Recovering Trees with Convex Clustering[*]

Eric C. Chi[†] and Stefan Steinerberger[‡]

Abstract. Hierarchical clustering is a fundamental unsupervised learning task, whose aim is to organize a collection of points into a tree of nested clusters. Convex clustering has been proposed recently as a new way to construct tree organizations of data that are more robust to perturbations in the input data than standard hierarchical clustering algorithms. In this paper, we present conditions that guarantee when the convex clustering solution path recovers a tree and also make explicit how affinity parameters in the convex clustering formulation modulate the structure of the recovered tree. The proof of our main result relies on establishing a novel property of point clouds in a Hilbert space, which is of potentially independent interest.

Key words. Convex optimization, Fused lasso, Hierarchical clustering, Penalized regression, Sparsity

AMS subject classifications. 46C05, 49J99, 52C35

1. Introduction.

Hierarchical clustering is a fundamental unsupervised learning task, whose aim is to organize a collection of points into a tree of nested clusters. To reinforce the idea that we seek a collection of nested clusters, we will often also refer to clusters as folders in this paper.

As an illustration, Figure 1 shows a collection of points in $\mathbb{R}^2$, labeled 1 to 18, that we seek to organize. Based on the Euclidean distances between the points, an intuitive organization is the following hierarchy of nested clusters. At the finest and first level of clustering, we partition the set $\{1, \ldots, 18\}$ into five subsets or folders:

$$
\begin{aligned}
F_{1,1} &= \{1,2,3,4,5\}, \quad F_{1,2} = \{6,7,8\}, \quad F_{1,3} = \{9,10,11,12,13\}, \\
F_{1,4} &= \{14,15,16\}, \text{ and } F_{1,5} = \{17,18\}.
\end{aligned}
$$

At the second level of clustering, we merge the folders from the first level into a partition of two folders: $F_{2,1} = F_{1,1} \cup F_{1,2}$ and $F_{2,2} = F_{1,3} \cup F_{1,4} \cup F_{1,5}$.

Finally, at the third level of clustering, we merge the folders from the second level into a single folder: $F_{3,1} = F_{2,1} \cup F_{2,2}$. Figure 2 illustrates the described tree organization. Since each level of the tree consists of a partition of the data points, we refer to such hierarchical organizations as “partition trees.”

There are many existing algorithms for automatically constructing partition trees, but perhaps the most often used algorithms in practice are collectively known as agglomerative hierarchical clustering methods [18, 21, 23, 30, 47]. Given a collection of points in $\mathbb{R}^p$, agglomerative hierarchical clustering methods recursively merge the points which are closest together

[*] Submitted to the editors DATE.

Funding: ECC was partially supported by the National Science Foundation under Grant DMS-1752692. SS was partially supported by the National Science Foundation under Grant DMS-1763179 and the Alfred P. Sloan Foundation.

[†] Department of Statistics, North Carolina State University, Raleigh, NC (eric_chi@ncsu.edu).

[‡] Department of Mathematics, Yale University, New Haven, CT (stefan.steinerberger@yale.edu).

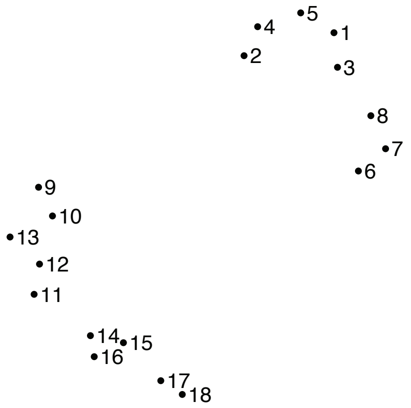


Figure 1: Eighteen points in $\mathbb{R}^2$ to organize.

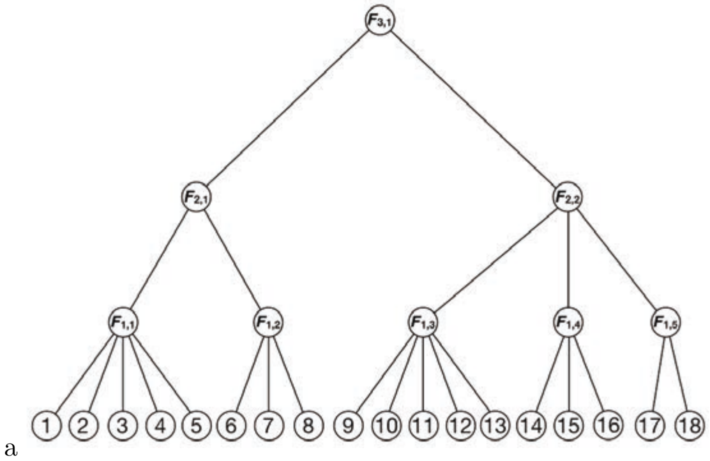
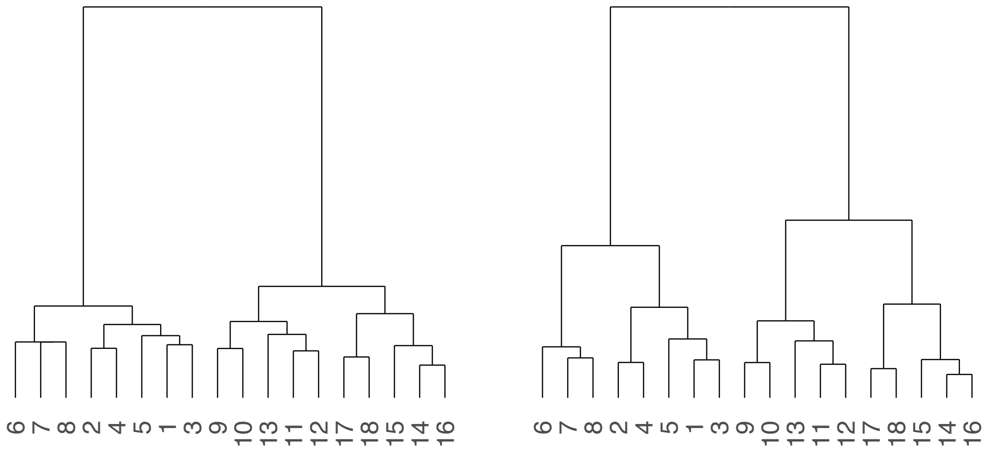


Figure 2: Partition Tree.

until all points are joined. Different choices in the definition of closeness lead to the different variants. Figure 3 shows two trees computed by two variants of the agglomerative hierarchical clustering. For each tree, the eighteen points reside in the "leaves" which are organized into a hierarchy of nested clusters that captures an increasingly coarser grouping structure as one progresses from the leaves to the root of the tree. The branch lengths in the tree quantify the similarity between pairs of points, or clusters at higher levels. We see that both trees recover binary partition trees that are similar to the ideal partition tree shown in Figure 2.



(a) Single-linkage tree

(b) Average-linkage tree

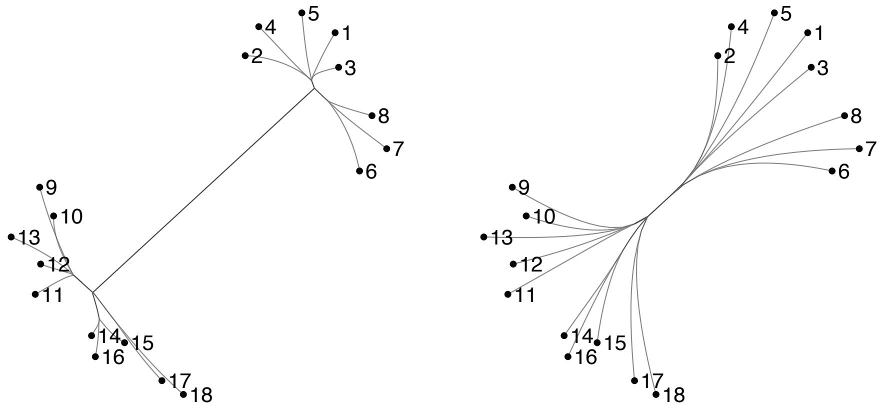
Figure 3: Hierarchical clustering of data in Figure 1 under two different agglomeration methods.

1.1. Convex Hierarchical Clustering?. Although agglomerative hierarchical methods are widely used in practice, the greedy manner in which trees are constructed often results in an unstable mapping between input data and output tree. Indeed, agglomerative hierarchical clustering methods have been shown to be highly sensitive to perturbations in the input data, namely the resulting output trees can vary drastically with the addition of a little Gaussian noise to the data [10].

One promising alternative strategy for constructing trees stably relies on formulating the clustering problem as a continuous optimization problem. Following up on the initial proposal by [33], several recent works have shown that solving a sequence of convex optimization problems can recover tree organizations [9, 12, 19, 25, 32, 41]. Given n points $x_1, \dots, x_n$ in $\mathbb{R}^p$, we seek cluster centers (centroids) u_i in $\mathbb{R}^p$ attached to point x_i that minimize the convex criterion

$$E_\gamma(u) = \frac{1}{2}\sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{i<j} w_{ij}\|u_i - u_j\|, \tag{1.1}$$

where γ is a nonnegative tuning parameter, w_{ij} is a nonnegative affinity that quantifies the similarity between x_i and x_j, and u is the vector in $\mathbb{R}^{np}$ obtained by stacking the vectors $u_1, \dots, u_n$ on top of each other. For now, we assume all norms are Euclidean norms; we will later consider arbitrary norms. The sum of squares data-fidelity term in (1.1) quantifies how well the centroids u_i approximate the data x_i, while the sum of norms regularization term penalizes the differences between pairs of centroids u_i and u_j. To expand on the latter, the regularization term is a composition of the group lasso [51] and the fused lasso [44] and incen-



(a) Gaussian Kernel Affinities

(b) Unit Affinities

Figure 4: Solution paths of convex clustering using different affinities w_{ij}.

tivizes sparsity in the pairwise differences of centroid pairs. Overall, $E_\gamma(u)$ can be interpreted as the energy of a configuration of centroids u for a given relative weighting γ between data-fidelity and model complexity as quantified by the regularization term. We next elaborate how $u(\gamma)$ varies as the tuning parameter γ varies.

Because the objective function $E_\gamma(u)$ in (1.1) is strongly convex, for each value of γ it possesses a unique minimizer $u(\gamma)$, whose n subvectors in $\mathbb{R}^p$ we denote by $u_i(\gamma)$. The tuning parameter γ trades off the relative emphasis between data fit and differences between pairs of centroids. When $\gamma = 0$, the minimum is attained when $u_i = x_i$, namely when each point occupies a unique cluster. As γ increases, the regularization term encourages cluster centroids to fuse together. Two points x_i and x_j with $u_i = u_j$ are said to belong to the same cluster. For sufficiently large γ, the u_i fuse into a single cluster, namely $u_i = \overline{x}$, where $\overline{x}$ is the average of the data x_i [12, 42]. Moreover, the unique global minimizer $u(\gamma)$ is a continuous function of the tuning parameter γ [10]; we refer to the continuous paths $u_i(\gamma)$, traced out from each x_i to $\overline{x}$ as γ varies, collectively as the solution path. Thus, by computing $u_i(\gamma)$ for a sequence of γ over an appropriately sampled range of values, we hope to recover a partition tree.

Figure 4 plots the u_i as a function of γ for two different sets of affinities w_{ij}. We will discuss the differences in the recovered trees shortly, but for now we point out that computing $u(\gamma)$ for a range of γ indeed appears to recover trees that bear similarity to the desired partition tree in Figure 2. Moreover, the $u_i(\gamma)$ are 1-Lipschitz functions of the data x_i [11]. Consequently, small perturbations to the input data x_i, are guaranteed to *not* result in disproportionately large variations in the output $u_i(\gamma)$.

At this point, the solution path of convex clustering appears to stably recover partition trees as desired. Nonetheless, questions remain as to whether convex clustering is a form of convex hierarchical clustering. Specifically, (i) when is the solution path guaranteed to produce a tree, and (ii) how do the affinities modulate the branch formation in the recovered tree?

Hocking et al. provide a partial answer to the first question [19]. They prove that if unit affinities are used, namely $w_{ij} = 1$ for all i and j, and if 1-norms are used in the regularization term in (1.1), then the solution path must be a tree. On the other hand, in the same paper, they also provide an example, using the Euclidean norm in the regularization term, where the solution path can fail to be a tree. Specifically, as the tuning parameter γ increases, it is possible for centroids to initially fuse and then "unfuse" before eventually fusing again. We provide an example of this phenomenon in Appendix A.

The differences in the two recovered trees shown in Figure 4 motivate the second question. Figure 4a shows the solution path when using Gaussian kernel affinities, namely for all i and j

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma}\right),$$

where σ is a positive scale parameter. Gaussian kernel affinities have been empirically shown to provide more aggressive fusion of folders closer to the leaves, and consequently more informative, hierarchical clustering results [10, 12, 19]. Figure 4b shows the solution path when using unit affinities. We see that Gaussian kernel affinities can generate a solution path that recovers the partition tree in Figure 2, while unit affinities can generate a solution path that recovers a less "nested" approximation to the partition tree in Figure 2. The same sets of points and folders are getting shrunk together in Figure 4a and Figure 4b, but less aggressively in the latter as γ increases. In Appendix B, we provide an additional real data example highlighting how different the recovered trees can be under the two sets of affinities. Our main result will complement these empirical observations with a theoretical argument for why certain data-driven affinities, including but not limited to Gaussian kernel affinities, should be preferred over others.

1.2. Contributions.

In this paper, we answer the open questions of (i) why the solution path of convex clustering can recover a tree and (ii) how affinities can be chosen to guarantee recovery of a given partition tree on the data. We first answer these questions in the case when Euclidean norms are employed in (1.1) and then later describe how our results can be extended to more general data-fidelity terms and arbitrary norms in the regularization term.

We clarify how the theoretical contributions in this paper differ from existing theoretical results in the convex clustering literature. Radchenko and Mukherjee in [34] present a population model for the convex clustering procedure and provide an analysis of the asymptotic properties of the sample convex clustering procedure. We note that their analysis is specific to using 1-norms in the regularization term, while we consider first the Euclidean norm before generalizing to arbitrary ones. Zhu et al. in [54] provide conditions under which two true underlying clusters can be identified by solving the convex clustering problem with appropriately chosen affinities. Similarly, She [39] and Sharpnack et al. [38] present results when the convex clustering solution can consistently recover groupings. Others present finite sample prediction error bounds for recovery of a latent set of clusters [42, 46].

Our contributions differ from these prior works in two ways. First, we provide conditions on the affinities that ensure that the solution path reconstructs an *entire* hierarchical partition tree and clarify how these affinities can be explicitly tuned to recover a specific target tree. With the exception of the work by Radchenko and Mukherjee in [34], all of the other works present theoretical guarantees for recovering a *single* partition level rather than a nested hierarchy of partitions. Second, in contrast to all of the previous work, we do not make any distributional assumptions on the data. Instead, we focus in this paper on understanding the behavior of the solution path as a function of the affinities used in the regularization term. By understanding this dependency, we gain insight into why a commonly used data-driven affinities choice, namely the Gaussian kernel, works so well in practice.

1.3. Outline.

The rest of this paper proceeds as follows. In Section 2, we define structures needed to construct affinities that will enable us to recover a desired partition tree and once equipped with the necessary building blocks, give an overview of our main result. In Section 3, we introduce a geometric lemma that is key to proving our main result. In Section 4, we give proofs of the geometric lemma and our main theorem. In Section 5, we show how our main result can be generalized to other data-fidelity terms and regularization term norms. In Section 7, we conclude with a discussion on our results within the broader context of penalized regression methods for clustering.

2. Setup and Overview of Main Result.

Our main result shows that if the affinities w_{ij} arise from an underlying partition tree, then that tree can be reconstructed from the solution path of the convex clustering problem. To proceed, we will need a formal definition of a partition tree and then a judicious assignment of weights to the edges in the tree graph corresponding to the partition tree.

2.1. Partition Tree.

Let $\Omega = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ be an arbitrary collection of points and let $[n]$ denote the set of indices $\{1, \ldots, n\}$. Following the notation and language employed in [2] and [29, 28], we say that $\mathcal{T}$ is a partition tree on the collection of points Ω consisting of $\mathcal{P}_0, \ldots, \mathcal{P}_L$ partitions of Ω if it has the following properties:

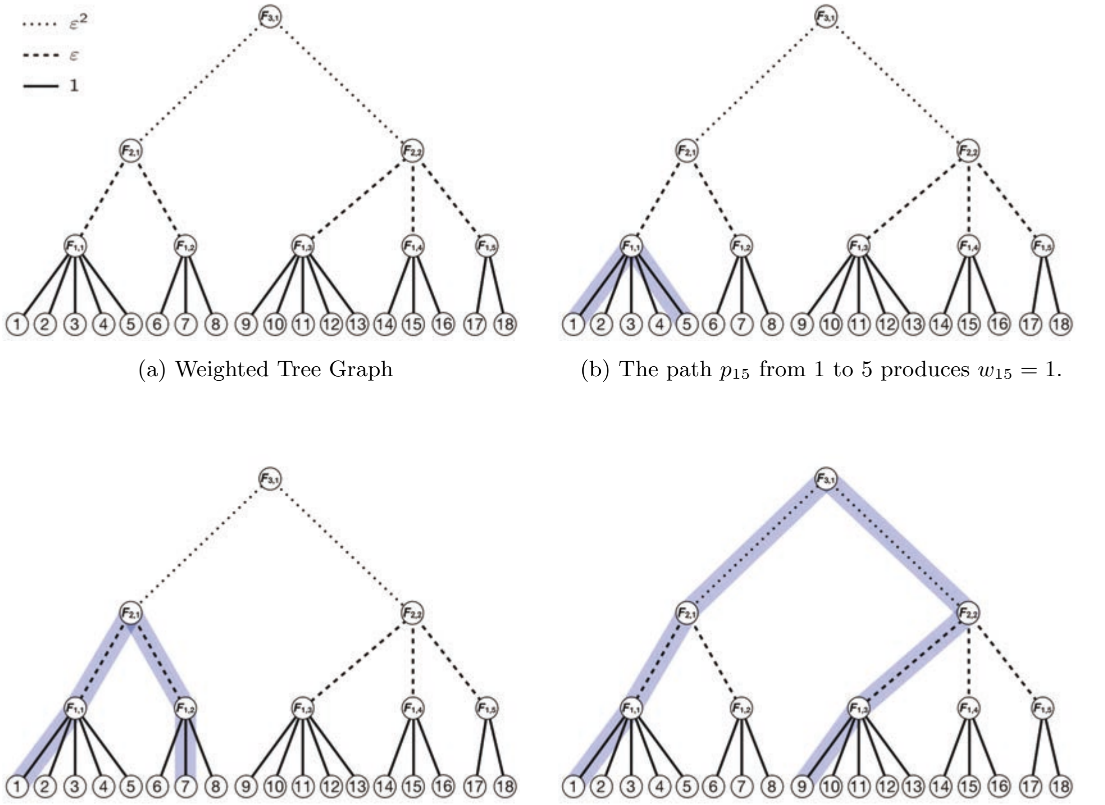
1. The partition $\mathcal{P}_l = \{F_{l,1}, \ldots, F_{l,n_l}\}$ at level l consists of n_l disjoint non-empty subsets of indices in $\{1, \ldots, n\}$, termed folders and denoted by $F_{l,i}, i \in [n_l]$.
2. The finest partition $\mathcal{P}_0$ contains $n_0 = n$ singleton "leaf" folders, namely $F_{0,i} = \{i\}$.
3. The coarsest partition $\mathcal{P}_L$ contains a single "root" folder, namely $F_{L,1} = [n]$.
4. Partitions are nested; if $F \in \mathcal{P}_l$, then $F \subseteq F'$ for some $F' \in \mathcal{P}_{l+1}$, namely each folder at level $l-1$ is a subset of a folder from level l. Note that we allow for $F = F'$.

A partition tree $\mathcal{T}$ on Ω can be seen as the collection of all folders at all levels, namely $\mathcal{T} = \{F_{l,i} : 0 \leq l \leq L, i \in [n_l]\}$.

2.2. Weighted Tree Graph.

We next assign every folder $F_{l,i} \in \mathcal{T}$ to a node and draw an edge between nested folders in adjacent levels. Thus, if $F \in \mathcal{P}_l, F' \in \mathcal{P}_{l+1}$, and $F \subset F'$, then we draw an edge (F, F') between F and F'. If we let $\mathcal{E}$ denote the set of all edges between nested folders in adjacent levels, then the resulting graph $\mathcal{G} = (\mathcal{E}, \mathcal{T})$ is a tree.

We next assign weights on the edges in $\mathcal{E}$ as follows. Let $\varepsilon > 0$ be a fixed parameter, whose value we will elaborate on shortly. Edges between level 0 folders and level 1 folders receive a weight of 1. Edges between level 1 folders and level 2 folders receive a weight of ε. Edges between level 2 folders and level 3 folders receive a weight of ε^2 and so on. Thus, edges between level l folders and level $l+1$ folders receive a weight of ε^l. Figure 5a shows the weighted tree graph $\mathcal{G}$ derived from the partition tree given in Figure 2.



(a) Weighted Tree Graph

(b) The path p_{15} from 1 to 5 produces $w_{15} = 1$.

(c) The path p_{17} from 1 to 7 produces $w_{17} = \varepsilon$.

(d) The path p_{19} from 1 to 9 produces $w_{19} = \varepsilon^2$.

Figure 5: Weighted Tree: Edges that are solid lines have weight 1. Edges that are dashed lines have weight ε. Edges that are dotted lines have weight ε^2.

We are finally ready to construct w_{ij} from the weighted tree graph. Let $F_{0,i}$ and $F_{0,j}$ be leaf nodes in the graph $\mathcal{G}$ and let p_{ij} be the sequence of edges in $\mathcal{E}$ that form the path between $F_{0,i}$ and $F_{0,j}$. Then we set w_{ij} to be the smallest weight of edges contained in p_{ij}. In other words, w_{ij} is the smallest edge weight one sees in traveling from i to j. Figure 5b shows that the path p_{15} from 1 to 5 in the weighted graph $\mathcal{G}$ leads to the affinity assignment $w_{15} = 1$. Figure 5c and Figure 5d show additional examples of how affinities are derived from the edge weights in $\mathcal{G}$.

2.3. Main Result.

We now state our main result.

Theorem 2.1. *There exists $\varepsilon_0 > 0$, depending on the data and the tree structure (which we assume defines the w_{ij} as outlined above in Section 2.2), so that for all $\varepsilon \in (0, \varepsilon_0)$ the solution* *path*

$$u(\gamma) = \underset{u_1,\ldots,u_n}{\arg\min} \sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{i,j=1}^{n} w_{ij}\|u_i - u_j\|,$$

as parametrized by $\gamma \in (0, \gamma_0)$ traces out exactly the partition tree structure underlying the affinities w_{ij} before collapsing into a point for some large, but finite, γ_0.

Informally speaking, this means that as γ increases, elements from the same folder collapse into a single point, these folders (now single points) move themselves (or rather, the fused points move in a coordinated manner) and then collapse again in a way predicted by the tree (i.e. folders sharing a parent folder collapse). This evolution continues on until all points have collapsed into a single point (which happens for a finite value γ_0). We have no precise bound on the times γ at which these collapses happen but by making ε_0 sufficiently small, there is an arbitrary long time between stages of collapsing. The proof of Theorem 2.1 also gives a bound on γ_0 as a byproduct.

Remarks Several additional remarks are in order.

1. At first blush, it appears that the data x_i plays no role in the recovered partition tree as the affinities w_{ij} dictate the trajectory of the solution path. In practice, however, one would *never* use w_{ij} that did not depend on the data. We study the convex clustering solution path separate of any particular data-driven choice of the affinities, but intuitively the affinity w_{ij} should be inversely proportional to the distance between x_i and x_j. Theorem 2.1 further clarifies a sufficient condition on how *rapidly* (i.e. geometrically fast) the affinity w_{ij} should decrease as the distance between x_i and x_j increases for all pairs of data points, to ensure the solution path is a tree. To further clarify the importance of using w_{ij} that respect the geometry of the data, we give an example of a solution path that is *not* a tree as a consequence of using w_{ij} that do not respect the geometry of the data in Appendix A.
2. The affinities do not need to have exactly the structure described in Section 2.2. A more precise statement would be that there exists an ε_0 such that whenever we associate weight $\varepsilon_1 \in (0, \varepsilon_0)$ to the first level, then there exists an ε (depending on everything and $\varepsilon_0, \varepsilon_1$) such that if we associate weight $\varepsilon_2 \in (0, \varepsilon)$ to the second level there exists an ε_3 (depending on everything and $\varepsilon_0, \varepsilon_1, \varepsilon_2$ etc.). Simply put, it suffices to have a sufficiently clear separation of scales encoded in the affinities.
   Indeed, Figure 6 shows the Gaussian kernel affinities w_{1j} between x_1 and the remaining x_j for $j = 2, \ldots, 18$ from the example in Figure 1. We observe clear separation of scales encoded in the Gaussian kernel affinities that align with the partition tree and corresponding weighted graph $\mathcal{G}$ in Figure 5a. Similar plots of the set of affinities associated with each data point reveal alignment with the partition tree and corresponding weighted graph $\mathcal{G}$. The key quality of the Gaussian kernel should be readily apparent, namely the Gaussian kernel naturally encodes, in a data-driven way, a geometric decay in weights that is sufficient to reconstruct a partition tree embedded in Euclidean space. We emphasize, however, that there is nothing special about the Gaussian kernel, and its rapid decay in weights is not even necessary. Any data-driven

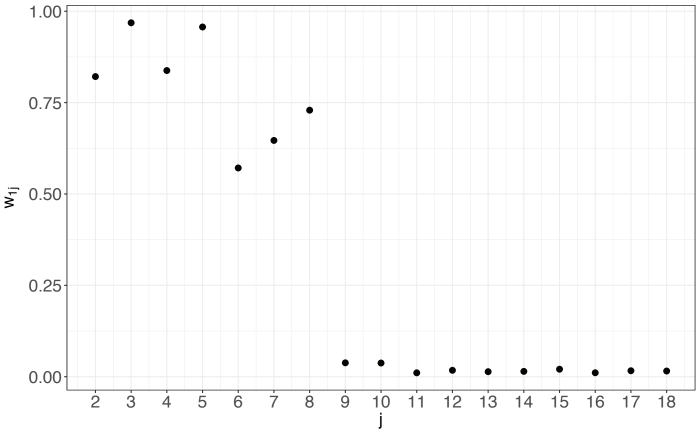


Figure 6: Gaussian kernel affinities w_{1j} between x_1 and the other x_j from the example in Figure 1.

affinities possessing a sufficient separation of scales will produce similar trees.

3. The result is completely independent of where the $\{x_1, \ldots, x_n\} \in \mathbb{R}^p$ are located in space. Their location, however, affects the critical scale ε_0.
4. The statement guarantees that points u_i fuse together with respect to the folder structure before moving to fuse with other points and their respective folder structure, however, we do not have clear control over whether they intersect (in the sense of two u_i, u_j belonging to different folders occupying the same point in space for some value of γ) in between or not. Generically, this will not happen but, for a non-generic set of x_i, it is possible to arrange for the u_i to indeed intersect, then move apart again before finally fusing for a larger value of γ. This is a consequence of our lack of conditions on the position of the points x_i. If the x_i are located in space in a way that actually reflects the tree structure, then they will fuse upon intersecting for the first time.

3. A Geometric Lemma.

We establish a geometric Lemma that is of intrinsic interest: it states that for any set of distinct points $\{u_1, \ldots, u_n\} \in \mathbb{R}^p$, one of these points u (indeed, one on the boundary of the convex hull of all the points) has the property that for a suitable "viewing direction" $v \in \mathbb{R}^p$ most points are clearly visible when standing in the point u and looking towards the viewing direction (in the sense of having a large inner product). We now phrase this more precisely below. Recall that the convex hull of a set S, denoted by $\operatorname{conv} S$ is the smallest convex set containing the set S.

Lemma 3.1. *For every set $S = \{u_1, \ldots, u_n\} \subset \mathbb{R}^p$ of $n \geq 3$ distinct points, there exists*

$$u \in S \cap \partial \operatorname{conv} S \quad \textit{and} \quad v \in \mathbb{R}^p \quad \textit{satisfying } \|v\| = 1$$

such that

$$\frac{1}{n}\sum_{\substack{i=1\\u_i\neq u}}^{n}\left\langle \frac{u_i-u}{\|u_i-u\|}, v\right\rangle \geq \frac{1}{2}. \tag{3.1}$$

The statement can be summarized as follows: for a suitable point $u \in S \cap \partial \operatorname{conv} S$, if we map the direction to all other points onto the unit sphere $\mathbb{S}^p$, then convexity implies that there is a great circle on $\mathbb{S}^p$ such that all these directions are on one side of the great circle or on it. This can be interpreted as the dualization of the fact that there is a supporting hyperplane touching the boundary of the convex hull in such a way that all of $\operatorname{conv} S$ is on one side. The statement claims the existence of a boundary point u such that the average projection point is bounded away from that great circle by a universal constant.

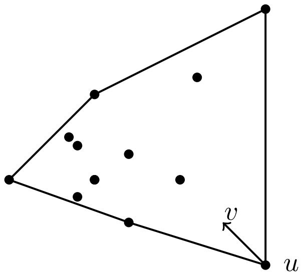


Figure 7: A set of points in $\mathbb{R}^2$: there exists a point u on the boundary of the convex hull and a direction v such that the average inner product of $(u_i-u)/\|u_i-u\|$ and v is bounded away from 0 by a universal constant.

We will use Lemma 3.1 to study the regularization term in (1.1), namely the functional

$$J(u) = \sum_{i,j=1}^{m} \|u_i - u_j\| \qquad \text{for a given set of distinct points } \{u_1, u_2, \ldots, u_m\} \subset \mathbb{R}^p.$$

The functional J is clearly minimized for any collection of u_i that are all identical. Consequently, any collection of distinct u_i represents a suboptimal configuration of centroids and therefore admits a descent direction that leads to a decrease in energy. The power of Lemma 3.1 is that it identifies a direction that guarantees a large amount of decrease in J. To see this, we write down the directional derivative of J explicitly.

The directional derivative of moving u_j in direction $v \in \mathbb{R}^p$, normalized to $\|v\| = 1$ is computed as

$$\begin{aligned}
\left\langle \frac{\partial J}{\partial u_j}, v \right\rangle &= \lim_{t\to 0}\frac{1}{t}\sum_{i\neq j}\|u_i-(u_j+tv)\| - \|u_i-u_j\| \\
&= \lim_{t\to 0}\frac{1}{t}\sum_{i\neq j}\sqrt{\langle u_i-(u_j+tv), u_i-(u_j+tv)\rangle} - \|u_i-u_j\| \\
&= \sum_{i\neq j}\lim_{t\to 0}\frac{1}{t}\left(\sqrt{\|u_i-u_j\|^2 - 2t\,\langle u_i-u_j, v\rangle + t^2} - \|u_j-u_i\|\right) \\
&= -\sum_{i\neq j}\left\langle \frac{u_i-u_j}{\|u_i-u_j\|}, v\right\rangle.
\end{aligned} \tag{3.2}$$

The expression for the directional derivative given in (3.2), in conjunction with Lemma 3.1, shows that it is always possible to find one point such that moving it δ in a certain direction decreases the entire functional by at least $(n/2)\delta$. The existence of a direction of guaranteed minimum decrease in J will be essential in proving Theorem 2.1.

The following variant of Lemma 3.1 will also be useful in applications.

Lemma 3.2. *For every set $S = \{u_1, \dots, u_n\} \subset \mathbb{R}^p$ of $n \geq 3$ points such that not all of them are in the same place, there exists*

$$u \in S \cap \partial\,\mathrm{conv}\,S \quad \textit{and} \quad v \in \mathbb{R}^p \quad \textit{satisfying } \|v\| = 1$$

such that

$$\frac{1}{n}\sum_{\substack{i=1\\ u_i\neq u}}^{n}\left\langle \frac{u_i-u}{\|u_i-u\|}, v\right\rangle \geq \frac{1}{4}. \tag{3.3}$$

Before proceeding to proofs of the geometric lemmata and main result, we also note the following consequence because of its intrinsic interest. We give a proof of Corollary 3.3 in Appendix C.

Corollary 3.3. *Let $S = \{u_1, \dots, u_n\} \subset \mathbb{R}^p$ be a set of distinct points. Then there exist at least $n/6$ points $u \in S$ having the property that for some $\|v\| = 1$*

$$\frac{1}{n}\sum_{\substack{i=1\\ u_i\neq u}}^{n}\left\langle \frac{u_i-u}{\|u_i-u\|}, v\right\rangle \geq \frac{1}{4}.$$

This simple statement has non-trivial implications: Lemma 3.1 may seem like these vantage points from which to observe the entirety of the set without having too many small inner products are rare. To the contrary, Corollary 3.3 declares that the property is surprisingly common and enjoyed by a universal fraction of all points. While we do not use Corollary 3.3 in the proof of our main result, we believe this result to be of substantial independent interest since it can be interpreted as a basic statement (with universal constants) in a general Hilbert space. It could be of interest to further pursue this line of investigation.

4. Proofs.

We now prove Lemma 3.1, Lemma 3.2, and Theorem 2.1.

4.1. Geometric Lemmata.

Proof of Lemma 3.1. Let $S = \{u_1, u_2, \ldots, u_n\}$. Select an arbitrary $u \in \partial S \cap \operatorname{conv} S$, and let $y \in S$ be a point in the set furthest from u (there may be more than one such point), formally

$$\|u - y\| = \max_{1 \leq i \leq n} \|u - u_i\| \tag{4.1}$$

It is easy to see that y resides on the boundary of the convex hull; y is in fact an extreme point. We now show that u, equipped with the viewing direction vector $v_1 = (y-u)/\|y-u\|$, or y, equipped with the viewing direction vector $v_2 = -v_1$, has the desired property. We first show that for every $u_i \notin \{u, y\}$

$$\left\langle \frac{u_i - u}{\|u_i - u\|}, v_1 \right\rangle + \left\langle \frac{u_i - y}{\|u_i - y\|}, v_2 \right\rangle \geq 1. \tag{4.2}$$

Since we are only dealing with three points u, y, and u_i, all angles are determined by the corresponding triangle, which we can assume without loss of generality to reside in $\mathbb{R}^2$. Moreover, the invariance under dilation, translation and rotation enables us to assume that $u = (0,0)$ and $y = (1,0)$. If we write $u_i = (a,b)$, then the expression on the left hand side of (4.2) simplifies to

$$\left\langle \frac{u_i - u}{\|u_i - u\|}, v_1 \right\rangle + \left\langle \frac{u_i - y}{\|u_i - y\|}, v_2 \right\rangle = \frac{a}{\sqrt{a^2+b^2}} + \frac{1-a}{\sqrt{(1-a)^2+b^2}}, \tag{4.3}$$

and the condition on the distances $\|u - u_i\|$ and $\|y - u_i\|$ required by (4.1) implies that

$$\max\left\{a^2+b^2, (1-a)^2+b^2\right\} \leq 1. \tag{4.4}$$

Minimizing the expression in (4.3) subject to the constraint in (4.4) gives us the desired inequality in (4.2); almost equality is attained for u_i very close to either u or y and equality is attained for $(a,b) = (1/2, \sqrt{3}/2)$. We then sum the left and right hand sides of (4.2) over $i = 1, \ldots, n$ to arrive at the inequality

$$\sum_{\substack{i=1 \\ u_i \neq u}}^{n} \left\langle \frac{u_i - u}{\|u_i - u\|}, v_1 \right\rangle + \sum_{\substack{i=1 \\ u_i \neq y}}^{n} \left\langle \frac{u_i - y}{\|u_i - y\|}, v_2 \right\rangle \geq n, \tag{4.5}$$

which follows from realizing that each of the sums contains one term that is equal to 1 and that the remaining sum runs over all $u_i \notin \{u, y\}$ yielding at least a total of $n-2$. Thus at least one of the two terms is size $n/2$ and we obtain the desired result. ■

Proof of Lemma 3.2. Let $S = \{u_1, u_2, \ldots, u_n\}$ be a set of points not all of which are in the same place. Then the diameter of the set is not 0 and there exist two points, that we call w.l.o.g. u_1 and u_2 such that $\|u_1 - u_2\| = \operatorname{diam}(S)$. Let us suppose the number of points that are co-located with u_1 is n_1, the number of points that are co-located with u_2 is n_2 and the number of points everywhere else is n_3. Clearly,

$$n_1 + n_2 + n_3 = n.$$

The main idea is now to derive two independent lower bounds. One of them will be tighter when n_1+n_2 is large (compared to n) and one will be tighter when n_1+n_2 is small (compared to n). We can then always apply the stronger of the two bounds and that will end up in resulting a lower bound of $n/4$ regardless of what the values of n_1 and n_2 are.

Bound 1. We could pick u to be u_1 and its viewing direction vector $v_1 = (u_2-u_1)/\|u_2-u_1\|$ or, conversely, the point u_2 and the vector $v_2 = (u_2-u_1)/\|u_2-u_1\|$ to be u and v respectively. We note that, since we chose the points to be of maximal distance, all arising inner products are nonnegative. Therefore

$$\sum_{\substack{i=1\\ u_i\neq u_1}}^{n} \left\langle \frac{u_i-u_1}{\|u_i-u_1\|}, v_1 \right\rangle \geq n_2$$

and

$$\sum_{\substack{i=1\\ u_i\neq u_2}}^{n} \left\langle \frac{u_i-u_2}{\|u_i-u_2\|}, v_2 \right\rangle \geq n_1.$$

Altogether, there is a pair of vectors u and v that achieves a sum of inner products of at least $\max\{n_1, n_2\}$, which is a good bound when either of those two numbers is large (but true in all cases). On the other hand, since we are only considering that small subset of points, the bounds naturally become quite loose when n_1+n_2 is small.

Bound 2. On the other hand, we can remove all the points co-located with either u_1 or u_2 except for one in each set, leaving us with $n-n_1-n_2+2$ points. We can now apply the previous argument which guarantees the existence of a vector u and a vector v with

$$\sum_{\substack{i=1\\ u_i\neq u}}^{n} \left\langle \frac{u_i-u}{\|u_i-u\|}, v \right\rangle \geq \frac{n-n_1-n_2+2}{2}.$$

We see that this bound is quite good when n_1 and n_2 are small, in particular we recover the original bound for distinct points whenever $n_1 = n_2 = 1$.

Conclusion. Having both bounds at our disposal, we can always guarantee the existence of a pair u and v such that the lower bound is at least

$$\max\left\{\frac{n-n_1-n_2+2}{2}, n_1, n_2\right\} \geq \frac{1}{2}\left(\frac{n-n_1-n_2+2}{2} + \frac{n_1+n_2}{2}\right) \geq \frac{n}{4}$$

where the last line makes use of the inequality

$$\max\{x,y,z\} \geq \frac{x}{2}+\frac{y}{4}+\frac{z}{4} \qquad \text{for all } x,y,z \geq 0$$

since the maximum has to exceed every weighted average. ■

4.2. Main Theorem.

Outline: The proof is based on the self-similarity of the statement. We essentially show that points at the lowest level fuse in the right way with points in the same leaves (those who have mutual affinity 1). Once they are fused, we show that they stay fused for all subsequent values of γ. The newly emerging problem turns out to be exactly of the same type as the original one: we re-interpret fused points as single points with a mutual interaction now at scale $\sim \varepsilon$ (which becomes the dominant scale since points with $w_{ij} = 1$ are already fused). This makes crucial use of the geometry of the 1-norm. At every step, the arguments will go through provided ε is sufficiently small (but positive) and since the tree is of finite height, the result follows. To be more precise, the argument will proceed as follows.

1. We assume that the x_i are fixed and that the u_i are solutions of the minimization problem
$$\inf_{u_1,\dots,u_n} \left[\sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{i,j=1}^{n} w_{ij} \|u_i - u_j\| \right].$$
Plugging in an example shows that the minimal energy is uniformly bounded in γ. This has some basic implications: the u_i cannot be too far away from the x_i and not too far away from each other.
2. We then study a subset of points $\{x_1, \dots, x_n\}$ contained in a leaf of the tree. This means that their mutual affinity satisfies $w_{ij} = 1$ and the affinity between any of these points to any other point not in the leaf of the partition is at most ε.
3. We then focus exclusively on these point sets and prove that for γ sufficiently large, these sets are necessarily fused in a point. This is where Lemma 3.2 will be applied.
4. Once we establish that for γ sufficiently large, the point sets in the leaf are fused into exactly one point as desired, the full statement essentially follows by induction since these fused points interact exactly as individual points used to do; having common parents in the tree becomes the next-level analogue of being associated to the same leaf. The result then follows.

Proof. We introduce the energy of the minimal energy configuration for $\gamma > 0$ as

$$E(\gamma) = \inf_u E_\gamma(u) = \inf_u \left[\sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{i<j} w_{ij} \|u_i - u_j\| \right].$$

By setting $u_1 = u_2 = \cdots = u_n$ and putting these points in the center of mass of $\{x_1, \dots, x_n\}$, we observe that this energy is uniformly bounded for all γ

$$E_{\text{sup}} = \sup_{\gamma > 0} \, E(\gamma) \leq \sum_{i=1}^{n} \left\| x_i - \frac{1}{n} \sum_{i=1}^{n} x_i \right\|^2 < \infty.$$

We decompose the energy functional $E(\gamma)$ as

$$E(\gamma) = E_1(\gamma) + E_2(\gamma), \tag{4.6}$$

where

$$E_1(\gamma) = \sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{(i,j) \in \mathcal{E}_1} \|u_i - u_j\|,$$

where $\mathcal{E}_1 = \{(i,j) : w_{ij} = 1\}$ and

$$E_2(\gamma) = \gamma \sum_{(i,j)\in\mathcal{E}_2} w_{ij}\|u_i - u_j\|,$$

where $\mathcal{E}_2 = \{(i,j) : w_{ij} \leq \varepsilon < 1\}$. The decomposition (4.6) makes explicit that, for ε sufficiently small, the functional $E_2(\gamma)$ can be interpreted as an error term, while the dominant dynamics are determined by $E_1(\gamma)$. We now claim that for γ sufficiently large (where sufficiently large depends on everything except the parameter ε) any subset of the points u_i whose mutual affinities are 1 (i.e. all the members of one of the leaves in the tree) are fused in a point. The argument can be made quantitative and we will give an explicit bound on γ that will be sufficient.

We will now ensure that we can assume that all points are distinct. The energy E is a continuous functional. This means that we can move any potentially clumped points apart by accepting an arbitrarily small increase of energy; the remainder of the argument works as follows: if points happen to be clumped together – but not in exactly one point but in several – then we may move all of them an arbitrarily small bit. We can accept an arbitrarily small increase of energy as long as we are able to then deduce a definite decrease in energy afterwards (that will depend on the diameter of the u_i); this contradiction shows that the clumping has to occur in exactly one point. The next step in the argument is dynamical: we compute the effect of moving one of the points an infinitesimal amount (this is already using the assumption that all u_i are distinct). Reusing the computation in (3.2), we see that

$$\left\langle \frac{\partial E}{\partial u_j}, v \right\rangle = 2\left\langle u_j - x_j, v\right\rangle - \gamma \sum_{\substack{i=1\\ i\neq j,(i,j)\in\mathcal{E}_1}}^{n} \left\langle \frac{u_i-u_j}{\|u_i-u_j\|}, v\right\rangle + \left\langle \frac{\partial}{\partial u_j}\gamma \sum_{(i,j)\in\mathcal{E}_2} w_{ij}\|u_i-u_j\|, v\right\rangle. \tag{4.7}$$

The first term on the right hand side of (4.7) is bounded above by

$$2\left|\langle u_j - x_j, v\rangle\right| \leq 2\|x_j - u_j\| \leq 2\sqrt{E_{\sup}}, \tag{4.8}$$

and the third term on the right hand side of (4.7) is bounded above by

$$\left\| \frac{\partial}{\partial u_j}\gamma \sum_{(i,j)\in\mathcal{E}_2} w_{ij}\|u_i-u_j\| \right\| = \gamma \left\| \sum_{i:(i,j)\in\mathcal{E}_2, i\neq j} w_{ij}\frac{u_i-u_j}{\|u_i-u_j\|} \right\| \leq \gamma\varepsilon n. \tag{4.9}$$

Lemma 3.2 guarantees that there exists u_j for which the second term on the right hand side of (4.7) is

$$-\gamma \sum_{\substack{i=1\\ i\neq j,(i,j)\in\mathcal{E}_1}}^{n} \left\langle \frac{u_i-u_j}{\|u_i-u_j\|}, v\right\rangle \leq -\frac{\gamma}{4}\#\left\{1 \leq i \leq n : (i,j)\in\mathcal{E}_1\right\}.$$

The proof of Lemma 3.1 is even stronger and guarantees that if $\|u_i - u_j\| = \operatorname{diam}\{u_1, \dots, u_n\}$, then either u_i or u_j has the desired property and can be moved in a suitable direction v. Plugging the u_j and v from Lemma 3.1 into both sides of (4.7) and applying inequalities (4.8) and (4.9), we arrive at the following inequality.

$$\left\langle \frac{\partial E}{\partial u_j}, v \right\rangle \leq D(\gamma) = 2\sqrt{E_{\text{sup}}} + \gamma \varepsilon n - \frac{\gamma}{4} \# \{1 \leq i \leq n : (i,j) \in \mathcal{E}_1\}. \tag{4.10}$$

A crucial observation is that for

$$\varepsilon < \frac{1}{4n} \# \{1 \leq i \leq n : (i,j) \in \mathcal{E}_1\}$$

we can conclude the existence of γ sufficiently large (depending on all the other parameters) so that $D(\gamma) < 0$. This, however, means the point configuration $\{u_1, \dots, u_n\}$ cannot be a minimizer of the functional since we found a point u_j and a direction v such that moving u_j into direction v decreases the functional. This is a contradiction unless we are somehow forbidden to apply Lemma 3.2: the only assumption in Lemma 3.2 is that not all points u_i are in the same place. Thus we see that, for γ sufficiently large, all points in $\mathcal{E}_1$ are fused. A simple computation shows that these points have to be fused for all

$$\gamma \geq \frac{4\sqrt{E_{\text{sup}}}}{\# \{1 \leq i \leq n : (i,j) \in \mathcal{E}_1\} - 4\varepsilon n}.$$

(This lower bound is not sharp; in practice, points will already be fused for smaller values of γ.) A careful inspection of the proof shows that we do not require $w_{ij} = 1$ for points in the same partition: it suffices if $1 \leq w_{ij} \leq c$ for some constant c if subsequent parameter choices of γ are allowed to depend on that. The full statement now follows by induction: points in leaves become a single point, their parent structure determines the next collection of leaves and the product of their affinities determines the new affinities. Since there are only finitely many levels to the tree, the process eventually terminates. ∎

5. Extensions of the Main Theorem.

The proof of Theorem 2.1 relies on rather elementary analysis and consequently is quite flexible. Indeed, the proof can be immediately extended to more general notions of energy of the type

$$E_\gamma(u) = \phi(x_1, \dots, x_n, u_1, \dots, u_n) + \gamma \sum_{i<j} w_{ij} \|u_i - u_j\|_X,$$

where X is an arbitrary norm on $\mathbb{R}^p$ and ϕ is assumed to satisfy the following properties:

1. The function $\phi : \mathbb{R}^{p \times n} \to \mathbb{R}_{\geq 0}$ is differentiable and enforces some degree of data-fidelity and compactness. More precisely, at one extreme ϕ should be minimized when $u_i = x_i$, for example ϕ is nonnegative for all u and $\phi(x_1, \dots, x_n, x_1, \dots, x_n) = 0$. At the other extreme, ϕ should diverge whenever $\|u\|$ diverges. We want ϕ to have the property of ensuring that minimizing the energy implies that all u_i are trapped in a universal convex set (determined by the x_i but independent of γ). This amounts to a type of growth condition on ϕ and many of the functions one would canonically choose will have that property.

2. For all u for which

$$\phi(x_1,\dots,x_n,u_1,\dots,u_n)+\gamma\sum_{i,j=1}^{n} w_{ij}\left\|u_i-u_j\right\|_X \le \inf_{x\in\mathbb{R}^p}\phi(x_1,\dots,x_n,x,\dots,x),$$

we have

$$\left\|\frac{\partial}{\partial u_i}\phi(x_1,\dots,x_n,u_1,\dots,u_n)\right\| \le c$$

where c only depends on γ and $\{x_1.\dots,x_n\}$.

The argument proceeds in exactly the same way and makes crucial use of the fact that any two norms in a finite-dimensional Euclidean space are equivalent up to constants, namely

$$c_5\|x\|_{\ell^2} \le \|x\|_X \le c_6\|x\|_{\ell^2}.$$

Since constants can always be absorbed in γ, this reduces to our case, namely $X = \ell^2$.

Proof. (Sketch of the argument) Setting all $u_i = x$ and minimizing over x implies that the energy is uniformly bounded in γ (with a bound depending only on $\{x_1,\dots,x_n\}$). Since the norm X is comparable to the Euclidean norm, this implies that any minimizing configuration $\{u_1,\dots,u_n\}$ has to have a bounded diameter (with a bound depending only on $\{x_1,\dots,x_n\}$). Then, for γ sufficiently large (depending on c), Lemma 3.1 implies a direction of decay and thus points are eventually fused. We leave the precise details to the interested reader. ■

We close this section by noting that the generality of our result opens the door to intriguing applications. For example, one potential application of our extension is to construct partition trees of regression coefficients in clustered regression [5, 22, 39, 48]. We leave these investigations as future work.

6. Convex Clustering in High-dimensional Spaces.

We now briefly provide some practical guidance in using convex clustering in high-dimensional spaces. Beyer et al. showed in [4] that over a broad class of data distributions, as the ambient dimensional increases, distances from a point to its nearest neighbors become indistinguishable from distances to its farthest neighbors. Thus, at first glance, it is unclear whether tree organizations can be recovered from high-dimensional data using convex clustering, a method in which distance metrics play a central role. Fortunately, many high-dimensional data sets encountered in engineering and science can be approximated reliably by a lower dimensional representation or embedding. In some cases, high-dimensional data consist of many features that contain little to no information about the clustering structure and should be dropped. In this case, one may consider computing a sparse convex clustering solution path [46]. In other cases, where there are more nuanced relationships among most or even all the features, we may turn to nonlinear dimension reduction methods. Indeed, manifold learning [3, 13, 15, 43, 35] has proven to be effective as a nonlinear dimension reduction technique in many scientific domains where very high-dimensional measurements are recorded such as in bioinformatics [17, 20, 27, 50] and neuroscience [7, 6, 8, 36, 40, 45]. Upon some reflection, this is not surprising, as these studies collect high-dimensional data that are generated from natural processes that are subject to physical constraints and are thus intrinsically low-dimensional.

In light of these observations, we recommend the following simple strategy. First, embed high-dimensional data into a low-dimensional space, and then compute a convex clustering solution path using the low-dimensional representation of the data. This strategy is especially natural if one uses diffusion maps, since the diffusion distance between two points in high-dimensions can be approximated by the Euclidean distance in the lower dimensional diffusion maps space [13]. Once points are embedded in the diffusion maps space, one can use Gaussian kernel affinities and compute the convex clustering solution path using the Euclidean norm in the regularization term.

7. Discussion.

In this paper, we answered the question of when the convex clustering solution path can recover a tree. The key to ensuring the recovery of a well nested partition tree is the use of affinities that encourage the fusions within a folder before fusions with higher level folders and so on as the tuning parameter γ increases. By choosing the edge weight parameter ε sufficiently small, different folders have very little incentive to interact, and the optimization problem is essentially decoupled. As γ increases, the same procedure repeats itself.

We end with a discussion on the relationship between convex and non-convex formulations of penalized regression based clustering. Although we focus in this paper on the ability of convex clustering to recover a potentially deep hierarchy of nested folders, our result also sheds light on a gap in theory and practice that convex clustering's performance can be significantly improved when using non-uniform data-driven affinities when seeking a shallow or single level of nested folders. In practice, Gaussian kernel affinities have been observed to work well, but these affinity choices have until now lacked formal justification.

Indeed, non-uniform affinities provide the link between convex clustering and other penalized regression-based clustering methods that use folded concave penalties. It is well known that 1-norm penalties lead to parameter estimates that are shrunk towards zero. This shrinkage toward zero is the price for simultaneously estimating the support, or locations of the nonzero entries, in a sparse vector as well the values of the nonzero entries. In the context of convex clustering, the centroid estimates u_i are shrunk towards the grand mean $\overline{x}$. Consequently, others have proposed employing a folded concave penalty instead of a norm in the regularization terms [31, 26, 49]. Folded concave penalties induce milder shrinkage in exchange for giving up convexity in the optimization problem, which means that iterative algorithms can typically at best converge only to a KKT point.

Suppose we were to employ a folded concave penalty, such as the smoothly clipped absolute deviation [16] or minimax concave penalty [53], and seek to minimize the following alternative objective to (1.1)

$$\tilde{E}_\gamma(u) = \frac{1}{2}\sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{i<j} \varphi\left(\|u_i - u_j\|\right), \tag{7.1}$$

where each $\varphi : [0,\infty) \mapsto [0,\infty)$ has the following properties: (i) φ is concave and differentiable on $(0,\infty)$, (ii) φ vanishes at the origin, and (iii) the directional derivative of φ exists and is positive at the origin.

Since φ is concave and differentiable, for all positive z and $\tilde{z}$

$$\varphi(z) \leq \varphi(\tilde{z}) + \varphi'(\tilde{z})(z - \tilde{z}).$$

In other words, the first order Taylor expansion of a differentiable concave function φ provides a tight global upper bound at the expansion point $\tilde{z}$. Thus, we can construct a function that is a tight upper bound of the function $\tilde{E}_\gamma(u)$

$$g_\gamma(u \mid \tilde{u}) = \frac{1}{2}\sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{i<j} w_{ij}\|u_i - u_j\| + c_7, \tag{7.2}$$

where c_7 is a constant that does not depend on u, and w_{ij} are affinities that depend on $\tilde{u}$, namely

$$w_{ij} = \varphi'\left(\|\tilde{u}_i - \tilde{u}_j\|\right).$$

Note that if we take $\tilde{u}_i$ to be the data x_i, and $\varphi(z)$ to be the following variation on the error function

$$\varphi(z) = \int_0^z e^{-\frac{\alpha^2}{\sigma}} d\alpha,$$

then the bounding function given in (7.2) coincides, up to an irrelevant shift and scaling, with the convex clustering objective using Gaussian kernel affinities.

The function $g_\gamma(u \mid \tilde{u})$ is said to majorize the function $\tilde{E}_\gamma(u)$ at the point $\tilde{u}$ [24] and minimizing it corresponds to performing one step of the local linear-approximation algorithm [37, 55], which is a special case of the majorization-minimization (MM) algorithm [24]. Thus, we can see that employing Gaussian kernel affinities corresponds to taking one step of a local linear-approximation algorithm applied to a penalized regression based clustering with an appropriately chosen folded concave penalty.

In practice, variants that employ folded concave penalties take multiple steps of the local linear approximation. So at the kth step,

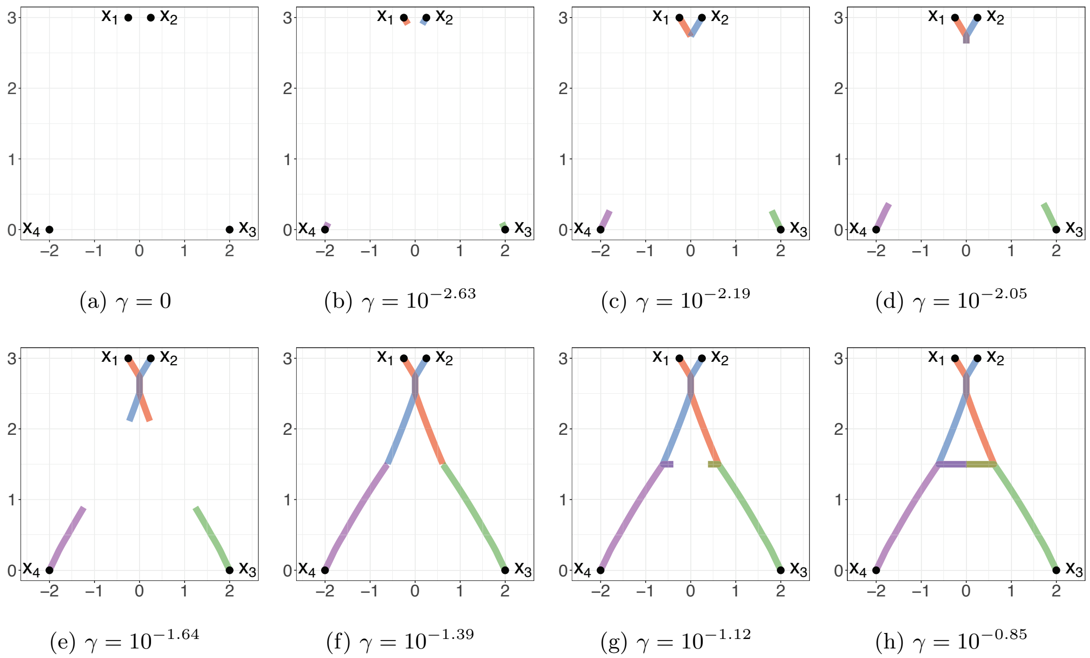
$$u^{(k)} = \arg\min_u \frac{1}{2}\sum_{i=1}^{n} \|x_i - u_i\|^2 + \gamma \sum_{i<j} \varphi'\left(\|u_i^{(k-1)} - u_j^{(k-1)}\|\right)\|u_i - u_j\|.$$

As affinities represent a data-driven way to approximate the partition tree, one can see that employing folded concave penalties corresponds to implicitly recomputing the affinities, which corresponds to refining our estimate of the partition tree based on the data.

In light of this current work, this last observation raises two interesting questions: (i) what partition tree is being recovered by a solution path of a penalized regression-based clustering method that uses a folded concave penalty and (ii) when is the recovered partition tree substantially different than the tree corresponding to a one-step local linear approximation? We leave these questions to future work.

Appendix A. Example of Non-Tree Solution Path.

We recreate a configuration of points in $\mathbb{R}^2$ and affinities similar to those used in [19], which yield a solution path that is not a tree. Consider the following four points, $x_1 = (-0.25, 3), x_2 = (0.25, 3), x_3 = (2, 0)$, and $x_4 = (-2, 0)$, and employ affinities $w_{12} = 9, w_{13} = w_{24} = 30$, and $w_{ij} = 1$ for all remaining i and j pairs. Figure 8 shows snapshots of the evolution of the solution paths for $u_1(\gamma)$ (red), $u_2(\gamma)$ (blue), $u_3(\gamma)$ (green), and $u_4(\gamma)$ (purple) as γ increases. We see that $u_1(\gamma) = u_2(\gamma)$ for a continuous range of γ greater than $10^{-2.05}$ and strictly less than $10^{-1.64}$ (Figure 8d and Figure 8e) but that $u_1(\gamma) \neq u_2(\gamma)$ for a continuous range of γ greater than $10^{-1.64}$ and less than $10^{-0.85}$ (Figure 8e, Figure 8f, and Figure 8g).



(a) $\gamma = 0$

(b) $\gamma = 10^{-2.63}$

(c) $\gamma = 10^{-2.19}$

(d) $\gamma = 10^{-2.05}$

(e) $\gamma = 10^{-1.64}$

(f) $\gamma = 10^{-1.39}$

(g) $\gamma = 10^{-1.12}$

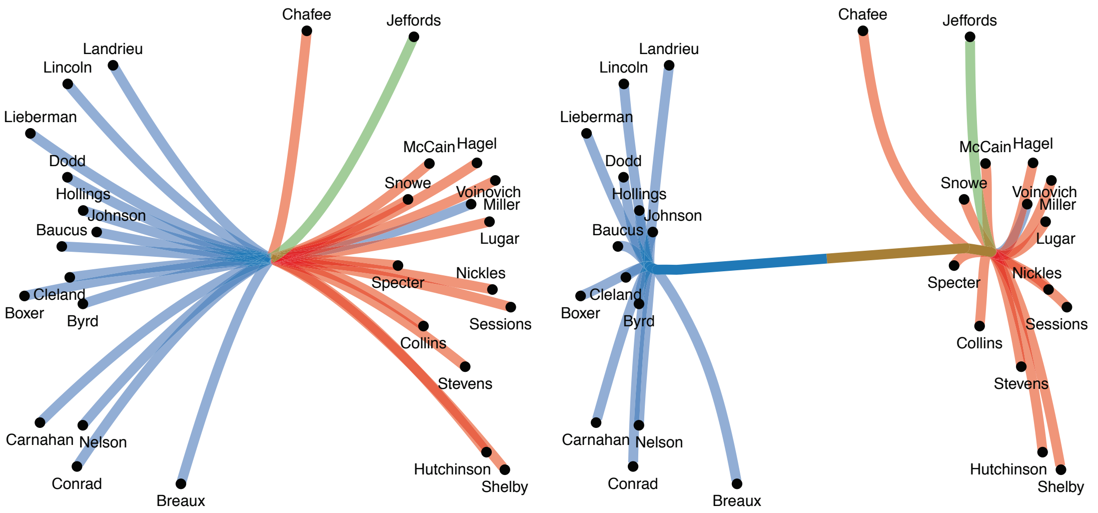
(h) $\gamma = 10^{-0.85}$

Figure 8: Snapshots of the solution path as the parameter γ increases.

We emphasize that in order to generate this degenerate solution path, we needed to use affinities that *do not* reflect the geometry of the data. The largest affinities, w_{13} and w_{24}, are between the two pairs of points that are furthest apart from each other.

Appendix B. Comparison of Unit versus Gaussian Kernel Affinities on Vote Data.

To illustrate the superiority of Gaussian kernel affinities over unit affinities often observed on real data, we compute the convex clustering solution paths under the two kinds of affinities on US senate voting data in 2001 [1, 14]. We removed duplicate voting records, restricting our attention to 29 senators – 15 Democrats, 13 Republicans, and 1 Independent (Jim Jeffords, who was a Republican prior to 2001) – and their votes on 13 issues ranging over domestic, foreign, economic, military, environmental, and social concerns. The raw data consisted of 29



(a) Unit Affinities (b) Gaussian Kernel Affinities

Figure 9: US Senate Vote Data: Solution path as the parameter γ increases.

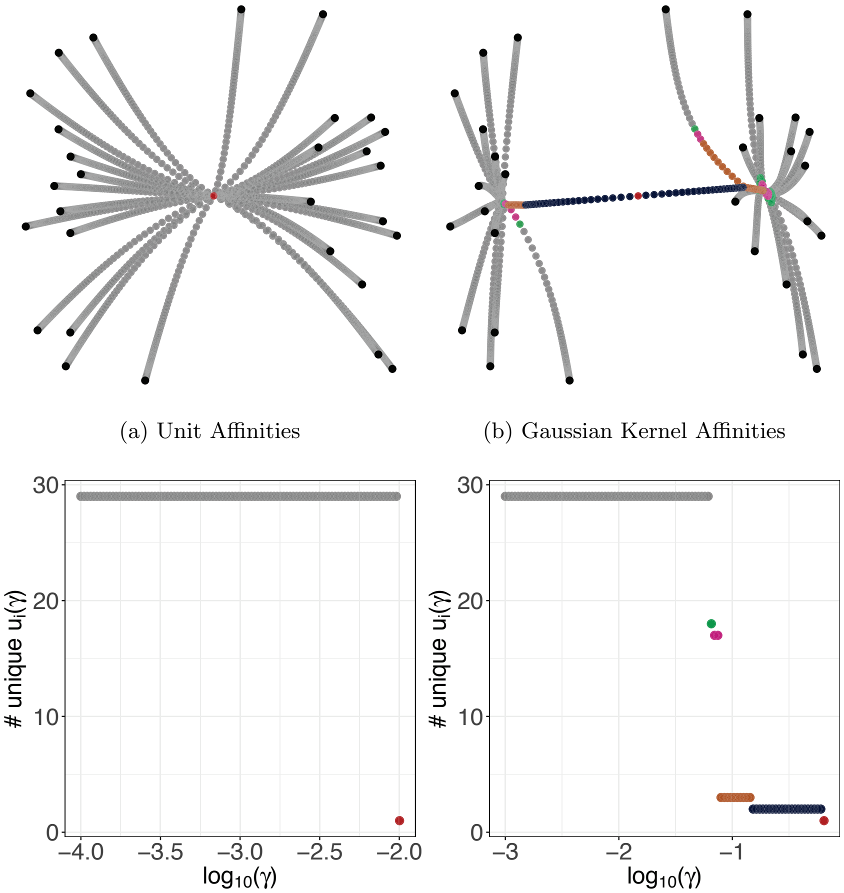
binary vectors of length 13, which we centered and scaled. Figure 9 shows the solution paths under the two kinds of affinities; for visualization purposes we projected $u_i(\gamma) \in \mathbb{R}^{13}$ onto the first two principal components of the centered and scaled data matrix. We color coded the solution paths to reflect senator party affiliations: Democrats in blue, Republicans in red, and the Independent in green. As an aside, we identify an outlying Democrat in Zell Miller, who had a track record for supporting Republican policies during his tenure. He notably supported Republican President George W. Bush against John Kerry, the Democratic nominee in the 2004 presidential election.

Figure 9a and Figure 9b show the resulting clustering paths under unit affinities, $w_{ij} = 1$ for all i and j, and Gaussian kernel affinities respectively. In the latter case, we use a commonly used data-driven strategy of choosing a local scale parameter σ_{ij} that is pair dependent [52], namely

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma_{ij}}\right).$$

We first compute a local measure of scale σ_i, which is the median Euclidean distance between the ith point x_i and its 5-nearest neighbors. We then set $\sigma_{ij} = \sigma_i \sigma_j$.

The solution path in Figure 9a exhibits exactly *one* fusion event as γ increases, namely at the end of the solution path. In contrast, the solution path in Figure 9b exhibits fusions that initially group together senators in their respective parties, before the two main groups fuse at the end of the solution path. Figure 10a and Figure 10b show points along the solution paths obtained from unit and Gaussian kernel affinities respectively, color coded according to



(a) Unit Affinities

(b) Gaussian Kernel Affinities

(c) Unit Affinities

(d) Gaussian Kernel Affinities

Figure 10: US Senate Vote Data: The number of unique $u_i(\gamma)$ as a function of γ.

the number of unique $u_i(\gamma)$ as γ varies. Figure 10c and Figure 10d plot the number of unique $u_i(\gamma)$ as γ varies under unit and Gaussian kernel affinities respectively. Indeed, we see that in this real example, the unit affinities produce a rather useless tree, namely one with *no* nesting at all. In contrast, the Gaussian kernel affinities produce a tree that organizes the senators into partitions that respect party affiliations. Figure 10b also shows that John Chaffee, who was one of the more liberal Republicans, fuses somewhat later to the Republican group and also shows that John Breaux, whose centrist voting tendencies at times led Republicans to seek his help in swaying a few critical Democratic votes, fuses somewhat later to the Democrat group.

Appendix C. Proof of Corollary 3.3.

Proof. Lemma 3.1 guarantees the existence of a point u, call it $\tilde{u}_1$, and viewing direction vector v_1 that satisfies inequality (3.1). Remove $\tilde{u}_1$ from the set $S = \{u_1, \ldots, u_n\}$ and apply Lemma 3.1 to the new set $S \backslash S_1$, where $S_1 = \{\tilde{u}_1\}$. Repeat this procedure k times and let S_k denote the set of k points, $\{\tilde{u}_1, \ldots, \tilde{u}_k\}$, that satisfy inequality (3.1) for the sets $S, S\backslash S_1, \ldots, S\backslash S_{k-1}$ respectively. Lemma 3.1 guarantees the existence of a point $u \in S\backslash S_k$ and viewing direction vector v such that

$$\text{(C.1)} \qquad \frac{1}{n-k} \sum_{\substack{u_i \in S\backslash S_k \\ u_i \neq u}} \left\langle \frac{u_i - u}{\|u_i - u\|}, v \right\rangle \geq \frac{1}{2}.$$

The Cauchy-Bunyakovsky-Schwarz inequality tells us that

$$\text{(C.2)} \qquad \left\langle \frac{u_i - u}{\|u_i - u\|}, v \right\rangle \geq -1,$$

for all $u_i \in S_k$. Inequalities (C.1) and (C.2) together imply that

$$\text{(C.3)} \qquad \sum_{\substack{i=1 \\ u_i \neq u}}^{n} \left\langle \frac{u_i - u}{\|u_i - u\|}, v \right\rangle \quad \geq \quad \frac{n-k}{2} - k$$

Finally, for $k \leq n/6$, we see that the right hand side of (C.3) is bounded below by $n/4$ which implies the desired result. ■

Acknowledgments. We thank Raphy Coifman for pointing out Corollary 3.3.

REFERENCES

[1] Americans for Democratic Action, *2001 voting record: Shattered promise of liberal progress*, ADA Today, 57 (2002), pp. 1–17.

[2] J. I. Ankenman, *Geometry and Analysis of Dual Networks on Questionnaires*, PhD thesis, Yale University, 2014.

[3] M. Belkin and P. Niyogi, *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*, Neural Computation, 15 (2003), pp. 1373–1396.

[4] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, *When Is "Nearest Neighbor" Meaningful?*, in Proceedings of the 7th International Conference on Database Theory, ICDT '99, London, UK, UK, 1999, Springer-Verlag, pp. 217–235.

[5] H. D. Bondell and B. J. Reich, *Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR*, Biometrics, 64 (2008), pp. 115–123.

[6] B. M. Broome, V. Jayaraman, and G. Laurent, *Encoding and Decoding of Overlapping Odor Sequences*, Neuron, 51 (2006), pp. 467–482.

[7] S. L. Brown, J. Joseph, and M. Stopfer, *Encoding a Temporally Structured Stimulus with a Temporally Structured Neural Representation*, Nature Neuroscience, 8 (2005), pp. 1568–76.

[8] L. Carrillo-Reid, F. Tecuapetla, D. Tapia, A. Hernández-Cruz, E. Galarraga, R. Drucker-Colin, and J. Bargas, *Encoding Network States by Striatal Cell Assemblies*, Journal of Neurophysiology, 99 (2008), pp. 1435–1450.

[9] G. K. Chen, E. C. Chi, J. M. Ranola, and K. Lange, *Convex Clustering: An Attractive Alternative to Hierarchical Clustering*, PLoS Computational Biology, 11 (2015), p. e1004228.

[10] E. C. Chi, G. I. Allen, and R. G. Baraniuk, *Convex Biclustering*, Biometrics, 73 (2017), pp. 10–19.

[11] E. C. Chi, B. R. Gaines, W. W. Sun, H. Zhou, and J. Yang, *Provable convex co-clustering of tensors*. arXiv:1803.06518 [stat.ME], 2018.

[12] E. C. Chi and K. Lange, *Splitting Methods for Convex Clustering*, Journal of Computational and Graphical Statistics, 24 (2015), pp. 994–1013.

[13] R. R. Coifman and S. Lafon, *Diffusion Maps*, Applied and Computational Harmonic Analysis, 21 (2006), pp. 5–30.

[14] J. de Leeuw and P. Mair, *Gifi methods for optimal scaling in R: The Package homals*, Journal of Statistical Software, 31 (2009), pp. 1–21.

[15] D. L. Donoho and C. Grimes, *Hessian Eigenmaps: Locally Linear Embedding Techniques for High-dimensional Data*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 5591–5596.

[16] J. Fan and R. Li, *Variable selection via nonconcave penalized likelihood and its oracle properties*, Journal of the American Statistical Association, 96 (2001), pp. 1348–1360.

[17] J. M. García-Gómez, J. Gómez-Sanchis, P. Escandell-Montero, E. Fuster-Garcia, and E. Soria-Olivas, *Sparse Manifold Clustering and Embedding to Discriminate Gene Expression Profiles of Glioblastoma and Meningioma Tumors*, Computers in Biology and Medicine, 43 (2013), pp. 1863–1869.

[18] J. C. Gower and G. J. S. Ross, *Minimum spanning trees and single linkage cluster analysis*, Journal of the Royal Statistical Society. Series C (Applied Statistics), 18 (1969), pp. 54–64.

[19] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert, *Clusterpath an algorithm for clustering using convex fusion penalties*, in Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 745–752.

[20] X. Jiang, X. Hu, H. Shen, and T. He, *Manifold Learning Reveals Nonlinear Structure in Metagenomic Profiles*, in 2012 IEEE International Conference on Bioinformatics and Biomedicine, 2012, pp. 1–6.

[21] S. C. Johnson, *Hierarchical clustering schemes*, Psychometrika, 32 (1967), pp. 241–254.

[22] Z. T. Ke, J. Fan, and Y. Wu, *Homogeneity pursuit*, Journal of the American Statistical Association, 110 (2015), pp. 175–194.

[23] G. N. Lance and W. T. Williams, *A general theory of classificatory sorting strategies: 1. hierarchical systems*, The Computer Journal, 9 (1967), pp. 373–380.

[24] K. Lange, D. R. Hunter, and I. Yang, *Optimization transfer using surrogate objective functions*, Journal of Computational and Graphical Statistics, 9 (2000), pp. 1–20.

[25] F. Lindsten, H. Ohlsson, and L. Ljung, *Just relax and come clustering! A convexification of k-means clustering*, tech. report, Linköpings Universitet, 2011.

[26] Y. Marchetti and Q. Zhou, *Solution path clustering with adaptive concave penalty*, Electronic Journal of Statistics, 8 (2014), pp. 1569–1603.

[27] E. Marras, A. Travaglione, and E. Capobianco, *Manifold Learning in Protein Interactomes*, Journal of Computational Biology, 18 (2010), pp. 81–96.

[28] G. Mishne, R. Talmon, I. Cohen, R. R. Coifman, and Y. Kluger, *Data-driven tree transforms and metrics*, IEEE Transactions on Signal and Information Processing over Networks, 4 (2018), pp. 451–466.

[29] G. Mishne, R. Talmon, R. Meir, J. Schiller, M. Lavzin, U. Dubin, and R. R. Coifman, *Hierarchical coupled-geometry analysis for neuronal structure and activity pattern discovery*, IEEE Journal of Selected Topics in Signal Processing, 10 (2016), pp. 1238–1253.

[30] F. Murtagh, *A survey of recent advances in hierarchical clustering algorithms*, The Computer Journal, 26 (1983), pp. 354–359.

[31] W. Pan, X. Shen, and B. Liu, *Cluster analysis: Unsupervised learning via supervised learning with a non-convex penalty*, Journal of Machine Learning Research, 14 (2013), pp. 1865–1889.

[32] A. Panahi, D. Dubhashi, F. D. Johansson, and C. Bhattacharyya, *Clustering by Sum of Norms: Stochastic Incremental Algorithm, Convergence and Cluster Recovery*, in Proceedings of the 34th International Conference on Machine Learning, D. Precup and Y. W. Teh, eds., vol. 70 of Proceedings of Machine Learning Research, International Convention Centre, Sydney, Australia, 06–11 Aug 2017, PMLR, pp. 2769–2777.

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

[34] P. Radchenko and G. Mukherjee, *Convex clustering via l1 fusion penalization*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 79 (2017), pp. 1527–1546.

[35] S. T. Roweis and L. K. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, Science, 290 (2000), pp. 2323–2326.

[36] D. Saha, K. Leong, C. Li, S. Peterson, G. Siegel, and B. Raman, *A Spatiotemporal Coding Mechanism for Background-Invariant Odor Recognition*, Nature Neuroscience, 16 (2013), pp. 1830–1839.

[37] E. D. Schifano, R. L. Strawderman, and M. T. Wells, *Majorization-minimization algorithms for nonsmoothly penalized objective functions*, Electronic Journal of Statistics, 4 (2010), pp. 1258–1299.

[38] J. Sharpnack, A. Singh, and A. Rinaldo, *Sparsistency of the edge lasso over graphs*, in AISTATS, 2012.

[39] Y. She, *Sparse regression with exact clustering*, Electronic Journal of Statistics, 4 (2010), pp. 1055–1096.

[40] M. Stopfer, V. Jayaraman, and G. Laurent, *Intensity versus Identity Coding in an Olfactory System*, Neuron, 39 (2003), pp. 991–1004.

[41] D. Sun, K.-C. Toh, and Y. Yuan, *Convex Clustering: Model, Theoretical Guarantee and Efficient Algorithm*. arXiv:1810.02677 [cs.LG], 2018.

[42] K. M. Tan and D. Witten, *Statistical properties of convex clustering*, Electronic Journal of Statistics, 9 (2015), pp. 2324–2347.

[43] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, Science, 290 (2000), pp. 2319–2323.

[44] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, *Sparsity and smoothness via the fused lasso*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2005), pp. 91–108.

[45] J. T. Vogelstein, Y. Park, T. Ohyama, R. A. Kerr, J. W. Truman, C. E. Priebe, and M. Zlatic, *Discovery of Brainwide Neural-Behavioral Maps via Multiscale Unsupervised Structure Learning*, Science, 344 (2014), pp. 386–392.

[46] B. Wang, Y. Zhang, W. W. Sun, and Y. Fang, *Sparse convex clustering*, Journal of Computational and Graphical Statistics, 27 (2018), pp. 393–403.

[47] J. H. Ward, *Hierarchical grouping to optimize an objective function*, Journal of the American Statistical Association, 58 (1963), pp. 236–244.

[48] D. M. Witten, A. Shojaie, and F. Zhang, *The cluster elastic net for high-dimensional regression with unknown variable grouping*, Technometrics, 56 (2014), pp. 112–122.

[49] C. Wu, S. Kwon, X. Shen, and W. Pan, *A new algorithm and theory for penalized regression-based clustering*, Journal of Machine Learning Research, 17 (2016), pp. 1–25.

[50] Z.-H. You, Y.-K. Lei, J. Gui, D.-S. Huang, and X. Zhou, *Using Manifold Embedding for Assessing and Predicting Protein Interactions from High-throughput Experimental Data*, Bioinformatics, 26 (2010), pp. 2744–2751.

[51] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68 (2006), pp. 49–67.

[52] L. Zelnik-Manor and P. Perona, *Self-tuning spectral clustering*, in Advances in Neural Information Processing Systems 17, L. K. Saul, Y. Weiss, and L. Bottou, eds., MIT Press, 2005, pp. 1601–1608.

[53] C.-H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, The Annals of Statistics, 38 (2010), pp. 894–942.

[54] C. Zhu, H. Xu, C. Leng, and S. Yan, *Convex optimization procedure for clustering: Theoretical revisit*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., 2014, pp. 1619–1627.

[55] H. Zou and R. Li, *One-step sparse estimates in nonconcave penalized likelihood models*, The Annals of Statistics, 36 (2008), pp. 1509–1533.