Convex Hierarchical Clustering for Graph-Structured Data

Claire Donnat
Department of Statistics
Stanford University
Stanford, CA, USA
cdonnat@stanford.edu

Susan Holmes
Department of Statistics
Stanford University
Stanford, CA, USA
susan@stat.stanford.edu

***Abstract*—Convex clustering [1] is a recent stable alternative to hierarchical clustering. It formulates the recovery of progressively coalescing clusters as a regularized convex problem. While convex clustering was originally designed for handling Euclidean distances between data points, in a growing number of applications, the data is directly characterized by a similarity matrix or weighted graph. In this paper, we extend the robust hierarchical clustering approach to these broader classes of similarities. Having defined an appropriate convex objective, the crux of this adaptation lies in our ability to provide: (a) an efficient recovery of the regularization path and (b) an empirical demonstration of the use of our method. We address the first challenge through a proximal dual algorithm, for which we characterize both the theoretical efficiency as well as the empirical performance on a set of experiments. Finally, we highlight the potential of our method by showing its application to several real-life datasets, thus providing a natural extension to the current scope of applications of convex clustering.**

I. Introduction and related work

From gene sequencing to biomedical studies [2]–[5], hierarchical clustering [6], [7] is currently one of the most widely-used procedures for data analysis. The dendrogram it provides yields a complete summary of the data and bypasses the need to prespecify an adequate number of clusters. The visualization of these clusters' progressive coalescence provides a comprehensive and intuitive view of their similarities. However, hierarchical clustering is an inherently greedy procedure, which typically constructs the clusters' fusion path by iteratively aggregating (or splitting) clusters. The recovered coalescence path is dependent on the choice of the linkage function, and has also been shown to be highly sensitive to outliers and perturbations of the dataset— thus allowing the formation of spurious clusters and consequently hindering the generalizability of the analysis [8]. This is particularly problematic in applications, where such multiscale representations of the data are compared, contrasted and analyzed. Such is the case in brain connectomics, where a topic of interest is the comparison of the multiscale representations of the network woven by white matter tracts across different people or groups. In such noisy regimes, the definition of a robust and optimal hierarchical clustering takes on a particular importance.

Convex clustering. To overcome these issues, convex clustering [8]–[10] is a recent alternative formulation of hierarchical clustering as the solution of a convex optimization problem with a regularization penalty. In its original form, denoting each observation i by its corresponding vector $X_i \in \mathbb{R}^d$(with $i = 1 \cdots N$), and introducing $U_i \in \mathbb{R}^d$ the centroid of the cluster associated to point i, convex clustering solves the following objective:

$$\underset{U \in \mathbb{R}^{d \times N}}{\operatorname{argmin}} \sum_{i=1}^{N} ||X_i - U_i||^2 + \lambda \sum_{i,j=1}^{N} W_{ij} \text{Pen}(U_i - U_j) \qquad (1)$$

where W_{ij} are coupling weights (typically chosen as the k-nearest neighbors of each observation). In the previous expression, *Pen* is a penalty function (typically the ℓ_q-norm, with $q \geq 1$) which encourages coupled observations to share the same centroid. The solution path $U^{(\lambda)}$ is comparable to the coalescence path recovered by hierarchical clustering, and the regularization parameter λ, to the different levels in the hierarchical clustering dendrogram [8], [9]:

- for $\lambda = 0$, the solution of Eq. 1 is $U^{(0)} = X$, and each point belongs its own cluster.
- as λ increases, the penalty term induces the centroids U_i to fuse, until in the limit, these centroids reach a consensus value, thus forming a single cluster $U^{(\infty)}$: $\forall i, j \in \{1, N\}, \quad U_i^{(\infty)} = U_j^{(\infty)}$.

The strict convexity of the objective function of Problem 1 guarantees the existence of a globally optimal solution, as well as its robustness against perturbations [8], thus making this convex formulation an extremely appealing alternative to hierarchical clustering. We refer the reader to [11] for a thorough review of the properties of convex clustering as well as a formal analysis of its parallel with hierarchical clustering.

Contributions and related work. One of the main drawbacks of convex clustering is that the optimization procedure associated to Problem 1 is computationally more involved than the greedy optimization performed by standard hierarchical clustering. While some work has already been put into the design of efficient solutions [1], to the best of our knowledge, the derivation of algorithms for convex clustering has been restricted to the setting where data are Euclidean: observations are represented by vectors in $\mathbb{R}^d$, and similarities are simply characterized through pairwise Euclidean distances. However, in an increasing number of applications, such representations

are difficult to obtain and the data come more readily as a graph in which nodes represent observations, and edges reflect some function of similarities between data points. In connectomics for instance, correlations between brain regions are summarized by a weighted graph, which provides a more amenable support to the study of functional connectivity [12], [13]. Similarly, in social sciences, relationships between individuals are readily modeled by a graph, where edges denote interactions between users. In many of these graph-structured datasets, hierarchical clustering is an indispensable tool since it allows the analysis of the data at different scales. The derivation of a convex multiscale summary of the data with the same global optimum and robustness guarantees as its Euclidean counterpart thus represents an impactful problem with many applications – a challenge which we propose to tackle in this paper. Our contributions consist in (a) the adaptation of the convex objective posed in Problem 1 to the graph setting and (b) the derivation of two provably efficient solutions. We analyze and validate our method through a set of synthetic experiment, and show its application on several real-world datasets.

II. Problem Statement

Throughout this paper, we assume that the data comes under the form of a weighted similarity matrix K between N elements (i.e, for instance, the adjacency matrix or diffusion map associated to a graph), which we suppose to be sparse. We adopt the standard convention of referring to the i^{th} column of any given matrix M as M_i.

Positive Definite Symmetric Input Matrices. We begin by studying the case where the similarity matrix K is symmetric and positive semi definite. By direct application of the spectral lemma, we can re-write K as a dot-product in a higher-dimensional space: $K = \Phi^T\Phi$ i.e. $\forall i,j, \quad K_{ij} = \Phi(X_i)^T\Phi(X_j)$. This provides an amenable setting for the generalization of convex clustering, where the goal becomes to recover the centroids U_i associated to each implicit high-dimensional vector $\Phi(X_i)$. Since each centroid lies in the convex hull of its corresponding vectors, we require U to have the form:

$$U = \Phi(X)\pi, \quad \text{where} \quad \pi\mathbf{1} = \mathbf{1} \quad \mathbf{1}^T\pi = \mathbf{1}^T \text{ and } \pi \geq 0 \quad (2)$$

In this setting, the doubly-stochastic matrix π benefits from a bi-dimensional interpretation: the columns correspond to the centroids' representation using the original observations as dictionary, while the rows can be interpreted as soft membership assignments of observations to clusters. *However, we highlight that this constraint further adds to the complexity of the original convex clustering algorithm, and is non-trivial to implement.* Using the kernel trick, Eq. 1 can be adapted here to:

$$\underset{\pi_\in\Delta_N}{\text{argmin}} \ \text{Tr}[\pi^T K\pi - 2K\pi] + \lambda\sum_{i,j} K_{ij}\text{Pen}(\pi_i - \pi_j) \quad (3)$$

where Δ_N is the set of doubly stochastic matrices. A full derivation of this formulation can be found in Appendix B of the extended version of this paper. The next important step consists in choosing the coupling penalty, which we take here to be a mixed total-variation penalty:

$$\text{Pen}(\pi_{\cdot i} - \pi_{\cdot j}) = \alpha||\pi_i - \pi_j||_2 + (1-\alpha)||\pi_i - \pi_j||_1.$$

This choice is motivated by the fact that the ℓ_1-penalty is known to provide nested sequences of clusters [9], while the ℓ_2-penalty allows the recovery of a more stable solution. Total variation distances have also been shown to encourage the recovery of piecewise linear functions and providing solutions with sharp edge contrasts [14]– a desirable property, since this amounts to "clamping" the centroids together as they progressively coalesce.

To ease notation, for any square matrix M, we denote as $\delta^{(M)} \in \mathbb{R}^{N\times N^2}$ the $N \times N^2$-dimensional matrix of pairwise differences such that: $\forall i,j,k \leq N, \quad \delta^{(M)}_{k,(i,j)} = M_{ij}(\mathbf{e}_{ki} - \mathbf{e}_{kj})$, where $\mathbf{e}_i$ is the i^{th} cartesian basis vector. The final constrained minimization problem can thus be compactly written as:

$$\underset{\pi_\in\Delta_N}{\text{argmin}} \Big\{\text{Trace}[\pi^T K\pi - 2K\pi] + \lambda\sum_{ij} K_{ij}\Big(\alpha||\delta^{(\pi)}_{ij}||_2 + (1-\alpha)||\delta^{(\pi)}_{ij}||_1\Big)\Big\} \quad (4)$$

As for its Euclidean counterpart, the solution of Eq. 4 is consistent with its interpretation as a cluster coalescence path:

- when $\lambda = 0$, the solution of the previous equation is the identity: $\pi^{(0)} = I_N$. Indeed, it is easy to check that I_N is a solution to $\text{argmin}_{\pi\in[0,1]^N}\text{Tr}[\pi^T K\pi - 2K\pi]$. Since I_N is evidently doubly-stochastic, by strict convexity of the objective in Eq. 4, we deduce that it is the solution to the constrained problem in Eq. 4 for $\lambda = 0$.
- when $\lambda = \infty$, on the other hand, the solution of Eq. 4 must be such that $\delta^{(M)}_{k,(i,j)}$. This given by the consensus matrix $\pi^{(\infty)} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$, which is the intersection of the set $\{A \in \mathbb{R}^{N\times N} : \forall i,j, \leq N \quad A_i = A_j, A \geq 0\}$ with the set of doubly-stochastic matrices.

Discussion. The assumption that K is positive definite is by no means restrictive. Indeed, in many applications (brain connectomes, etc.), the kernel K corresponds to some transformation of a positive definite similarity (typically, to some thresholded-measure of the correlation between vertices). Even if this is not the case, Positive-Semi-Definiteness can be achieved by regularizing the kernel: $\hat{K} = K + \gamma I_n$. As in many clustering algorithms (choice of the most adequate distance metric in k-means, bandwidth in spectral clustering, etc.), the choice of the appropriate transformation is left to the data analyst.

We also highlight that, in contrast to hierarchical clustering, only for the choice $\alpha = 0$ is the algorithm ensured to output a nested sequence of clusters [9]. However, we emphasize that the goal of our paper is to extract robust multiscale representations (rather than strictly nested ones): in this case, the regularization path induces progressively coarser and coarser representations of the data (by fusing centroids). Hence, the strict convexity of the objective in Eq. 4 ensures its global optimality, which in this case is a more desirable quality than nestedness.

III. Algorithm

The main challenge consists in devising an efficient algorithm for solving the previous optimization problem. Indeed, while this problem is strongly convex, exact solvers are extremely

slow to find a solution for each value of λ, making the computation of the full regularization path almost intractable. In this paper, we propose two methods. The first is based on an adpatation of the Fast Iterative Shrinkage and Thresholding Algorithm [15], a method originally proposed by Beck and Teboulle for image deblurring in 2009 [16] and which we have selected for both its theoretical efficiency and its empirical performance. The second uses ADMM [17] to solve the corresponding problem. While neither of these methods is novel– FISTA has been applied to image deblurring [16], while ADMM has already been suggested to solve the original convex clustering problem has [8])—, our contribution lies in the adaptation of these methods to the evermore-challenging setting of Eq. 4, in which the optimization has to be done on the set of doubly stochastic matrices — a much more constrained and complicated setting than in the aforementioned problems. As shown in section V, our experiments show that FISTA produces better results, while ADMM scales better to the analysis of large graphs. For the sake of clarity, we outlay in the main text the derivation of FISTA, and leave the derivation of the ADMM algorithm to Appendix C.

Algorithm. Broadly speaking, FISTA [15] is an algorithm for efficiently solving optimization problems of the form: $\min_x f(x)+g(x)$, where g is proper convex (but not necessarily smooth, as typically for ℓ_1 penalties and indicator set functions) and the subgradients of f are Lipschitz. One of the most appealing characteristics of FISTA lies in (a) the absence of any user-defined parameters—making it a completely parameter-free method, in contrast to ADMM—and (b) a $1/k^2$-accelerated convergence rate. In the spirit of the algorithm proposed by Beck and Teboulle [16] for image denoising and deblurring under total-variation penalty, we propose to solve our similarity-based convex problem 4 using FISTA on the dual. The additional challenges that our approach faces with respect to this original method are two-fold: (a) our set of constraints is given by the graph adjacency matrix K and is thus more general than the regular 2D-grid in [16], and (b) we are optimizing over the set of doubly stochastic matrices, thus calling for the need of an efficient projection algorithm.

From primal to dual. As in the previous section, we begin by supposing that the similarity matrix K is positive semi definite. K factorizes as: $K = \Phi^T\Phi$, where, by writing $K = U\Lambda U^T$ the spectral decomposition of K, we have: $\Phi = \Lambda^{1/2}U^T$. We emphasize that, while we introduce this (potentially computationally expensive) decomposition to highlight the parallel with image deblurring, we will never have to explicitly compute it. Eq. 4 can thus be equivalently re-written as:

$$\text{Minimize}_{\pi\in\mathbb{R}^{N\times N}} ||\Phi\pi - \Phi||_F^2 + \mathbb{1}_{\pi\in\Delta_N} + 2\lambda\Big(\alpha\sum_{i,j}||K_{ij}\delta_{ij}^{(\pi)}||_2 + (1-\alpha)\sum_{i,j}||K_{ij}\delta_{ij}^{(\pi)}||_1\Big)$$

This is akin to an image deblurring problem, with π playing the role of the true image, and Φ the observed image and blurring process. Similarly to Beck and Teboulle, we thus propose to start with the associated image denoising problem, and will generalize to the original deblurring problem in a subsequent step:

$$\underset{\pi\in R^{N\times N}}{\text{Minimize}} ||\pi - \Phi||_F^2 + \mathbb{1}_{\pi\in\Delta_N} + 2\lambda\Big(\alpha\sum_{i,j}||K_{ij}\delta_{ij}^{(\pi)}||_2 + (1-\alpha)\sum_{i,j}||K_{ij}\delta_{ij}^{(\pi)}||_1\Big) \quad (5)$$

Proposition 1: The dual of Eq. 5 is given by:

$$\max_{p\in\mathcal{P},q\in\mathcal{Q}} ||\Pi_{(\Delta_N)^C}\Big(\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)\Big)||_F^2 - ||\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T||_F^2 \quad (6)$$

where we denote as Π_{Δ_N} the orthogonal projection operator on the set Δ_N and $\Pi_{\Delta_N^C} = I - \Pi_{\Delta_N}$ the projection onto its complement, and where the sets $\mathcal{P}$ and $\mathcal{Q}$ are respectively the ℓ_2- sphere and the unit cube in $\mathbb{R}^N$:

$$\mathcal{P} = \{p\in\mathbb{R}^{N\times N^2} : \forall i,j\in[1,N]^2, \quad ||p_{\cdot,ij}||_2 \leq 1\}$$
$$\text{and } \mathcal{Q} = \{q\in\mathbb{R}^{N\times N^2} : \forall i,j\in[1,N]^2, \quad ||q_{\cdot,ij}||_\infty \leq 1\}$$

The subgradients associated to this objective are Lipschitz with constant $L = 16\lambda^2\max_i||K_i^2||_2$.

Proof: We begin by observing that:

$$\max_{p\in\mathbb{R}^N:||p||_2\leq 1} p^Tx = \sqrt{\sum_{i=1}^n x_i^2} \text{ and } \max_{q\in\mathbb{R}^N:||q||_\infty\leq 1} q^Tx = ||x||_1.$$

The derivation of these observations is quite simple and given in Appendix B of the extended version of this paper. This allows Eq.4 to be re-written as:

$$\min_{\pi\in\Delta_N} ||\pi-\Phi||_F^2 + 2\lambda\max_{p\in\mathcal{P},q\in\mathcal{Q}} \text{Trace}\Big(\alpha p^T\pi\delta_K + (1-\alpha)q^T\pi\delta_K\Big)$$

The corresponding dual problem $h(p,q)$ is thus given by:

$$\max_{p\in\mathcal{P},q\in\mathcal{Q}} \min_{\pi\in\Delta_N} ||\pi-\Phi||_F^2 + 2\lambda\text{Trace}\Big(\alpha\delta_K p^T\pi + (1-\alpha)\delta_K q^T\pi\Big)$$
$$= \max_{p\in\mathcal{P},q\in\mathcal{Q}} \min_{\pi\in\Delta_N} ||\pi - \Big(\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)\Big)||_F^2 - ||\Phi - \lambda\Big(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T\Big)||_F^2$$

The inner expression here is minimized by the projection of $\Phi - \lambda\Big(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T\Big)$ onto the set Δ_N, allowing an explicit formulation of the dual as $\max_{p\in\mathcal{P},q\in\mathcal{Q}} h(p,q)$, with:

$$h(p,q) = ||\Pi_{\Delta_N^C}\Big(\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)\Big)||_F^2 - ||\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)||_F^2$$

which concludes the first part of the proof.

We now have to prove that the subgradients of the dual $h(p,q)$ are Lipschitz. Taking derivatives with respect to p and q, we can show that h is Lipschitz with constant:

$$L(h) = 16\lambda^2\max[\alpha^2,(1-\alpha)^2] \times (\max_i||K_i||_2^2)$$

We defer the proof to Appendix A ■

This proposition lays the grounds for using accelerated ascent algorithms such as FISTA on the dual: given that we have shown that the subgradients of dual is Lipschitz, FISTA ensures

to solve the objective with a convergence rate in $O(\frac{1}{k^2})$, where k denotes the number of iterations [15]. However, our setting is further complicated by the presence of the "blurring" matrix Φ. However, the update of π is complicated by the multiplication by Φ. While we have assumed here K to be positive definite and could potentially solve exactly the projection update(*), this update would in particular require the inversion of the operator $\Phi^T\Phi = K$ — a costly operation that does not transfer well in the case where K is nearly singular. Instead, we adopt the approximate strategy of Beck and Teboulle, and view it as a rough equivalent to their deblurring problem. Denoting the solution of the denoising problem by $D(\Phi, \lambda)$, the authors show the optimal solution of the deblurring problem can be obtained by iteratively solving:

$$D(Y - \frac{2}{L}\Phi^T(\Phi\pi - \Phi)), \frac{2\lambda}{L}) = D(Y - \frac{2}{L}(K\pi - K)), \frac{2\lambda}{L})$$

Empirical results (section V) validate our approach.

The FISTA updates of the dual variables are described in Algorithm 1.

Input: (fixed) variables π^0**,** K
Output: Denoising problem output
Initialization: $(p, q) = (s_0, r_0) = (\mathbf{0}_{N\times N^2}, \mathbf{0}_{N\times N^2})$
while not converged **do**
$(p_k, q_k) = \Pi_{\mathcal{P},\mathcal{Q}}\Big[r_k + \frac{2\lambda(\alpha,1-\alpha)}{L(h)}\Pi_{\Delta_N}[\pi_k - \frac{2}{L}(K\pi_k - K) - (\alpha\lambda r_k\delta_K^T + (1-\alpha)\lambda s_k\delta_K^T)]\delta_K\Big]$
Or equivalently:
$p_k = \Pi_{\mathcal{P}}\Big[r_k + \frac{\alpha}{8\lambda\max[\alpha^2,(1-\alpha)^2]\ \times(\max_i||K_i||_2^2)}\Pi_{\Delta_N}[\Phi - (\alpha\lambda r_k\delta_K^T + (1-\alpha)\lambda s_k\delta_K^T)]\delta_K\Big]$
$q_k = \Pi_{\mathcal{Q}}\Big[s_k + \frac{1-\alpha}{8\lambda\max[\alpha^2,(1-\alpha)^2]\ \times(\max_i||K_i||_2^2)}\Pi_{\Delta_N}[\Phi - (\alpha\lambda r_k\delta_K^T + (1-\alpha)\lambda s_k\delta_K^T)]\delta_K\Big]$
$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
$(r_{k+1}, s_{k+1}) = (p_k, q_k) + \frac{t_k-1}{t_{k+1}}(p_k - p_{k-1}, q_k - q_{k-1})$
$\pi_{k+1} = \Pi_{\Delta_N}[\pi^k - \frac{2}{L}(K\pi^k - K) - (\alpha\lambda r_k\delta_K^T + (1-\alpha)\lambda s_k\delta_K^T)]\delta_K\Big]$
end while

Algorithm 1: Update for π

Proposition 2: The projections operators onto the sets $\mathcal{P}, \mathcal{Q}$ are given by:

- $\Pi_{\mathcal{P}}[p] = \frac{p_k}{\max[1,||p||_2]}$
- $\Pi_{\mathcal{Q}}[q] = \frac{q_k}{\max[1,|q_k|]}$

In particular, the previous two-step procedure has the advantage of bypassing the need to compute explicitly and invert Φ. In order to compute efficiently the updates on the set of doubly stochastic matrices, we use the scalable scheme devised by [18], which is an iterative scheme with closed-form updates, detailed in Algorithm 2.

The procedure is summarized below in Algorithms 2 and 3 and a Python implementation has also been made public[1].

[1]https://github.com/donnate/HC_dev

Input: K: similarity; λ: regularization parameter
Output: π^*: optimal solution for Problem 4
Initialization;
while not converged **do**
$p_k = \Pi_{\mathcal{P}}[r_k + \frac{1}{2||K||_2^*}\Pi_{\Phi\Delta_N}[\Phi - \lambda\mathcal{L}(r_k, s_k)]]$;
$q_k = \Pi_{\mathcal{Q}}[s_k + \frac{1}{2||K||_2^*}\Pi_{\Phi\Delta_N}[\Phi - \lambda\mathcal{L}(r_k, s_k)]]$;
$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
$(r_{k+1}, s_{k+1}) = (p_k, q_k) + \frac{t_k-1}{t_{k+1}}(p_k - p_{k-1}, q_k - q_{k-1})$
end while
Return π^* such that $\Phi = \Phi\pi^* = \Pi_{\Phi\Delta_N}[\Phi - \lambda\mathcal{L}(r_k, s_k)]]$.

Algorithm 2: Updates for π.

Ymatrix to project onto Δ_N
$P^* = \arg\min_{D\in\Delta_N}||Y - D||_F^2$
Initialization: $P \leftarrow Y$*;*
while not converged **do**
$P \leftarrow P + (\frac{1}{n}I + \frac{\mathbf{1}^TP\mathbf{1}}{n^2}I - \frac{1}{n}P)\mathbf{1}\mathbf{1}^T - \frac{1}{n}11^TP$
$P \leftarrow \frac{P+|P|}{2}$
end while
Return π^* such that $\Phi\pi^* = \Pi_{\Phi\Delta_N}[\Phi - \lambda\mathcal{L}(r_k, s_k)]]$

Algorithm 3: Projection onto Δ_N

IV. Analysis of the performance

Computational cost analysis. A closer inspection of the updates in (*) reveals that only the coordinates for which δ_K is identically 0 are updated — that is, the variables p_{ij} such that $K_{ij} = 0$ remain identically zero throughout the updates. Denoting $\mathcal{E}$ as the number of non-zero entries in K, the memory required to store both p and q is thus $O(|N\mathcal{E}|)$. At each step, the algorithm thus relies on either (a) element-wise operations on matrices of size $O(|N \times \mathcal{E}|)$ or (b) matrix multiplications with cost at most $O(|\mathcal{E}N^2|)$. As such, the overall complexity and memory storage of the algorithm grows linearly with the number of edges, but quadratically with the number of nodes. We have not as yet optimized the design of an algorithm capable of efficiently storing a representation of K.

Validation of the empirical efficiency. We begin assessing the efficiency of our algorithm through a set of synthetic experiments. In each of these experiments, we generate a synthetic graph with 3-layer fractal structure: the coarsest level corresponds to an Erdos-Rényi graph on 10 "meta nodes". Each of these meta nodes can be further divided in a set of communities on 5 "super nodes", each corresponding to a dense clique on five nodes. Figure 1(D) illustrates the division process. We run our hierarchical method on the induced graph and assess the results both visually through the associated t-SNE [19] plots and quantitatively by running k-means on the induced representation.

Performance Metrics. We quantify the amount of structure recovered at each regularization level through the effective rank er [20] of the similarity between centroids: letting D_π be the

distance matrix between observations (i.e., $D_\pi[i,j] = \pi_i^T K \pi_j$) and $\{\sigma^{(D_\pi)}\}_j$ its eigenvalues, the effective rank is defined as:

$$er(\pi) = \exp\{-\sum_{k=1}^{N} \frac{\sigma_k^{(D_\pi)}}{\sum_{j=1}^{N} \sigma_j^{(D_\pi)}} \log\left(\frac{\sigma_k^{(D_\pi)}}{\sum_{j=1}^{N} \sigma_j^{(D_\pi)}}\right)\}.$$

This measures effectively the entropy of the eigenvalue distributions of the similarity between centroids, and thus progressively decreases from N to 1 as the regularization parameter increases. We also consider the distribution of the between/within cluster distances $\rho = \frac{\sum_{i,j:C_i \neq C_j} D_{ij}}{\sum_{i,j:C_i = C_j} D_{ij}}$ as an indicator of the sharpness of the clustering induced on the data centroids.

Benchmarks. Finally, we compare the performance of our method with an ADMM-based implementation [17]. Implementation details are provided in Appendix C of the extended version of this paper.

This formulation of the problem makes it particularly amenable to classification: the columns of the recovered matrix $\pi(\lambda)$ provide a representation of the centroids using the observations as dictionary – allowing to use any off-the-shelf machine learning algorithm to analyze these representations. Figure 1 shows the consistency of the recovered cluster path with hierarchical clustering: the effective rank (computed as the sum of the singular values of $\pi(\lambda)$ decreases as λ increase: for very small values of λ, the clusters consist essentially in one node –the t-Stochastic Neighborhood Embedding [19] does not exhibit any particular structure. However, as λ increases, the clusters progressively fuse, as highlighted by the progressively purer clusters. This behavior is further quantified in Table I: here, we run k-means with $k = 10$ or 50 nodes (corresponding to communities at two different scales in the generated graph) on the different representations $\hat{\pi}(\lambda)$ and report the 10-fold cross-validation accuracy. It becomes apparent from Table I that the different coarsened graph representations induced by λ correspond to different level of information: we note in particular the high accuracy obtained by small values of λ for classifying the local structures. However, these local representations fail for the detection of coarser communities. Figure 1E also shows the performance of an exact solver (CVXPY) against our proposed method, highlighting an increase in time of almost 20 fold for even very small graphs. Interestingly, the comparison of our method (using total variation penalty, rather than classical ℓ_2 smoothing losses) have empirically been shown to yield much sparser representations, thus more effectively merging centroids' coordinates.

V. Real-life Experiments

Our method was driven by its application to multi-resolution graph analysis. In this setting, the goal is typically to obtain a coarser and coarser approximation of the similarity matrix (progressively fusing the clusters together) in order to capture the underlying structure of the graph at multiple scales. With this objective in mind, we now provide a few examples of the performance of our method on three real datasets[2].

The recipes network. We begin by visualizing the coarsened coalescence path that our convex clustering algorithm achieves on a recipes network. This dataset was obtained by scraping over 40,000 recipes of 3 major US culinary websites, each belonging to one of 49 different cuisines (Chinese, American, etc). We represent of each of the 49 cuisines as a 10,000-dimensional vector in which each entry corresponds to the frequency at which a particular ingredient is used. To create a network of cuisines, we compute the cosine similarity between their ingredient frequency representation. Each edge e_{ij} in the network is thus the cosine between the ingredient frequency vectors of cuisines i and j. We then apply our algorithm to this small, yet dense recipes network.

Fig. 2 shows the t-Stochastic Neighborhood Embedding [19] representation of the cuisine's centroids at different points along the cluster fusion path. In this case, hierarchical clustering recovers three distinct clusters: Western/European (in blue), Mediterranean (in green) as well as Asian (in red). Interestingly, the differentiation between the blue and the two latter clusters occurs earlier in the cluster path — highlighting a greater similarity between Mediterranean and Asian cuisine than to their Western counterpart.

Connectomics. In this application, we wish to compare the structural connectomes of healthy individuals, undergoing a longitudinal test-retest Reliability and Dynamical Resting-State fMRI study. The data is a subset of the HNU1 cohort [21]. In particular, we focus on the structural connectomes of 5 subjects obtained over the course of 10 distinct scan sessions (three days apart from another)[3]. For each connectome, we compute its convex clustering representation for various values of the parameter λ. This yields a multiscale representation from the raw connectomes, which we then compare. The goal is to assess whether the multiscale representations obtained via Convex Clustering are more consistent and robust across subjects and scans than the ones obtained via traditional single linkage Hierarchical Clustering (HC). To compare the output of our convex clustering procedure (i.e a set of centroids) and the dendrogram obtained via single linkage HC, we compare the distance matrices between centroids that these output induce (in particular, we use the cophenetic distance [22] to convert the HC dendrogram into a distance matrix).

Fig. 3a shows, on the left side, the Kendall rank correlation between these similarity matrices for two values of λ, as well as their correlation with the cophenetic distance induced by single linkage HC. Interestingly, both HC and Convex clustering recovered multiscale representations with a strong subject effect, as highlighted by the red blocks along the diagonal: representations corresponding to different scans of the same subject are more alike than scans across subjects.

[2]The code and notebooks for the experiments are provided at the following anonymous link: https://https://goo.gl/kZdZwD.

[3]The preprocessed structural connectomes are readily available at https://neurodata.io/mri-cloud/.

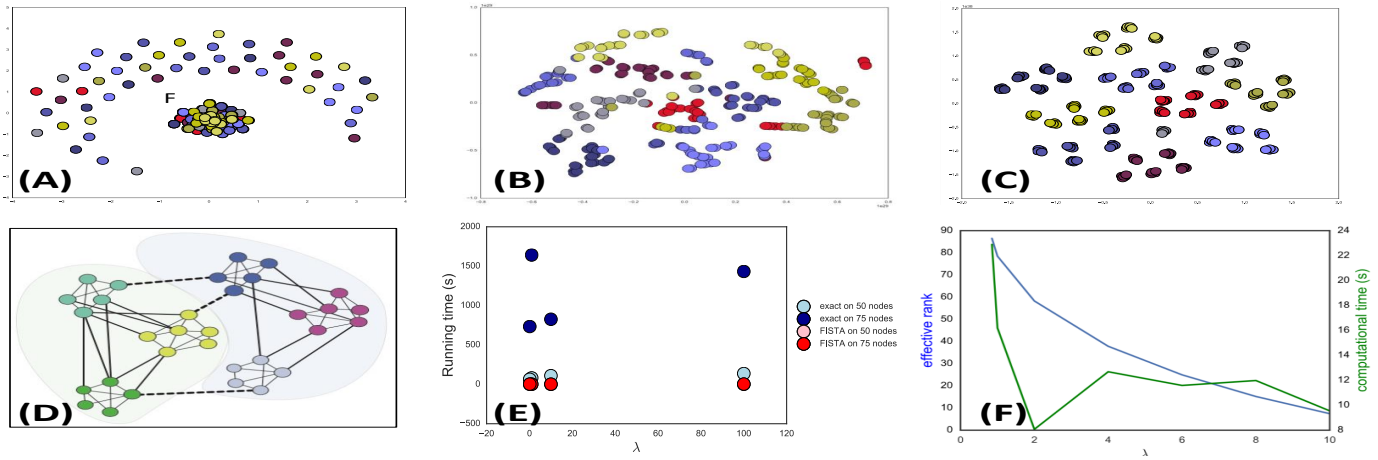


Fig. 1: Application of the convex hierarchical clustering algorithm to a synthetic graph on 250 nodes. t-SNE representation of the nodes for **(A)** $\lambda = 0.01$, **(B)** $\lambda = 1.0$ and **(C)** $\lambda = 10.0$ **(D)** Example of 3-layer hierarchy.**(E)** Comparison of the running time of CVXPY vs our method at different values of the regularization path **(F)** Effective rank of the recovered representation $\hat{r} = \sum_{i=1}^{N} \sigma_i$ and computation time.

	50 classes			10 classes		
λ	Accuracy	Homogeneity	Completeness	Accuracy	Homogeneity	Completeness
0.001	0.10	0.09	0.62	0.00	0.73	0.93
1	0.12	0.91	0.96	0.00	0.96	0.98
10	0.06	0.94	0.98	0.04	0.98	0.98
10	0.06	0.94	0.98	0.04	0.98	0.98
1000	0.096	0.86	0.93	0.00	1.00	1.00

TABLE I: Performance of running kmeans with respectively 10 and 50 classes as ground truth labels.

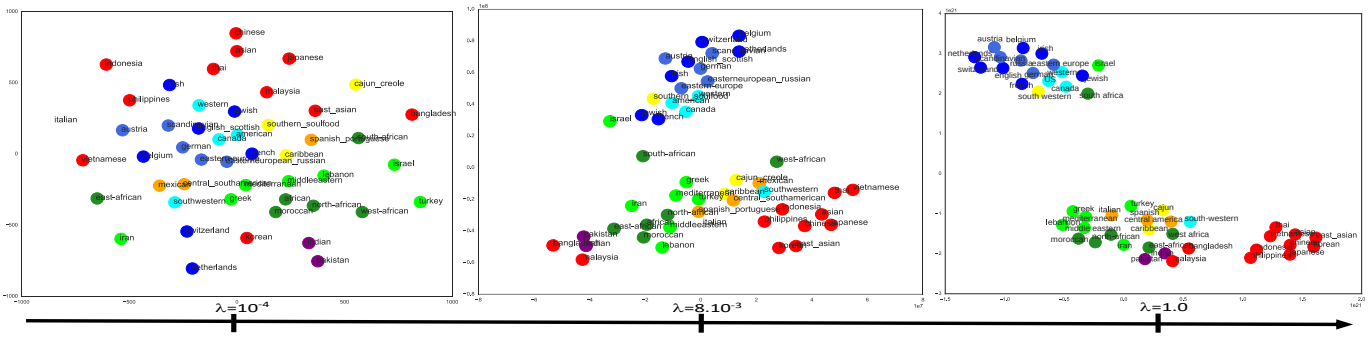
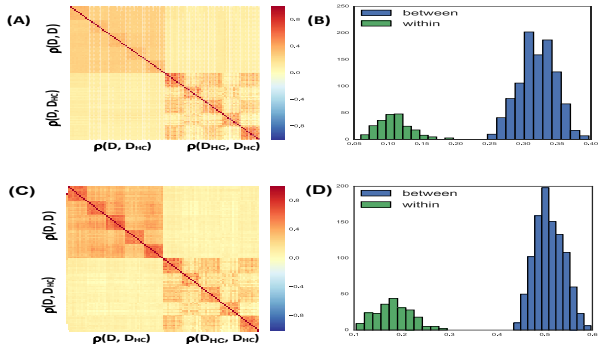


Fig. 2: t-SNE [19] plot showing the coalescence path of the cuisines' centroids.

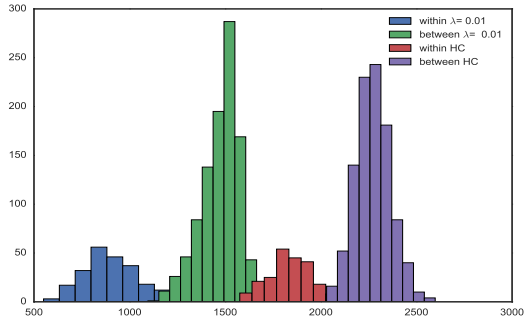
This is highlighted by the column on the right side of Fig. 3a, which shows a clear separation in the distances between scans belonging to the same subject ("within distances") and scans across different subjects ("between"). This effect fades away as the regularization increases. This is consistent with our expectation that the overall organization of the brain is globally the same across subjects, while differences between individuals are more salient at the fine-grain scale.

To quantify the relative performance of our algorithm with standard HC, we estimate the variability of its output: for each scan and each value of λ, we compute the 5 nearest-neighbor graphs that the coarsened similarity matrices induce. We then compute the distances between these 5 nearest-neighbor graphs (using the Hamming distance, that is, the raw ℓ_2-distance between adjacency matrices). We observe that the variability in these graph is smaller for convex clustering than for graphs obtained using single linkage HC: the distribution for two values of λ are plotted on Fig. 3b, and we observe that these differences are significantly inferior than to the ones of HC. This indicates that the 5-nearest neighbor graphs recovered by our convex clustering procedure induce more robust and consistent multiscale representations of the connectomes across subjects and scans.

Khan gene expression. We now demonstrate the robustness of our method by applying it to the Khan dataset [23]. This dataset consists of gene expression profiles of four types of cell tumors of childhood. In this case, we want to show that the clusters recovered by our procedure are more robust than the



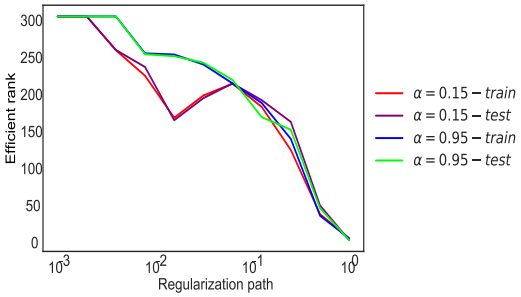
(a) Results for the connectomics study using different values of the regularization (A, B: $\lambda = 0$ and C,D: $\lambda = 0.01$). Left-column: pairwise Kendall rank correlation between coarsened brain networks induced by convex clustering $\rho(D, D)$ and HC's associated cophenetic distance $\rho(D_{HC}, D_{HC})$ as well as Kendall cross-correlation between representations $\rho(D, D_{HC})$ (matrix sketch on the left of the picture). Right column: distribution of the distances between coarsened DTI scan representations for scans belonging to the same subject ("within") and across different subjects ("between").



(b) Comparison of the "within" and "between" subject distances between coarsened representations of DWI scans for various levels of λ.

new recovered by standard hierarchical clustering, in that the multiscale representation of the similarities between genes that they capture are more reproducible: we split the dataset into a "training" (more complete) and testing dataset, and assess the similarity between the multi scale representations that we extract out of those. In this case, a set of 64 arrays and 306 gene expression values are used for training, and 25 arrays for testing. We apply hierarchical clustering on the similarity matrix induced by the data's 10 nearest-neighbor graph and assess the stability of the induced hierarchy: at each level, we aggregate the centroids in both training and testing based on their efficient rank and compute the clusters' homogeneity score using the training labels as ground truth. We compare this against standard agglomerative clustering. Interestingly, the results (displayed in the table in Fig. 4) indicate a better homogeneity of our method with respect to the greedy one for intermediary values of the regularization, indicating that the more unstable clusters of standard HC's clustering are at the intermediary levels. Fig. 5 also shows that the distances between train and test k-nn graphs (as in the connectome study) is consistently smaller than for the k-nn graphs induced using HC's cophenetic distance. This indicates greater consistency between test and train results for convex clustering.

Fig. 4: Results for the Khan Dataset



(a) Figure: Efficient rank for recovered along the regularization path, on the test and train sets for two values of α.

λ	**Effective rk** $er(\pi)$	**Homogeneity FISTA** ($\alpha = 0.95$)	**Homogeneity standard HC**
0.032	242	**0.937**	0.935
0.256	141	**0.774**	0.721
0.512	37	**0.446**	0.292
1.024	7	**0.100**	0.087

(b) Table: Homogeneity score between test and train predictions at different points of the regularization path.

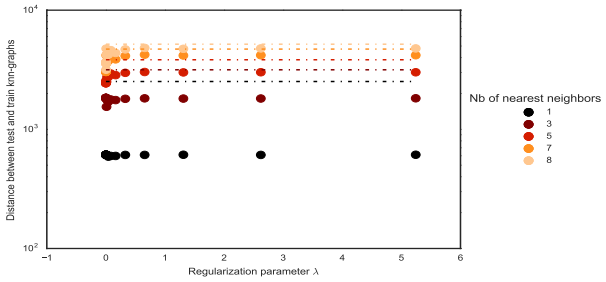


Fig. 5: Distances between test and train k-nearest neighbor graphs as the regularization λ increases (colored by k). Dashed lines indicate the HC-cophenetic baseline for the number of neighbors considered.

VI. Conclusion

In conclusion, we have created an efficient set of algorithms for solving convex hierarchical clustering in the case where the data are directly a graph or a similarity matrix. We have shown the performance of our method on both synthetic and real datasets, highlighting its ability to recover different important scales with better consistency and robustness than standard Hierarchical Clustering. One intrinsic limit to the scalability of our method lies in its requirement to store a matrix of size N^2 – an aspect that we leave for future work.

References.

[1] E. C. Chi, G. I. Allen, and R. G. Baraniuk, "Convex biclustering," *Biometrics*, vol. 73, no. 1, pp. 10–19, 2017.

[2] H. Chipman and R. Tibshirani, "Hybrid hierarchical clustering with applications to microarray data," *Biostatistics*, vol. 7, no. 2, pp. 286–301, 2005.

[3] F. Corpet, "Multiple sequence alignment with hierarchical clustering," *Nucleic acids research*, vol. 16, no. 22, pp. 10 881–10 890, 1988.

[4] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[5] E. Segal and D. Koller, "Probabilistic hierarchical clustering for biological data," in *Proceedings of the sixth annual international conference on Computational biology*. ACM, 2002, pp. 273–280.

[6] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[7] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[8] G. K. Chen, E. C. Chi, J. M. O. Ranola, and K. Lange, "Convex clustering: An attractive alternative to hierarchical clustering," *PLoS computational biology*, vol. 11, no. 5, p. e1004228, 2015.

[9] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert, "Clusterpath an algorithm for clustering using convex fusion penalties," in *28th international conference on machine learning*, 2011, p. 1.

[10] K. Pelckmans, J. De Brabanter, J. A. Suykens, and B. De Moor, "Convex clustering shrinkage," in *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*, 2005.

[11] K. M. Tan and D. Witten, "Statistical properties of convex clustering," *Electronic journal of statistics*, vol. 9, no. 2, p. 2324, 2015.

[12] C. Kelly, X.-N. Zuo, K. Gotimer, C. L. Cox, L. Lynch, D. Brock, D. Imperati, H. Garavan, J. Rotrosen, F. X. Castellanos *et al.*, "Reduced interhemispheric resting state functional connectivity in cocaine addiction," *Biological psychiatry*, vol. 69, no. 7, pp. 684–692, 2011.

[13] P. Tétreault, A. Mansour, E. Vachon-Presseau, T. J. Schnitzer, A. V. Apkarian, and M. N. Baliki, "Brain connectivity predicts placebo response across chronic pain clinical trials," *PLoS biology*, vol. 14, no. 10, p. e1002570, 2016.

[14] A. Chambolle, V. Duval, G. Peyré, and C. Poon, "Geometric properties of solutions to the total variation denoising problem," *Inverse Problems*, vol. 33, no. 1, p. 015002, 2016.

[15] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[16] ——, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[18] Y. Lu, K. Huang, and C.-L. Liu, "A fast projected fixed-point algorithm for large graph matching," *Pattern Recognition*, vol. 60, pp. 971–982, 2016.

[19] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[20] O. Roy and M. Vetterli, "The effective rank: A measure of effective dimensionality," in *Signal Processing Conference, 2007 15th European*. IEEE, 2007, pp. 606–610.

[21] X.-N. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. C. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos *et al.*, "An open science resource for establishing reliability and reproducibility in functional connectomics," *Scientific data*, vol. 1, p. 140049, 2014.

[22] P. H. Sneath, R. R. Sokal *et al.*, *Numerical taxonomy. The principles and practice of numerical classification.*, 1973.

[23] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson *et al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature medicine*, vol. 7, no. 6, p. 673, 2001.

[24] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An admm algorithm for a class of total variation regularized estimation problems," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 83–88, 2012.

APPENDIX A. LIPSCHITZ CONSTANT

We here show that the dual problem is Lipschitz with respect to each of the variables p and q. We have:

$$h(p,q) = ||\Pi_{\Delta_N^C}\Big(\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)\Big)||_F^2 - ||\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)||_F^2$$

Thus:

$$\begin{aligned}\nabla_p h(p,q) &= -\lambda\alpha\Big(2\Pi_{\Delta_N^C}[\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)] \\ &\quad - 2(\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T))\Big)\delta_K \quad (*) \\ &= 2\lambda\alpha\Pi_{\Delta_N}[\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)]\delta_K\end{aligned}$$

where $(*)$ follows from the fact that $\frac{\partial||\Pi_{\Delta_N^C}[x]||_F^2}{\partial x} = \frac{\partial||\Pi_{\Delta_N^C}[x]||_F^2}{\partial\Pi_{\Delta_N^C}[x]}\frac{\partial\Pi_{\Delta_N^C}[x]}{\partial x} = 2\Pi_{\Delta_N^C}[x]$.

Similarly:

$$\nabla_q h(p,q) = 2\lambda(1-\alpha)\Pi_{\Delta_N}[\Phi - \lambda(\alpha p\delta_K^T + (1-\alpha)q\delta_K^T)]\delta_K$$

We note that, by definition of δ_K:

$$\forall M \in \mathbb{R}^{N\times N^2}, \quad ||M\delta_K||_{ij}^2 = K_{ij}^2||M_i - M_j||_2^2$$

Hence:

$$\begin{aligned}&\forall M \in \mathbb{R}^{N\times N^2}, ||M\delta_K||_F^2 \\ &= \sum_{ij} K_{ij}^2||M_i - M_j||^2 \leq \sum_{ij} 2K_{ij}^2(||M_i||^2 + ||M_j||^2) \\ &\leq \sum_{ij}(2K_{ij}^2||M_i||^2 + 2K_{ji}^2||M_j||^2) \text{ by symmetry of } K \\ &\leq 4\sum_i ||K_{i,\cdot}||_2^2||M_i||^2 \leq 4\max_i\{||K_{i\cdot}||_2^2\} \times ||M||_F^2\end{aligned} \tag{7}$$

Hence, using the non-expensiveness property of the orthogonal projection operator, we can show that the subgradients of h are Lipschitz, since:

$$\begin{aligned}&||\nabla h(p_1,q_1) - \nabla h(p_2,q_2)||_F^2 \\ &= ||\nabla_p h(p_1,q_1) - \nabla_p h(p_2,q_2)||_F^2 \\ &+ ||\nabla_q h(p_1,q_1) - \nabla_q h(p_2,q_2)||_F^2 \\ &\leq 8\lambda^2\max[\alpha^2,(1-\alpha)^2] \times 4 \times \max_i||K_{i\cdot}||_2^2 \\ &\times ||\Pi_{\Delta_N}\Big(\Phi - \lambda(\alpha p_1\delta_K^T + (1-\alpha)q_1\delta_K^T)\Big) \\ &- \Pi_{\Delta_N}\Big(\Phi - \lambda(\alpha p_2\delta_K^T + (1-\alpha)q_2\delta_K^T)\Big)||_F^2 \\ &\leq 32\lambda^4\max[\alpha^2,(1-\alpha)^2] \times (\max_i||K_{i\cdot}||_2^2) \\ &\times ||\Big(\alpha(p_1-p_2) + (1-\alpha)(q_1-q_2)\Big)\delta_K^T||_F^2 \\ &||\nabla h(p_1,q_1) - \nabla h(p_2,q_2)||_F^2 \\ &\leq 128\lambda^4\max[\alpha^2,(1-\alpha)^2] \times (\max_i||K_{i\cdot}||_2^2) \\ &\times ||(\alpha(p_1-p_2) + (1-\alpha)(q_1-q_2))||_F^2 \\ &\leq 128\lambda^4\max[\alpha^4,(1-\alpha)^4] \times (\max_i||K_{i\cdot}||_2^2) \\ &\times 2(||p_1-p_2||_F^2 + ||q_1-q_2))||_F^2) \\ &\leq 256\lambda^4\max[\alpha^4,(1-\alpha)^4](\max_i||K_{i\cdot}||_2^2) \\ &\times ||(p_1,q_1) - (p_2,q_2)||_F^2\end{aligned} \tag{8}$$

$$\tag{9}$$

Thus:

$$||\nabla h(p_1,q_1) - \nabla h(p_2,q_2)||_F \leq 16\lambda^2 \times \max[\alpha^2,(1-\alpha)^2] \times (\max_i||K_{i\cdot}||_2)||(p_1,q_1) - (p_2,q_2)||_F$$

APPENDIX B. MORE ON THE DERIVATION OF THE FISTA UPDATES

In this appendix, we provide more in-depth descriptions of the propositions and proofs derived in this manuscript.

Derivation of the adapted convex objective problem. We begin by showing how, in the case where the kernel matrix K is assumed to be positive definite, the adaptation of convex clustering proposed in Eq. 3 naturally follows. To begin with, we remind the reader that, by Mercer's theorem, we can simply write a high-dimensional equivalent formulation $\hat{\mathcal{P}}$ of convex clustering as:

$$\hat{\mathcal{P}} = \min_{\pi_\in \Delta_N} \sum_{i=1}^{N} ||\Phi(X_i) - \sum_j \Phi(X_j)\pi_{ji}||^2 + \lambda \sum_{i,j} W_{ij} ||\sum_k \pi_{ki}\Phi(X_k) - \sum_k \pi_{k'j}\Phi(X_{k'})|| \quad (10)$$

This induces the following set of equivalents:

$$\begin{aligned}
\hat{\mathcal{P}} &\iff \text{arg min}_{\pi_\in \mathcal{S}} \sum_{i=1}^{n} \Big(||\Phi(X_i)||^2 + ||\Phi(X)\pi_{\cdot,i}||^2 \\
&\quad - 2\Phi(X_i)^T(\Phi(X)\pi_{\cdot,i}) \Big) + \lambda \sum_{i,j} W_{ij} \Big(||U_i - U_j|| \Big) \\
&\iff \text{arg min}_{\pi_\in \mathcal{S}} Tr[\pi^T \Phi(X)^T \Phi(X)\pi] \\
&\quad - 2Tr\Big(\Phi(X)^T(\Phi(X)\pi) \Big) + \lambda \sum_{i,j} W_{ij} \Big(||U_i - U_j|| \Big) \\
&\iff \text{arg min}_{\pi_\in \mathcal{S}} Tr[\pi^T \Phi(X)^T \Phi(X)\pi] \\
&\quad - 2Tr\Big(\Phi(X)^T(\Phi(X)\pi) \Big) \\
&\quad + \lambda \sum_{i,j} W_{ij} \underbrace{\Big(||\Phi(X)[\pi_{\cdot i} - \pi_{\cdot j}]|| \Big)}_{\leq L||\pi_{\cdot i} - \pi_{\cdot j}||} \\
&\implies \text{arg min}_{\pi_\in \mathcal{S}} Tr[\pi^T K \pi] \\
&\quad - 2Tr\Big[K\pi \Big] + \lambda \sum_{i,j} W_{ij} \Big(||\pi_{\cdot i} - \pi_{\cdot j}|| \Big)
\end{aligned} \quad (11)$$

where, in the last line, we have used the fact that the columns of U are in fact the coordinated of the centroids in the dictionary of the original observations – hence, penalizing the pairwise differences between euclidean representation is equivalent to penalizing the dictionary coordinates.

Proof of statement III. We here provide a brief proof of the statement in Eq. III:

$$\max_{p\in\mathbb{R}^N : ||p||_2 \leq 1} p^T x = \sqrt{\sum_{i=1}^{n} x_i^2} \text{ and } \max_{q\in\mathbb{R}^N : ||q||_\infty \leq 1} q^T x = ||x||_1$$

To see this, let us first consider the equality on p, and introduce the Lagrangian corresponding to the constraint:

$$\mathcal{L}(p,\lambda) = -p^T x + \lambda(p^T p - 1), \quad \lambda \geq 0$$

where the primal is $\min_p \max_{\lambda\in\mathbb{R}^+} \mathcal{L}(p,\lambda)$, and the dual can be written as: $\max_{\lambda\in\mathbb{R}^+} \min_p \mathcal{L}(p,\lambda)$. The latter inner minimization with respect to p is achieved for:

$$\nabla_p \mathcal{L}(p,\lambda) = -x + 2\lambda p = 0 \iff p = \frac{1}{2\lambda} x,$$

and the dual problem reduces to:

$$\max_\lambda -\frac{||x||^2}{2\lambda} - \lambda(\frac{1}{4\lambda^2}||x||^2 - 1) = \max_\lambda -\frac{||x||^2}{4\lambda} + \lambda$$

The latter is achieved for $\lambda = \frac{||x||}{2}$, and thus: $p = \frac{1}{||x||}x$. Hence, $\max_{p\in\mathbb{R}^n, p^T p \leq 1} [p^T x] = ||x||_2$, which concludes the proof. Similarly for q, it is easy to check that:

$$||x||_1 = \max_{s:\ \ s_i\in\{-1,1\}} s^T x.$$

By relaxing the constraint on s, we have: $||x||_1 = \max_{s:\ \ s_i\in[-1,1]} s^T x$, which concludes the proof.

APPENDIX C. DERIVATION OF THE ADMM UPDATES

In this appendix, we provide the derivations of the ADMM algorithm used to benchmark our FISTA-based approach in section IV.

1) Description of the algorithm: The Alternating Direction Method of Multipliers [17] is a popular algorithm for solving convex optimization problems with a large number of constraints. Indeed, with a guaranteed speed of convergence in $O(\frac{1}{k})$ iterations, this algorithm has become the work-horse of convex problems with coupling constraints. However, contrary to the parameter-free implementation of convex clustering with FISTA, ADMM requires the selection of the parameter ρ, whose choice has been shown to considerably affect the speed of convergence [17], [24]. In what follows, in order to simplify the notations, denoting as e the vectors of the Cartesian basis, we introduce the pairwise-difference matrix $\delta \in \mathbb{R}^{N\times N^2} : \delta_{k,ij} = \mathbf{e}_{k,i} - \mathbf{e}_{k,j}$ Introducing the variables $Z_{ij} = \pi_i - \pi_j$ and dual variables u_{ij}, the ADMM-augmented Lagrangian can be written as:

$$\begin{aligned}
\min_{\pi\in\Delta_N} \frac{1}{2}\text{Tr}(\pi^T K\pi - 2K^T\pi) + \frac{\rho}{2}\sum_{ij} ||\pi\delta + u_{ij} - Z_{ij}||^2 \\
+ \lambda \sum_{ij} K_{ij}(\alpha||Z_{ij}||_1 + 1 - \alpha)||Z_{ij}||_2) \\
\text{s. t. } \pi \in \Delta_N, \quad \forall i,j, \quad \pi_i - \pi_j = \pi\delta_{ij} = Z_{ij}
\end{aligned} \quad (12)$$

The full algorithm and derivation of the updates are provided in the following subsection and the whole procedure is summarized in Alg. 4, and the corresponding updates are derived in the following paragraphs.

Input: Similarity matrix K, regularization parameter λ
Output: Optimal solution $\pi^{(\lambda)}$
Initialization; $Z, U = \mathbf{0} \in \mathbb{R}^{N\times N^2}, t = 0$
while not converged **do**
  $\pi^{t+1} = \text{Update}_\pi(Z^t, U^t)$ {explicited in Algorithm 5}
  $Z^{t+1} \leftarrow \text{SoftThreshold}_{\frac{\alpha\lambda||Z^t||}{\rho||Z^t||+(1-\alpha)\lambda}} \Big[\frac{\pi^{t+1}\delta + U^t}{(1+\frac{(1-\alpha)\lambda}{\rho||Z^t||})} \Big]$
  $U^{t+1} \leftarrow U^t + (\pi^{t+1}\delta - Z^{t+1})$
  $t \leftarrow t+1$
end while
Return $\pi^* = \Pi_{\Delta_N}(\pi)$

Algorithm 4: ADMM

2) Updates: **Updating** π. The objective in Eq. 12 reads as quadratic linear optimization problem in π. The updates in X are unfortunately not computable in closed form. However, provided that we have access to an efficient projection on the set of doubly stochastic matrices Δ_N, we can solve the corresponding update using an accelerated Proximal Descent algorithm. In particular, the gradients with respect to π are given by:

$$\nabla_\pi F(\pi, Z, u) = K\pi - K + \rho(\pi\delta + U - Z)\delta^T$$

We note that these gradients are in particular Lipschitz (with respect to π, all other variables being fixed):

$$\nabla_\pi F(\pi_1, Z, u) - \nabla_\pi F(\pi_2, Z, u) = K(\pi_1 - \pi_2) + \rho(\pi_1 - \pi_2)\delta\delta^T$$

We also have:

$$\begin{aligned}
\delta\delta^T &= \Big(\sum_{ij}(e_{ki}-e_{kj})(e_{li}-e_{lj})\Big)_{kl} = 2\Big(\sum_{j}(e_{lk}-e_{lj})\Big)_{kl} \\
&= 2\Big(ne_{lk} - e_{ll}\Big) = 2nI - 2\mathbf{1}\mathbf{1}^T \\
\implies ||\delta\delta^T||_F^2 &= \text{Trace}[4n^2 I - 8n\mathbf{1}\mathbf{1}^T + 4n\mathbf{1}\mathbf{1}^T] \\
\implies ||\delta\delta^T||_F^2 &\leq 4n^3 \\
\implies ||\nabla_\pi F(\pi_1, Z, u) - \nabla_\pi F(\pi_2, Z, u)||_F & \\
&\leq \sqrt{||K||_F^2 + \rho^2||\delta\delta^T||^2}||\pi_1 - \pi_2||_F \\
&\leq \sqrt{||K||_F^2 + 4\rho^2 n^3}||\pi_1 - \pi_2||_F
\end{aligned} \tag{13}$$

Hence, to solve for π, we can use an accelerated proximal method (such as FISTA), with constant step-size $L = \sqrt{||K||_F^2 + 4\rho^2 n^3}$. Since, the projection onto Δ_N does not have a closed form solution either, the literature typically resorts to fixed-point algorithms such as the one proposed in [18], yielding the procedure described in Algorithm 5.

Algorithm: Updates for $\pi^t(\lambda)$
Input:(fixed) variables K, Z and U initialization: $t_k = 1$;
while not converged **do**
  $\pi^k = \Pi_{\text{1-round}}^{\Delta_N}(Y_k - \frac{1}{\sqrt{||K||_F^2 + 4\rho^2 n^3}}\nabla_\pi F(Y^k, Z^t, U^t))$
  (Projection $\Pi_{\text{1-round}}^{\Delta_N}$ described in Alg. 6)
  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
  $Y_{k+1} \leftarrow \pi_k + \frac{t_k - 1}{t_{k+1}}(\pi_k - \pi_{k-1})$
end while

Algorithm 5: Updates for π.

Objective: Projection on Δ_N: $\Pi_{\text{1-round}}^{\Delta_N}$
Input:square matrix Y
Initialization: $P = Y$
while not converged **do**
  $P \leftarrow P + (\frac{1}{n}I + \frac{\mathbf{1}^T P \mathbf{1}}{n^2}I - \frac{1}{n}P)\mathbf{1}\mathbf{1}^T - \frac{1}{n}1 1^T P$
  $P \leftarrow \frac{P+|P|}{2}$
end while
Return P

Algorithm 6: One-round of the fixed point iterative algorithm ($\Pi_{\text{1-round}}^{\Delta_N}$) for projecting unto Δ_N as proposed in [18].

Updating Z. The updates in terms of Z are more explicit, since Z is simply the solution to a denoising problem with an elastic net penalty.

Taking the gradient with respect to Z yields:

$$\rho(Z - X\delta - U) + \lambda\alpha\,\mathbf{sign}\,Z + \lambda(1-\alpha)\frac{Z}{||Z||} = 0$$

We solve the later through a set of sequential updates:

$$\begin{aligned}
&(1 + \frac{(1-\alpha)\lambda}{\rho||Z^{t-1}||})Z^t = X\delta + U + \frac{\lambda\alpha}{\rho}\,\mathbf{sign}\,Z^t \\
\implies &Z^t = \text{SoftThreshold}_{\frac{\alpha\lambda||Z^{t-1}||}{\rho||Z^{t-1}||+(1-\alpha)\lambda}}\Big[\frac{1}{(1+\frac{(1-\alpha)\lambda}{\rho||Z^{t-1}||})}(X\delta + U)\Big].
\end{aligned}$$