k}} \mathrm{SSE}(\mathbf{Y}) = \sum_{j=1}^{k} \sum_{i=1}^{n} y_{ij} \|\phi(\mathbf{x}_i) - \frac{1}{\sum_{i'=1}^{n} y_{i'j}} \sum_{i''=1}^{n} \phi(\mathbf{x}_{i''}) y_{i''j}\|^2 \tag{3}$$

$$\text{s.t. } \mathbf{Y}\mathbf{e}_k = \mathbf{e}_n, \mathbf{Y}^\top \mathbf{e}_n \geq \mathbf{e}_k.$$

where $\mathbf{Y}\mathbf{e}_k = \mathbf{e}_n$ states that each vector should be assigned to exactly one cluster and $\mathbf{Y}^\top \mathbf{e}_n \geq \mathbf{e}_k$ indicates that each cluster should contain at least one vector.

The SSE criterion is non-linear and non-convex in (y_{ij}). In addition, the discrete nature of Problem (3) makes it NP-hard.

Problem (3) can be formulated from a graph viewpoint. Furthermore, the kernel trick allows the k-means algorithm to be extended to kernel functions (see [21] for example). For any mapping ϕ, one can define for all $i, i' = 1, \dots, n$, the kernel function $\kappa(\mathbf{x}_i, \mathbf{x}_{i'}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_{i'}) \rangle$ where $\langle ., . \rangle$ is the inner-product in the RKHS. Let denote $\kappa(\mathbf{x}_i, \mathbf{x}_{i'})$ by $\kappa_{ii'}$, and introduce the kernel matrix $\mathbf{K} = (\kappa_{ii'})$. Then, by expanding (2) in terms of inner-products it comes (see for example [41, 11]):

$$\mathrm{SSE}(\mathbf{Y}) = \sum_{i=1}^{n} \kappa_{ii} - \sum_{i=1}^{n} \sum_{i'=1}^{n} \kappa_{ii'} \sum_{j=1}^{k} \frac{1}{n_j} y_{ij} y_{i'j}$$

Next, we introduce the pairwise comparison matrix of order n, $\mathbf{X} = (x_{ii'})$ whose general term $x_{ii'} = \sum_{j=1}^{k} \frac{1}{n_j} y_{ij} y_{i'j}$ boils down to $x_{ii'} = \frac{1}{n_j}$ if $\mathbf{x}_i, \mathbf{x}_{i'} \in C_j$ and $x_{ii'} = 0$ otherwise. Using matrix notations one has the following relationship (see for example [10, 31]):

$$\mathbf{X} = \mathbf{Y}(\mathbf{Y}^\top \mathbf{Y})^{-1} \mathbf{Y}^\top \tag{4}$$

Then it is possible to define SSE w.r.t. $\mathbf{X}$:

$$\mathrm{SSE}(\mathbf{X}) = \mathrm{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{X}))$$

where Tr(.) is the matrix trace operator. More specifically, [31] showed that Problem (3) is equivalent to[5]:

$$\begin{aligned} &\min_{\mathbf{X} \in \mathbb{R}^{n \times n}} \mathrm{SSE}(\mathbf{X}) = \mathrm{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{X})) \\ &\text{s.t. } \mathbf{X} \geq \mathbf{0}_n, \mathbf{X} = \mathbf{X}^\top, \mathbf{X}\mathbf{e}_n = \mathbf{e}_n, \mathbf{X} = \mathbf{X}^2, \mathrm{Tr}(\mathbf{X}) = k. \end{aligned} \tag{5}$$

In our case, since we follow the spectral clustering methodology, we assume that $\mathbf{K} \geq \mathbf{0}_n$ (negative values are replaced by 0 if need be) is the weighted adjacency matrix of an undirected graph whose vertices represent the objects and the weight of edges the similarity (or affinity) value between pairs of objects. Then, the graph partitioning problem consists in approximating $\mathbf{K}$ with a doubly stochastic and idempotent matrix $\mathbf{X}$ (a clustering matrix), which properly encodes a partition using matrices.

Under the constraints of Problem (5), we can relate the graph-based formulation of SSE and the Frobenius inner-product and distance between $\mathbf{K}$ and $\mathbf{X}$. Recall that for any two matrices $\mathbf{A}$ and $\mathbf{B}$ in $\mathbb{R}^{n \times n}$, the Frobenius inner-product, denoted $\langle \mathbf{A}, \mathbf{B} \rangle_F$, is given by $\mathrm{Tr}(\mathbf{A}^\top \mathbf{B})$. The related Frobenius norm is therefore defined by $\|\mathbf{A}\|_F = \sqrt{\mathrm{Tr}(\mathbf{A}^\top \mathbf{A})}$ for any $\mathbf{A} \in \mathbb{R}^{n \times n}$.

Concerning the SSE objective function, observe firstly that $\min_{\mathbf{X}} \mathrm{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{X})) \Leftrightarrow \max_{\mathbf{X}} \mathrm{Tr}(\mathbf{K}\mathbf{X}) \Leftrightarrow \max_{\mathbf{X}} \langle \mathbf{K}, \mathbf{X} \rangle_F$, since $\mathbf{K}^\top = \mathbf{K}$. Secondly, according to Corollaries 1 and 2, one has $\|\mathbf{X}\|_F^2 = \mathrm{Tr}(\mathbf{X}^\top \mathbf{X}) = \mathrm{Tr}(\mathbf{X}^2) = \mathrm{Tr}(\mathbf{X}) = k$, the number of clusters, which is a parameter and not a variable of the model. Using this fact, it comes: $\min_{\mathbf{X}} \mathrm{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{X})) \Leftrightarrow \min_{\mathbf{X}} \|\mathbf{K}\|_F^2 + \|\mathbf{X}\|_F^2 - 2\langle \mathbf{K}, \mathbf{X} \rangle_F \Leftrightarrow \min_{\mathbf{X}} \|\mathbf{K} - \mathbf{X}\|_F^2$. Consequently, we can equivalently replace $\min_{\mathbf{X}} \mathrm{Tr}(\mathbf{K}(\mathbf{I}_n - \mathbf{X}))$ in Problem (5) with $\max_{\mathbf{X}} \langle \mathbf{K}, \mathbf{X} \rangle_F$ or $\min_{\mathbf{X}} \|\mathbf{K} - \mathbf{X}\|_F^2$.

[5]Note that [31] was not aware of Sinkhorn's result [33] that we recalled in Theorem 1.

Thereby, as stated above, Problem (5) which is based on SSE, can actually be seen as approximating the kernel matrix $\mathbf{K}$ by a clustering matrix $\mathbf{X}$ with k clusters under the Frobenius distance criterion.

However, it is important to underline that if one removes the constraint $\mathrm{Tr}(\mathbf{X}) = k$ in Problem (5), then the equivalence between minimizing $\mathrm{SSE}(\mathbf{X})$ and minimizing $\|\mathbf{K} - \mathbf{X}\|_F$ does not hold any more. More precisely, if the number of clusters is not fixed, then clearly, the $\mathrm{SSE}(\mathbf{X})$ criterion is minimized for the trivial partition with n clusters, $\mathbf{X} = \mathbf{I}_n$, since $\mathrm{SSE}(\mathbf{I}_n) = 0$. On the contrary, in this case, the Frobenius distance criterion naturally penalizes the number of clusters through the term $\|\mathbf{X}\|_F^2$ since $\min_{\mathbf{X}} \|\mathbf{K} - \mathbf{X}\|_F^2 \Leftrightarrow \max_{\mathbf{X}} \langle \mathbf{K}, \mathbf{X} \rangle_F - \frac{1}{2}\|\mathbf{X}\|_F^2$. In our approach, we do not require the number of clusters to be a parameter. More generally, we argue that without any prior information on the number of clusters, one should favor the Frobenius distance criterion instead of SSE. Therefore, the method we introduce in this paper relies on the Frobenius distance as we shall further detail in section 3.

2.3. Clustering matrices, orthogonal projection matrices and necessary conditions

Problem (5) is called 0-1 semi-definite programming (SDP) in [30]. The idempotency constraint $\mathbf{X}^2 = \mathbf{X}$ echoes the integrality constraint $x^2 = x$ in 0-1 integer programming. As highlighted in the introduction, in order to approximately solve such a NP-hard problem, one can proceed in the three steps (i), (ii), (iii) that we previously recalled. This paper focuses on step (i).

In Problem (5), all types of constraints but one are linear. It is idempotency that particularly makes this optimization problem difficult to solve. In this instance, several combinations of simpler necessary conditions can be used. An important observation to make is the relationship between the assignment matrix $\mathbf{Y}$ and the clustering matrix $\mathbf{X}$. Indeed, (4) brings to light the fact that $\mathbf{X}$ is the *orthogonal projection matrix* on the subspace spanned by the columns of $\mathbf{Y}$. Consequently, $\mathbf{X}$ enjoys the following properties (see [38] for example).

Proposition 1. *Let* $\mathbf{X}$ *be an orthogonal projection matrix on a* k *dimensional*

subspace $\mathbb{E}$ *of* $\mathbb{R}^n$*, then the following relations hold:*

$$\mathbf{X} \textit{ is symmetric} : \mathbf{X}^\top = \mathbf{X} \tag{6}$$
$$\mathbf{X} \textit{ is idempotent} : \mathbf{X}^2 = \mathbf{X} \tag{7}$$
$$\mathrm{Rk}(\mathbf{X}) = \mathrm{Tr}(\mathbf{X}) = k \tag{8}$$
$$\textit{There exists } \mathbf{G} \in \mathbb{R}^{n\times k}\textit{, } k = \mathrm{Rk}(\mathbf{X})\textit{, such that } \mathbf{X} = \mathbf{G}\mathbf{G}^\top, \mathbf{G}^\top\mathbf{G} = \mathbf{I}_k \tag{9}$$
$$\mathbf{X} \textit{ has } \mathrm{Rk}(\mathbf{X}) \textit{ eigenvalues equal to 1 and } n - \mathrm{Rk}(\mathbf{X}) \textit{ equal to 0} \tag{10}$$
$$\mathbf{X} \textit{ is positive semi-definite} : \mathbf{X} \succeq \mathbf{0}_n \tag{11}$$
$$\mathbf{I}_n - \mathbf{X} \textit{ is the orthogonal projection matrix on } \mathbb{E}^\perp \tag{12}$$
$$\mathbf{X}(\mathbf{I}_n - \mathbf{X}) = (\mathbf{I}_n - \mathbf{X})\mathbf{X} = \mathbf{0}_n \tag{13}$$

where $\mathrm{Rk}(\mathbf{X})$ *denotes the rank of matrix* $\mathbf{X}$*.*

Properties (6) and (7) are in fact necessary and sufficient conditions for orthogonal projection matrices. Note that clustering matrices are orthogonal projection matrices that are doubly stochastic.

In fact, numerous clustering models can be understood as relaxed versions of Problem (5) where one or several properties listed in Proposition 1 are employed as constraints. In our case, relations (12) and (13) are particularly relevant as we shall expose in what follows.

2.4. Relaxed clustering matrices and learning doubly stochastic affinity matrices

We now turn our attention to prior works that further promote double stochasticity. Indeed, unlike several clustering methods focusing on the orthogonality condition such as low-rank matrix factorization techniques [12], Zass and Shashua in [39] underlined the non-negativity and double stochasticity conditions by putting more emphasis on a probabilistic view of cluster analysis. Accordingly, they proposed an algorithm for transforming a given affinity matrix $\mathbf{K} \geq \mathbf{0}_n$ into a doubly stochastic one, or put another way, for learning a *doubly stochastic affinity matrix*. The procedure consists in iterating the following update formula until convergence:

$$\mathbf{K}^{t+1} \leftarrow \mathbf{D}_{\mathbf{K}^t}^{1/2}\mathbf{K}^t\mathbf{D}_{\mathbf{K}^t}^{1/2} \tag{14}$$

where $\mathbf{D}_{\mathbf{K}^t}$ is the degree matrix associated to $\mathbf{K}^t = (\kappa_{ii'}^t)$. Precisely, $\mathbf{D}_{\mathbf{K}^t}$ is a diagonal matrix whose ith term on its diagonal equals $\sum_{i'=1}^n \kappa_{ii'}^t$. This

procedure is in fact, a symmetrized version of the Sinkhorn-Knopp algorithm [34] which will be denoted SSK for Symmetric Sinkhorn-Knopp in the sequel.

The SSK algorithm is actually an iterative procedure that solves the following optimization problem whose objective function is the Kullback-Leibler divergence between $\mathbf{X}$ and the given $\mathbf{K}$ [40, 36]:

$$\begin{aligned} &\min_{\mathbf{X}\in\mathbb{R}^{n\times n}} \mathrm{KL}(\mathbf{X}|\mathbf{K}) = \sum_{i=1}^{n}\sum_{i'=1}^{n} x_{ii'}\log\frac{x_{ii'}}{\kappa_{ii'}} + \kappa_{ii'} - x_{ii'} \\ &\text{s.t. } \mathbf{X} \geq \mathbf{0}_n, \mathbf{X} = \mathbf{X}^\top, \mathbf{X}\mathbf{e}_n = \mathbf{e}_n. \end{aligned} \tag{15}$$

Note that when using KL, it is mandatory to assume $\mathbf{K} > \mathbf{0}_n$.

Then, in [40], the same authors successfully introduced an alternative optimization problem using the Frobenius distance in place of the Kullback-Leibler divergence:

$$\begin{aligned} &\min_{\mathbf{X}\in\mathbb{R}^{n\times n}} \|\mathbf{K} - \mathbf{X}\|_F^2 \\ &\text{s.t. } \mathbf{X} \geq \mathbf{0}_n, \mathbf{X} = \mathbf{X}^\top, \mathbf{X}\mathbf{e}_n = \mathbf{e}_n. \end{aligned} \tag{16}$$

Problem (16) is convex and can be efficiently solved by using Von Neumann successive projection lemma [25] or its more general form known as Projections On Convex Sets (POCS) (that we shall employ afterward as well). Their algorithm is denoted DSN for Doubly Stochastic Normalization, and similarly to SSK, it is an iterative procedure. Problems (15) and (16) can be solved efficiently and do not have any hyper-parameter to tune. They also do not require to set the number of clusters.

In contrast, more recent works have integrated additional constraints in the goal of further bridging the gap between a doubly stochastic matrix and a clustering matrix. Unlike the iterative procedures SSK and DSN, but similarly to low-rank matrix factorization techniques [12], these papers assume that the number of clusters k is known and integrate the constraint $\mathrm{Rk}(\mathbf{X}) = k$, following (8) and Corollary 2. Using spectral graph theory, one can equivalently use the related Laplacian matrix $\mathbf{L}_\mathbf{X}$ instead of $\mathbf{X}$ in this context. Recall that given a weighted adjacency matrix $\mathbf{X} = (x_{ii'})$ of order n, the related degree matrix $\mathbf{D}_\mathbf{X} = (d_{ii'})$ is such that $d_{ii} = \sum_{i'=1}^{n} x_{ii'}$ and $d_{ii'} = 0$ whenever $i \neq i'$, and the related Laplacian matrix $\mathbf{L}_\mathbf{X}$ is the $n \times n$ matrix given by $\mathbf{L}_\mathbf{X} = \mathbf{D}_\mathbf{X} - \mathbf{X}$. In this case, if $\mathbf{X}$ contains exactly k connected components then $\mathrm{Rk}(\mathbf{L}_\mathbf{X}) = n - k$. Nonetheless, the rank constraint

makes the problem non-convex. To overcome this drawback, the sum of the k smallest eigenvalues of $\mathbf{L_X}$ is penalized and by applying Ky Fan's Theorem [16], [27] eventually proposes to solve the following problem:

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times n},\mathbf{G}\in\mathbb{R}^{n\times k}} \|\mathbf{K}-\mathbf{X}\|_F^2 + 2\lambda \mathrm{Tr}(\mathbf{G}^\top \mathbf{L}_{\mathbf{X}^s}\mathbf{G}) \tag{17}$$
$$\text{s.t.} \begin{cases} \mathbf{X} \geq \mathbf{0}_n, \mathbf{X}\mathbf{e}_n = \mathbf{e}_n, \\ \mathbf{X}^s = \frac{\mathbf{X}+\mathbf{X}^\top}{2}, \mathbf{L}_{\mathbf{X}^s} = \mathbf{D}_{\mathbf{X}^s} - \mathbf{X}^s, \\ \mathbf{G}^\top\mathbf{G} = \mathbf{I}_k. \end{cases}$$

where $\lambda > 0$ is a penalty hyper-parameter.

Constraining the Laplacian matrix rank while learning a doubly stochastic matrix, has been studied in other papers [37, 28]. A common point of these latter works is the combination of several conditions expressed through different representations but which are mutually related. In that respect, $\mathbf{X}$, $\mathbf{L_X}$ and $\mathbf{G}$ are used in a same model and this provides flexibility when framing the search space. However, this type of modeling leads to more complex optimization problems.

In our approach, we only use $\mathbf{X}$ and $\mathbf{L_X}$ and leverage relationships between these two matrices that, to our knowledge, has not been studied in clustering models so far. Moreover, our model does not require the number of clusters for learning the affinity matrix. From this viewpoint, our main competing methods are SSK and DSN to which we compare ourselves latter on.

3. Learning Doubly Stochastic and Nearly Idempotent affinity matrix (DSNI)

3.1. Clustering matrices with no pre-defined number of clusters

Our approach, denoted DSNI, aims to obtain a *Doubly Stochastic and Nearly Idempotent affinity matrix*. The number of clusters is not a parameter of our model. Indeed, we argue that the relevance of an affinity matrix should be independent of k and that the same affinity matrix could provide partitions that are close to the ground-truth with varying number of clusters (such as nested partitions of the ground-truth). In practice, we stick to the spectral clustering methodology, where the question of setting k is addressed in steps $(\iota\iota)$ and $(\iota\iota\iota)$ and not in step (ι).

As a result, we discard the constraint $\mathrm{Tr}(\mathbf{X}) = k$. Hence, $\mathbf{X}$ is a clustering matrix whose number of clusters is assumed to be unknown. The graph

partitioning we are interested in, relies on the Frobenius distance criterion which, unlike the SSE criterion, avoids the trivial solution $\mathbf{X} = \mathbf{I}_n$ as we already pointed out in sub-section 2.2. Accordingly, DSNI addresses the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{X}\in\mathbb{R}^{n\times n}} \|\mathbf{K}-\mathbf{X}\|_F^2 \\ &\text{s.t. } \mathbf{X}\geq \mathbf{0}_n, \mathbf{X}=\mathbf{X}^\top, \mathbf{X}\mathbf{e}_n=\mathbf{e}_n, \mathbf{X}^2=\mathbf{X}. \end{aligned} \tag{18}$$

Problem (18) is NP-hard and our purpose in what follows, is to introduce a new relaxed problem that can be efficiently solved.

3.2. Clustering matrices and their associated Laplacian matrices

Our approach relies on the following properties between a clustering matrix $\mathbf{X}$ and its associated Laplacian matrix. Since $\mathbf{X}$ is doubly stochastic, its degree matrix $\mathbf{D}_\mathbf{X} = \mathbf{I}_n$ the identity matrix and thus $\mathbf{L}_\mathbf{X} = \mathbf{I}_n - \mathbf{X}$. By using this latter linear relationship between $\mathbf{X}$ and $\mathbf{L}_\mathbf{X}$, it is easy to deduce, from the already exposed properties of $\mathbf{X}$, the conditions that $\mathbf{L}_\mathbf{X}$ should satisfy:

$$\left\{\begin{array}{l} \mathbf{X}+\mathbf{L}_\mathbf{X}=\mathbf{I}_n, \\ \mathbf{X}\geq\mathbf{0}_n, \\ \mathbf{X}=\mathbf{X}^\top, \\ \mathbf{X}\mathbf{e}_n=\mathbf{e}_n, \\ \mathbf{X}^2=\mathbf{X}. \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} \mathbf{X}+\mathbf{L}_\mathbf{X}=\mathbf{I}_n, \\ \mathbf{L}_\mathbf{X}\leq\mathbf{I}_n, \\ \mathbf{L}_\mathbf{X}=\mathbf{L}_\mathbf{X}^\top, \\ \mathbf{L}_\mathbf{X}\mathbf{e}_n=\mathbf{n}_n, \\ \mathbf{L}_\mathbf{X}^2=\mathbf{L}_\mathbf{X}. \end{array}\right. \tag{19}$$

where $\mathbf{n}_n$ is the n dimensional null vector.

$\mathbf{L}_\mathbf{X}$ is symmetric and idempotent thus it is also an orthogonal projection matrix. More specifically, according to Proposition 1, $\mathbf{L}_\mathbf{X}$ is the *unique complementary orthogonal projection matrix* of $\mathbf{X}$ and moreover, $\mathbf{X}\mathbf{L}_\mathbf{X} = \mathbf{L}_\mathbf{X}\mathbf{X} = \mathbf{0}_n$.

Therefore, by applying $\mathbf{X} = \mathbf{I}_n - \mathbf{L}_\mathbf{X}$ and (19), Problem (18) can be equivalently expressed w.r.t. $\mathbf{L}_\mathbf{X}$ as follows:

$$\begin{aligned} &\min_{\mathbf{L}_\mathbf{X}\in\mathbb{R}^{n\times n}} \|\mathbf{I}_n-\mathbf{K}-\mathbf{L}_\mathbf{X}\|_F^2 \\ &\text{s.t. } \mathbf{L}_\mathbf{X}\leq \mathbf{I}_n, \mathbf{L}_\mathbf{X}=\mathbf{L}_\mathbf{X}^\top, \mathbf{L}_\mathbf{X}\mathbf{e}_n=\mathbf{n}_n, \mathbf{L}_\mathbf{X}^2=\mathbf{L}_\mathbf{X}. \end{aligned} \tag{20}$$

Note that in this latter model, we use the notation $\mathbf{L}_\mathbf{X}$ for the unknown in order to employ already introduced notations. However, it should be clear that the clustering matrix $\mathbf{X}$ is not involved in Problem (20). It can be obtained afterwards through the relation $\mathbf{X} = \mathbf{I}_n - \mathbf{L}_\mathbf{X}$.

3.3. Joint learning of $\mathbf{X}$ *and* $\mathbf{L_X}$ *and the DSNI model*

We can mix both Problems (18) and (20) into:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{L_X}\in\mathbb{R}^{n\times n}} \|\mathbf{K}-\mathbf{X}\|_F^2 + \|\mathbf{I}_n-\mathbf{K}-\mathbf{L_X}\|_F^2 \\ &\text{s.t. } \begin{cases} \mathbf{X}\geq\mathbf{0}_n, \mathbf{X}=\mathbf{X}^\top, \mathbf{X}\mathbf{e}_n=\mathbf{e}_n, \mathbf{X}^2=\mathbf{X}, \\ \mathbf{L_X}\leq\mathbf{I}_n, \mathbf{L_X}=\mathbf{L_X}^\top, \mathbf{L_X}\mathbf{e}_n=\mathbf{n}_n, \mathbf{L_X}^2=\mathbf{L_X}. \end{cases} \end{aligned} \tag{21}$$

This does not seem interesting at first sight because $\mathbf{X}$ and $\mathbf{L_X}$ are independent and solve twice the same problem but with possibly two distinct solutions since the problem is not convex. Yet, we can link both orthogonal projection matrices by making one the complementary of the other *via* $\mathbf{X}+\mathbf{L_X}=\mathbf{I}_n$. Under this condition, $\mathbf{X}^2=\mathbf{X}$ and $\mathbf{L_X}^2=\mathbf{L_X}$ are equivalent to $\mathbf{X}\mathbf{L_X}=\mathbf{0}_n$ since one has:

$$\begin{cases} \mathbf{X}+\mathbf{L_X}=\mathbf{I}_n \\ \mathbf{X}^2=\mathbf{X} \end{cases} \Leftrightarrow \begin{cases} \mathbf{X}+\mathbf{L_X}=\mathbf{I}_n \\ \mathbf{L_X}^2=\mathbf{L_X} \end{cases} \Leftrightarrow \begin{cases} \mathbf{X}+\mathbf{L_X}=\mathbf{I}_n \\ \mathbf{X}\mathbf{L_X}=\mathbf{0}_n \end{cases} \tag{22}$$

As a consequence, a *joint learning model of* $\mathbf{X}$ *and* $\mathbf{L_X}$ which is equivalent to Problems (18) and (20) is:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{L_X}\in\mathbb{R}^{n\times n}} \|\mathbf{K}-\mathbf{X}\|_F^2 + \|\mathbf{I}_n-\mathbf{K}-\mathbf{L_X}\|_F^2 \\ &\text{s.t. } \begin{cases} \mathbf{X}\geq\mathbf{0}_n, \mathbf{X}=\mathbf{X}^\top, \mathbf{X}\mathbf{e}_n=\mathbf{e}_n, \\ \mathbf{L_X}\leq\mathbf{I}_n, \mathbf{L_X}=\mathbf{L_X}^\top, \mathbf{L_X}\mathbf{e}_n=\mathbf{n}_n, \\ \mathbf{X}+\mathbf{L_X}=\mathbf{I}_n, \mathbf{X}\mathbf{L_X}=\mathbf{0}_n. \end{cases} \end{aligned} \tag{23}$$

Problem (23) is still NP-hard, but this formulation makes it possible to relax the clustering problem in a new manner. We actually keep the linear relationship $\mathbf{X}+\mathbf{L_X}=\mathbf{I}_n$ in the set of constraints but we discard the quadratic one $\mathbf{X}\mathbf{L_X}=\mathbf{0}_n$ which is difficult to handle. In return, we add in the objective function a penalization term, $\|\mathbf{X}\mathbf{L_X}\|_F^2$, to encourage $\mathbf{X}$ and $\mathbf{L_X}$ to be *nearly idempotent*. More formally, this results in the following model:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{L_X}\in\mathbb{R}^{n\times n}} \frac{1}{2}\|\mathbf{K}-\mathbf{X}\|_F^2 + \frac{1}{2}\|\mathbf{I}_n-\mathbf{K}-\mathbf{L_X}\|_F^2 + \frac{\mu}{2}\|\mathbf{X}\mathbf{L_X}\|_F^2 \\ &\text{s.t. } \begin{cases} \mathbf{X}\geq\mathbf{0}_n, \mathbf{X}=\mathbf{X}^\top, \mathbf{X}\mathbf{e}_n=\mathbf{e}_n, \\ \mathbf{L_X}\leq\mathbf{I}_n, \mathbf{L_X}=\mathbf{L_X}^\top, \mathbf{L_X}\mathbf{e}_n=\mathbf{n}_n, \\ \mathbf{X}+\mathbf{L_X}=\mathbf{I}_n. \end{cases} \end{aligned} \tag{24}$$

where $\mu \geq 0$ is a penalty hyper-parameter.

Our model relies on two variables $\mathbf{X}$ and $\mathbf{L}_{\mathbf{X}}$ that are linked to each other through the linear constraint $\mathbf{X} + \mathbf{L}_{\mathbf{X}} = \mathbf{I}_n$. By dropping strict idempotency of the unknowns, the search space consists of only linear constraints which make the problem more tractable. Nevertheless, DSNI does emphasize idempotency of $\mathbf{X}$ and $\mathbf{L}_{\mathbf{X}}$ through the penalization term $\|\mathbf{X}\mathbf{L}_{\mathbf{X}}\|_F^2$.

In the goal of emphasizing the innovative points of our proposal w.r.t. similar previous works, we indicate the main distinctions between DSNI on the one hand, and DSN and SSK on the other hand.
Unlike SSK which is based on the Kullback-Leibler divergence, DSNI uses the Frobenius distance as a loss function. It was already shown in [40, 36] that the latter criterion provided better clustering performances in comparison to the former one. Our experimental results provide similar evidences.
As far as the DSN approach is concerned, it uses the Frobenius distance similarly to DSNI. In fact, our DSNI method generalizes the DSN approach which is recovered from the former model by setting $\mu = 0$.
Unlike SSK and DSN, DSNI emphasizes idempotency in order to encourage the learnt affinity matrix to be closer to a clustering matrix. As we shall see in the section devoted to experiments, this can indeed lead to better performances.

3.4. Optimization using the ADMM

In order to solve Problem (24), we propose to apply an Alternating Direction Method of Multipliers (ADMM) scheme (see for example [4]). ADMM is an efficient algorithm for convex optimization problems and is popular in the machine learning community as mentioned in [4]. It relies on the properties of the augmented Lagrangian and provides a flexible framework to deal with several types of variables and/or constraints and can benefit from distributed computing. In our clustering model, there are two variables and ADMM is appropriate in this case, since it allows us to alternate the optimization procedure between $\mathbf{X}$ and $\mathbf{L}_{\mathbf{X}}$ in an efficient way. However, Problem (24) is not convex but bi-convex: when $\mathbf{X}$ is fixed, the problem w.r.t. $\mathbf{L}_{\mathbf{X}}$ is convex and *vice-versa*. More precisely, when $\mathbf{X}$ ($\mathbf{L}_{\mathbf{X}}$) is constant, then the objective function and the constraints are, respectively, quadratic and linear in the variable $\mathbf{L}_{\mathbf{X}}$ ($\mathbf{X}$ respectively). In this situation, an ADMM approach can still be applied as indicated in [4, Section 9.2] but convergence to a stationary point is not guaranteed. More generally, the convergence of ADMM for non-convex problems is still an active research topic (see [22] for example).

The different steps of our (scaled) ADMM procedure are:

1. Update $\mathbf{L}_{\mathbf{X}}^{t+1}$ with $\mathbf{X}^t$ fixed:

$$\begin{aligned}\mathbf{L}_{\mathbf{X}}^{t+1} \leftarrow \underset{\mathbf{L}_{\mathbf{X}} \in \mathbb{R}^{n \times n}}{\arg\min} \frac{1}{2}\|\mathbf{I}_n - \mathbf{K} - \mathbf{L}_{\mathbf{X}}\|_F^2 + \frac{\mu}{2}\|\mathbf{X}^t \mathbf{L}_{\mathbf{X}}\|_F^2 \\ + \frac{\rho}{2}\|\mathbf{X}^t + \mathbf{L}_{\mathbf{X}} - \mathbf{I}_n + \mathbf{U}^t\|_F^2 \\ \text{s.t. } \mathbf{L}_{\mathbf{X}} \leq \mathbf{I}_n, \mathbf{L}_{\mathbf{X}} = \mathbf{L}_{\mathbf{X}}^\top, \mathbf{L}_{\mathbf{X}}\mathbf{e}_n = \mathbf{n}_n.\end{aligned} \tag{25}$$

2. Update $\mathbf{X}^{t+1}$ with $\mathbf{L}_{\mathbf{X}}^{t+1}$ fixed:

$$\begin{aligned}\mathbf{X}^{t+1} \leftarrow \underset{\mathbf{X} \in \mathbb{R}^{n \times n}}{\arg\min} \frac{1}{2}\|\mathbf{K} - \mathbf{X}\|_F^2 + \frac{\mu}{2}\|\mathbf{X}\mathbf{L}_{\mathbf{X}}^{t+1}\|_F^2 \\ + \frac{\rho}{2}\|\mathbf{X} + \mathbf{L}_{\mathbf{X}}^{t+1} - \mathbf{I}_n + \mathbf{U}^t\|_F^2 \\ \text{s.t. } \mathbf{X} \geq \mathbf{0}_n, \mathbf{X} = \mathbf{X}^\top, \mathbf{X}\mathbf{e}_n = \mathbf{e}_n.\end{aligned} \tag{26}$$

3. Update $\mathbf{U}^{t+1}$:

$$\mathbf{U}^{t+1} \leftarrow \mathbf{U}^t + \mathbf{X}^{t+1} + \mathbf{L}_{\mathbf{X}}^{t+1} - \mathbf{I}_n \tag{27}$$

4. Repeat 1., 2., 3. until a stopping criterion is satisfied.

3.5. Solving sub-problems using POCS

Sub-problems (25) and (26) are convex and can be solved by a Projection On Convex Sets (POCS) procedure. In brief, POCS algorithms are designed for the convex feasibility problem where one seeks a point that belongs to the intersection of several convex subsets of a given space. POCS typically consists in projecting onto each convex subset in a sequential and cyclic fashion until convergence to a fixed point (see [3] for a review and [7] for its relationships with proximal splitting methods including ADMM).

To make easier the text to read, we introduce the following notations for the convex subsets of $\mathbb{R}^{n \times n}$ we deal with:

- $\mathcal{U}_{\mathbf{I}} = \{\mathbf{X} \in \mathbb{R}^{n \times n} : \mathbf{X} \leq \mathbf{I}_n\}$,
- $\mathcal{L}_{\mathbf{0}} = \{\mathbf{X} \in \mathbb{R}^{n \times n} : \mathbf{X} \geq \mathbf{0}_n\}$,
- $\mathcal{S} = \{\mathbf{X} \in \mathbb{R}^{n \times n} : \mathbf{X} = \mathbf{X}^\top\}$,

- $\mathcal{D}_{\mathbf{n}} = \{\mathbf{X} \in \mathbb{R}^{n\times n} : \mathbf{X}\mathbf{e}_n = \mathbf{X}^\top \mathbf{e}_n = \mathbf{n}_n\}$,
- $\mathcal{D}_{\mathbf{e}} = \{\mathbf{X} \in \mathbb{R}^{n\times n} : \mathbf{X}\mathbf{e}_n = \mathbf{X}^\top \mathbf{e}_n = \mathbf{e}_n\}$.

Similarly, we denote the projection operators on these subsets by $\Pi_{\mathcal{U}_\mathbf{I}}$, $\Pi_{\mathcal{L}_\mathbf{0}}$, $\Pi_{\mathcal{S}}$, $\Pi_{\mathcal{D}_\mathbf{n}}$ and $\Pi_{\mathcal{D}_\mathbf{e}}$, respectively.

We propose to address Sub-problem (25) as follows:

1. Solve the unconstrained problem:

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow \underset{\mathbf{L}_\mathbf{X}\in\mathbb{R}^{n\times n}}{\arg\min} \frac{1}{2}\|\mathbf{I}_n - \mathbf{K} - \mathbf{L}_\mathbf{X}\|_F^2 + \frac{\mu}{2}\|\mathbf{X}^t\mathbf{L}_\mathbf{X}\|_F^2 + \frac{\rho}{2}\|\mathbf{X}^t + \mathbf{L}_\mathbf{X} - \mathbf{I}_n + \mathbf{U}^t\|_F^2 \tag{28}$$

2. Project $\widehat{\mathbf{L}_\mathbf{X}}$ onto $\mathcal{S}$:

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow \Pi_{\mathcal{S}}\widehat{\mathbf{L}_\mathbf{X}} \tag{29}$$

3. Project $\widehat{\mathbf{L}_\mathbf{X}}$ onto $\mathcal{D}_\mathbf{n}$:

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow \Pi_{\mathcal{D}_\mathbf{n}}\widehat{\mathbf{L}_\mathbf{X}} \tag{30}$$

4. Project $\widehat{\mathbf{L}_\mathbf{X}}$ onto $\mathcal{U}_\mathbf{I}$:

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow \Pi_{\mathcal{U}_\mathbf{I}}\widehat{\mathbf{L}_\mathbf{X}} \tag{31}$$

5. Repeat 3., 4. until a stopping criterion is satisfied.

$\Pi_{\mathcal{D}_\mathbf{n}}$ and $\Pi_{\mathcal{U}_\mathbf{I}}$ preserve symmetry and thus there is no need to apply $\Pi_{\mathcal{S}}$ after the first iteration. Interestingly, all Sub-problems (28)-(31) have closed-form solutions.

Proposition 2. *The solutions to* (28), (29), (30) *and* (31) *are respectively given by:*

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow \left((1+\rho)\mathbf{I}_n + \mu(\mathbf{X}^t)^2\right)^{-1}\left(\mathbf{I}_n - \mathbf{K} + \rho(\mathbf{I}_n - \mathbf{X}^t - \mathbf{U}^t)\right) \tag{32}$$

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow \frac{\widehat{\mathbf{L}_\mathbf{X}} + \widehat{\mathbf{L}_\mathbf{X}}^\top}{2} \tag{33}$$

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow (\mathbf{I}_n - \mathbf{J}_n)\widehat{\mathbf{L}_\mathbf{X}}(\mathbf{I}_n - \mathbf{J}_n) \tag{34}$$

$$\widehat{\mathbf{L}_\mathbf{X}} \leftarrow \min(\widehat{\mathbf{L}_\mathbf{X}}, \mathbf{I}_n) \text{ (\textit{elementwise} min \textit{operator})} \tag{35}$$

Note that (34) is the double centering operator.

The proof of Proposition 2 is given in the supplementary materials.

We now turn our attention to the second sub-problem with unknown $\mathbf{X}$. We proceed in a similar fashion in order to tackle Sub-problem (26):

1. Solve the unconstrained problem:

$$\begin{aligned}\widehat{\mathbf{X}} \leftarrow \underset{\mathbf{L}_{\mathbf{X}} \in \mathbb{R}^{n \times n}}{\arg\min} \frac{1}{2}\|\mathbf{K} - \mathbf{X}\|_F^2 + \frac{\mu}{2}\|\mathbf{X}\mathbf{L}_{\mathbf{X}}^{t+1}\|_F^2 \\ + \frac{\rho}{2}\|\mathbf{X} + \mathbf{L}_{\mathbf{X}}^{t+1} - \mathbf{I}_n + \mathbf{U}^t\|_F^2\end{aligned} \tag{36}$$

2. Project $\widehat{\mathbf{X}}$ onto $\mathcal{S}$:

$$\widehat{\mathbf{X}} \leftarrow \Pi_{\mathcal{S}}\widehat{\mathbf{X}} \tag{37}$$

3. Project $\widehat{\mathbf{X}}$ onto $\mathcal{D}_{\mathbf{e}}$:

$$\widehat{\mathbf{X}} \leftarrow \Pi_{\mathcal{D}_{\mathbf{n}}}\widehat{\mathbf{X}} \tag{38}$$

4. Project $\widehat{\mathbf{X}}$ onto $\mathcal{L}_{\mathbf{0}}$:

$$\widehat{\mathbf{X}} \leftarrow \Pi_{\mathcal{L}_{\mathbf{0}}}\widehat{\mathbf{X}} \tag{39}$$

5. Repeat 3., 4. until a stopping criterion is satisfied.

The closed-form solutions are provided below.

Proposition 3. *The solutions to* (36), (37), (38) *and* (39) *are respectively given by:*

$$\widehat{\mathbf{X}} \leftarrow \left(\mathbf{K} + \rho(\mathbf{I}_n - \mathbf{L}_{\mathbf{X}}^{t+1} - \mathbf{U}^t)\right)\left((1+\rho)\mathbf{I}_n + \mu(\mathbf{L}_{\mathbf{X}}^{t+1})^2\right)^{-1} \tag{40}$$

$$\widehat{\mathbf{X}} \leftarrow \frac{\widehat{\mathbf{X}} + \widehat{\mathbf{X}}^\top}{2} \tag{41}$$

$$\widehat{\mathbf{X}} \leftarrow (\mathbf{I}_n - \mathbf{J}_n)\widehat{\mathbf{X}}(\mathbf{I}_n - \mathbf{J}_n) + \mathbf{J}_n \tag{42}$$

$$\widehat{\mathbf{X}} \leftarrow \max(\widehat{\mathbf{X}}, \mathbf{0}_n) \text{ } \textit{(elementwise } \max \textit{ operator)} \tag{43}$$

The proof of Proposition 3 is similar to that of Proposition 2.

Algorithm 1 DSNI: Learning Doubly Stochastic and Nearly Idempotent affinity matrix.

Require: $\mathbf{K}$, μ ($\sqrt{n}$ default), ρ (1 default)
Ensure: $(\mathbf{X}^*, \mathbf{L}^*_{\mathbf{X}})$

1: $t \leftarrow 0$, $\mathbf{U}^t \leftarrow \mathbf{0}_n$, $\mathbf{X}^t \leftarrow \mathbf{K}$ ▷ Initialization
2: **repeat** ▷ ADMM loop
3: Update $\widehat{\mathbf{L}_{\mathbf{X}}}$ with (32) ▷ Sub-pb (25)
4: Update $\widehat{\mathbf{L}_{\mathbf{X}}}$ with (33)
5: **repeat**
6: Update $\widehat{\mathbf{L}_{\mathbf{X}}}$ with (34)
7: Update $\widehat{\mathbf{L}_{\mathbf{X}}}$ with (35)
8: **until** Stopping criterion is satisfied
9: $\mathbf{L}^{t+1}_{\mathbf{X}} \leftarrow \widehat{\mathbf{L}_{\mathbf{X}}}$
10: Update $\widehat{\mathbf{X}}$ with (40) ▷ Sub-pb (26)
11: Update $\widehat{\mathbf{X}}$ with (41)
12: **repeat**
13: Update $\widehat{\mathbf{X}}$ with (42)
14: Update $\widehat{\mathbf{X}}$ with (43)
15: **until** Stopping criterion is satisfied
16: $\mathbf{X}^{t+1} \leftarrow \widehat{\mathbf{X}}$
17: Update $\mathbf{U}^{t+1}$ with (27)
18: $t \leftarrow t + 1$
19: **until** Stopping criterion is satisfied
20: $(\mathbf{X}^*, \mathbf{L}^*_{\mathbf{X}}) \leftarrow (\mathbf{X}^{t+1}, \mathbf{L}^{t+1}_{\mathbf{X}})$

3.6. DSNI algorithm

We wrap up all previous results in Algorithm 1. Sub-problems (25) and (26) are respectively carried out through lines 3 to 8 and 10 to 15. The stopping criterion in these two cases is the convergence of $\widehat{\mathbf{L}_{\mathbf{X}}}$ and $\widehat{\mathbf{X}}$ to fixed points. As for the global ADMM scheme, the stopping criterion in line 19 is primarily based on the convergence to 0 of the primal residuals $\|\mathbf{X}^t + \mathbf{L}^t_{\mathbf{X}} - \mathbf{I}_n\|_F$. However, since Problem (24) is not convex, the convergence of ADMM in this case is not guaranteed and in practice, we also provide a maximal number of iterations.

Algorithm 1 involves two kinds of penalty hyper-parameters, μ and ρ. The former one concerns the penalization of $\mathbf{X}\mathbf{L}_{\mathbf{X}} = \mathbf{0}_n$, while the latter

one is inherent to ADMM which deals here with the bi-affine constraint $\mathbf{X} + \mathbf{L}_{\mathbf{X}} - \mathbf{I}_n = \mathbf{0}_n$. As highlighted above, our approach does not require to set a number of clusters for learning the affinity matrix.

3.7. Geometric property of DSNI affinity matrices

Problem (24) can also be used for kernel (or metric) learning, in the sense that one can derive a positive semi-definite affinity matrix from its solutions.

Proposition 4. *Let* $(\mathbf{X}^*, \mathbf{L}^*_{\mathbf{X}})$ *be a solution to Problem* (24)*. Then the following matrix* $\mathbf{K}^*$ *is non-negative, symmetric and positive semi-definite:*

$$\mathbf{K}^* = 2\mathbf{I}_n - \mathbf{L}^*_{\mathbf{X}} = \mathbf{I}_n + \mathbf{X}^* \tag{44}$$

The proof of Proposition 4 is given in the supplementary materials.

In other words, adding 1 to the diagonal entries of $\mathbf{X}^*$ leads to a positive semi-definite matrix. Note, however, that this diagonal shift does not change the eigenvectors which remain the same ones for both $\mathbf{X}^*$ and $\mathbf{K}^*$.

4. Experiments

4.1. Settings

We compared our approach to methods that aim to obtain doubly stochastic affinity matrices without any prior on the number of clusters. More precisely, we compared the clustering outputs provided by the spectral clustering approach but using the following different affinity matrices as input:

- $\mathbf{K}$, a given initial affinity matrix which in our case is a kernel matrix,
- the affinity matrix of the k nearest neighbor graph, with $k = \lceil n/10 \rceil$, that is extracted from $\mathbf{K}$,
- the doubly stochastic affinity matrix obtained when DSN is applied to $\mathbf{K}$,
- the doubly stochastic affinity matrix obtained when SSK is applied to $\mathbf{K}$,
- the doubly stochastic and nearly idempotent affinity matrix obtained when DSNI is applied to $\mathbf{K}$.

These methods will be referred to as K, k-NN, DSN, SSK and DSNI, respectively in what follows.
K and k-NN represent the baselines and correspond to two types of conventional spectral clustering methods [35]. The former one uses the full kernel matrix $\mathbf{K}$, whereas the second one, by employing a sparsified $\mathbf{K}$, assumes that the data rather belong to a non-linear manifold (see for example [35]). DSN [40] and SSK [39, 34] are two unsupervised learning methods whose goal is to provide a doubly stochastic affinity matrix from $\mathbf{K}$. They rely on two distinct objective functions. They represent our main rivals. Finally, DSNI is the method we promote in this work. From an empirical viewpoint, our goal is to demonstrate that taking into account idempotency in addition to doubly stochasticity is beneficial.

In all our experiments, the initial affinity matrix $\mathbf{K} = (\kappa_{ii'})$ is given by the Gaussian kernel which can be considered as a default similarity function in spectral clustering [35, 24]:

$$\kappa_{ii'} = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2\right), \forall i, i' = 1, \ldots, n. \tag{45}$$

where $\gamma = 1/p$ by default, p being the dimension of the space the vectors $\{\mathbf{x}_i\}_{i=1,\ldots,n}$ belong to. Note that since our purpose is to compare the different learning methods we have not tried to tune the hyper-parameter γ and chose the default setting used in the `LIBSVM` tool [6].

Under these circumstances, $\mathbf{K} > \mathbf{0}_n$, and the SSK algorithm can be applied. SSK and DSN do not have any hyper-parameter unlike DSNI. For our method, we used the default values $\mu = \sqrt{n}$ and $\rho = 1$ in all our experiments.

As for stopping conditions, we set an error precision threshold to 0.001. In the case of DSN, if the difference between two subsequent objective function values is lower than 0.001 then the procedure stops. Regarding SSK, the algorithm stops when all the differences between each row sum and the value one is lower than 0.001. Concerning DSNI, since it is based on ADMM, we compute a relative error following the recommendation provided in [4, Section 3.3.1]. In particular, the stopping condition is based on the primal residuals and in this case both ϵ^{abs} and ϵ^{rel} in the previous reference are set to 0.001.
In addition to the previous conditions on error measures, we also consider a maximal number of iterations for DSN and DSNI. Regarding DSN, the number of iterations cannot exceed 300. In the case of DSNI, the number of maximal iterations for ADMM and for POCS is set to 100.

Following the spectral clustering procedure, once step (ι) is carried out using one of the aforementioned method, the resulting affinity matrix is passed to step ($\iota\iota$) which computes the spectral decomposition of its related Laplacian matrix. In the final step ($\iota\iota\iota$), the k-means algorithm is applied on the feature matrix composed of the k first eigenvectors, k being set to the number of clusters which is provided by the ground-truth. Since the k-means results depend on a random initialization phase, we ran 10 times the algorithm and the partition obtaining the best objective value is selected.

Next, in order to assess the quality of the obtained partition, we compared it with the ground-truth using the Normalized Mutual Information (NMI) measure. Let $C = \{C_1, \ldots, C_k\}$ be the partition found by spectral clustering and let $D = \{D_1, \ldots, D_k\}$ be the ground-truth. NMI is an entropy-based measure. We denote by $\mathrm{H}(C)$ the entropy of C: $\mathrm{H}(C) = -\sum_{j=1}^{k} P(C_j)\log(P(C_j))$ and by $\mathrm{MI}(C, D)$ the mutual information between C and D: $\mathrm{MI}(C, D) = \sum_{j,j'=1}^{k} P(C_j, D_{j'})\log(P(C_j, D_{j'})/(P(C_j)P(D_{j'})))$. The formal definition of NMI is given by:

$$\mathrm{NMI}(C, D) = \frac{2\mathrm{MI}(C, D)}{\mathrm{H}(C) + \mathrm{H}(D)}$$

NMI values range from 0 to 1 and the higher the value is, the closer to the ground-truth the partition is. Note that we also tested with the Adjusted Rand Index (ARI) but found similar outcomes. Therefore, we chose to not mention these experimental results.

4.2. Computing environment

We implemented DSNI, SSK and DSN in `Python` then use the `sci-kit learn` library [29] to perform spectral clustering using the `SpectralClustering` function. The `sci-kit learn` library offers tools for assessing the clustering performances including the computation of the NMI measure. This is done with the `normalized_mutual_info_score` function.

4.3. Datasets

Our experiments focus on real-world case studies. We used nine datasets freely available online. They are described in Table 1 and vary w.r.t. the number of instances (n), dimensions (p) and number of clusters (k). All datasets can be accessed using the `sklearn.datasets` utilities. The Olivetti Faces, Breast cancer, Digits datasets are directly loadable. The same applies

Dataset	n # instances	p # dimensions	k # clusters
Glass	214	9	6
Ionosphere	351	34	2
Olivetti Faces	400	4096	40
Vowel	528	10	11
Breast cancer	569	30	2
Vehicle	846	18	4
Yeast	1484	8	10
Digits	1797	64	10
Segment.scale	2310	19	7

Table 1: List of datasets.

to Vowel, Segment.scale, Ionosphere (scale) but these datasets were taken from the `LIBSVM` dataset repository[6]. Using this module makes it possible to fetch data from the `openml.org` repository as well. This is the resource where the three remaining datasets listed in Table 1 come from.

4.4. Results

We report in Table 2, the NMI measures we obtained for each dataset and each method. Then, we translate these performances into rankings of the models for each dataset. These results are indicated in Table 3. In order to have a global comparison of the different approaches, we computed their respective average ranking. This overall evaluation is displayed in the last row of Table 3.

The baseline K, which is the regular spectral clustering method using the full Gaussian kernel matrix, is ranked the last with an average ranking of 4.22. The three techniques involving double stochasticity, SSK, DSN, DSNI, perform better than K. This confirms the benefit of applying unsupervised learning methods for determining doubly stochastic affinity matrices in graph-based clustering.

Among these three competing methods, our model performs the best. Furthermore, DSNI provides robust results. Indeed, when it does not give the best NMI value, it gives the second best one. DSNI average ranking is

[6] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

Dataset	K	k-NN	SSK	DSN	DSNI
Glass	0.253	0.281	0.276	0.243	**0.297**
Ionosphere	0.038	0.082	0.066	0.076	**0.131**
Olivetti Faces	0.782	0.755	0.786	0.803	**0.855**
Vowel	0.382	0.412	0.321	0.206	**0.423**
Breast cancer	0.010	**0.677**	0.010	0.010	0.670
Vehicle	0.013	0.132	0.135	**0.203**	0.171
Yeast	0.070	**0.329**	0.258	0.256	0.263
Digits	0.015	0.552	0.044	0.743	**0.767**
Segment.scale	0.012	0.010	0.341	0.449	**0.522**

Table 2: NMI performances.

Dataset	K	k-NN	SSK	DSN	DSNI
Glass	4	2	3	5	**1**
Ionosphere	5	2	4	3	**1**
Olivetti Faces	4	5	3	2	**1**
Vowel	3	2	4	5	**1**
Breast cancer	3	**1**	3	3	2
Vehicle	5	4	3	**1**	2
Yeast	5	**1**	3	4	2
Digits	5	3	4	2	**1**
Segment.scale	4	5	3	2	**1**
Average	4.22	2.77	3.33	3	**1.33**

Table 3: Rankings of methods.

1.33 and it clearly outperforms SSK and DSN whose average rankings are 3.33 and 3 respectively. This result accounts for the importance of integrating idempotency in cluster analysis.

In the experiments we conducted, DSNI also outperforms the k-NN method whose average ranking is 2.77.

5. Conclusion

Graph partitioning is a NP-hard problem. We have examined this topic from the angle of the so-called clustering matrices, which are doubly stochas-

tic and idempotent matrices. These two conditions are crucial when addressing the graph partitioning problem and many cluster analysis models actually amount to search for relaxed clustering matrices.

In this context, we have introduced a new optimization model that leverages the complementary relationships between a clustering matrix and its associated Laplacian matrix. Our relaxed model, DSNI, allows us to fit a given affinity matrix with a doubly stochastic and nearly idempotent affinity matrix. Moreover, when comparing DSNI with similar techniques, SSK and DSN, in the framework of spectral clustering, our experimental results exhibit very good performances even without tuning the two hyper-parameters it involves.

As for future work, it would be interesting to study the sensitivity of DSNI w.r.t. its hyper-parameters. An adaptive ρ in the ADMM framework may give even better clustering performances. More generally, the convergence conditions of Algorithm 1 should be further analyzed. Furthermore, it seems interesting to investigate DSNI's ability to grasp the intrinsic geometry of the data. The good behavior of DSNI as compared to k-NN is encouraging in that respect.

Acknowledgment

We would like to thank the three anonymous referees for their valuable comments. This work was supported by the Agence Nationale de la Recherche of the French government through the program "Investissements d'avenir" ANR-10-LABX-14-01.

References

[1] Ahmadi, R. H. and Tang, C. S. (1991). An operation partitioning problem for automated assembly system design. *Operations Research*, 39(5):824–835.

[2] Aloise, D., Deshpande, A., Hansen, P., and Popat, P. (2009). Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248.

[3] Bauschke, H. H. and Borwein, J. M. (1996). On projection algorithms for solving convex feasibility problems. *SIAM review*, 38(3):367–426.

[4] Boyd, S., Parikh, N., and Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

[5] Caraballo, L. E., Díaz-Báñez, J.-M., and Kroher, N. (2021). A polynomial algorithm for balanced clustering via graph partitioning. *European Journal of Operational Research*, 289(2):456–469.

[6] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

[7] Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer.

[8] De Amorim, S. G., Barthélemy, J.-P., and Ribeiro, C. C. (1992). Clustering and clique partitioning: simulated annealing and tabu search approaches. *Journal of Classification*, 9(1):17–41.

[9] Delgoshaei, A. and Ali, A. (2019). Evolution of clustering techniques in designing cellular manufacturing systems: A state-of-art review. *International Journal of Industrial Engineering Computations*, 10(2):177–198.

[10] Dhillon, I. S., Guan, Y., and Kulis, B. (2004a). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556.

[11] Dhillon, I. S., Guan, Y., and Kulis, B. (2004b). *A unified view of kernel k-means, spectral clustering and graph cuts*. Citeseer.

[12] Ding, C., He, X., and Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM.

[13] Ding, C. H., He, X., Zha, H., Gu, M., and Simon, H. D. (2001). A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 IEEE international conference on data mining*, pages 107–114. IEEE.

[14] Dorndorf, U. and Pesch, E. (1994). Fast clustering algorithms. *ORSA Journal on Computing*, 6(2):141–153.

[15] Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011). Cluster analysis 5th ed.

[16] Fan, K. (1949). On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11):652.

[17] Gan, G., Ma, C., and Wu, J. (2020). *Data clustering: theory, algorithms, and applications*. SIAM.

[18] Grötschel, M. and Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1):59–96.

[19] Hansen, P. and Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical programming*, 79(1):191–215.

[20] Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons, Inc.

[21] Hofmann, T., Schlkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171 – 1220.

[22] Hong, M., Luo, Z.-Q., and Razaviyayn, M. (2016). Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364.

[23] Marcotorchino, F. and Michaud, P. (1982). Agregation de similarites en classification automatique. *Revue de statistique appliquée*, 30(2):21–44.

[24] Nascimento, M. C. and De Carvalho, A. C. (2011). Spectral methods for graph clustering–a survey. *European Journal of Operational Research*, 211(2):221–231.

[25] Neumann, J. V. (1950). *Functional Operators (AM-22), Volume 2: The Geometry of Orthogonal Spaces. (AM-22)*. Princeton University Press.

[26] Ngai, E. W., Xiu, L., and Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert systems with applications*, 36(2):2592–2602.

[27] Nie, F., Wang, X., Jordan, M., and Huang, H. (2016). The constrained laplacian rank algorithm for graph-based clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

[28] Park, J. and Kim, T. (2017). Learning doubly stochastic affinity matrix via Davis-Kahan theorem. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 377–384. IEEE.

[29] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[30] Peng, J. and Wei, Y. (2007). Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 18(1):186–205.

[31] Peng, J. and Xia, Y. (2005). A new theoretical framework for k-means-type clustering. In *Foundations and advances in data mining*, pages 79–96. Springer.

[32] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., and Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267:664–681.

[33] Sinkhorn, R. (1968). Two results concerning doubly stochastic matrices. *The American Mathematical Monthly*, 75(6):632–634.

[34] Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348.

[35] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

[36] Wang, F., Li, P., and Konig, A. C. (2010). Learning a bi-stochastic data similarity matrix. In *2010 IEEE International Conference on Data Mining*, pages 551–560. IEEE.

[37] Wang, X., Nie, F., and Huang, H. (2016). Structured doubly stochastic matrix for graph based clustering. In *Proceedings of the 22nd ACM SIGKDD International conference on Knowledge discovery and data mining*, pages 1245–1254.

[38] Yanai, H., Takeuchi, K., and Takane, Y. (2011). Projection matrices. In *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*, pages 25–54. Springer.

[39] Zass, R. and Shashua, A. (2005). A unifying approach to hard and probabilistic clustering. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 294–301. IEEE.

[40] Zass, R. and Shashua, A. (2007). Doubly stochastic normalization for spectral clustering. In *Advances in neural information processing systems*, pages 1569–1576.

[41] Zha, H., He, X., Ding, C., Gu, M., and Simon, H. D. (2002). Spectral relaxation for k-means clustering. In *Advances in neural information processing systems*, pages 1057–1064.