Pattern Recognition Letters
journal homepage: www.elsevier.com

Forward-Stagewise Clustering: An Algorithm for Convex Clustering

Mimi Zhang[a,**]

[a] *School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, Ireland*

ABSTRACT

This paper proposes an exceptionally simple algorithm, called forward-stagewise clustering, for convex clustering. Convex clustering has drawn recent attention since it nicely addresses the instability issue of traditional non-convex clustering methods. While existing algorithms can precisely solve convex clustering problems, they are sophisticated and produce (agglomerative) clustering paths that contain splits. This motivates us to propose an algorithm that only produces no-split clustering paths. The approach undertaken here follows the line of research initiated in the area of regression. Specifically, we apply the forward-stagewise technique to clustering problems and prove that the algorithm can only produce no-split clustering paths. We then modify the forward-stagewise clustering algorithm to deal with noise and outliers. We further suggest rules of thumb for the algorithm to be applicable to cases where clusters are non-convex. The performance of the proposed algorithm is evaluated through simulations and a real data application.

1. Introduction

The field of clustering is crowded with diverse methods that make particular assumptions about data and address different issues. One limitation of traditional clustering methods is the non-convexity of the corresponding optimization problems. Another common challenge comes from the hyper-parameter deciding in terms of the number of clusters. Recently, several convex clustering methods have been proposed (Hocking et al., 2011; Lindsten et al., 2011). Convex clustering leverages sparsity-inducing norms and enjoys many attractive theoretical properties. Particularly, convex clustering applies convex relaxations on traditional non-convex clustering criteria and does not need the cluster number a priori or careful initializations. Speed and scalability of these algorithms make them increasingly popular for clustering analysis of big data.

Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ denote the data matrix: $\boldsymbol{X}' = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, where the prime represents the transpose operator, and each object $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ is described by p features. Hocking et al. (2011) formulate the clustering of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ as a convex optimization problem:

$$\min_{\boldsymbol{U}\in\mathbb{R}^{n\times p}} F_\lambda(\boldsymbol{U}) = \min_{\boldsymbol{U}\in\mathbb{R}^{n\times p}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\|_F^2 + \lambda\sum_{i<j} w_{ij}\|\boldsymbol{u}_i-\boldsymbol{u}_j\|_q, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $\|\cdot\|_q$ is the ℓ_q norm with $q \in \{1, 2, \infty\}$. $\boldsymbol{U}' = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$, where $\boldsymbol{u}_i$ $(i = 1, \ldots, n)$ is the centroid of the cluster that object $\boldsymbol{x}_i$ belongs to. Two objects $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with $\boldsymbol{u}_i = \boldsymbol{u}_j$ are said to belong to the same cluster. λ is a nonnegative tuning parameter, and the w_{ij}'s are nonnegative weights. The weights $\{w_{ij} : 1 \le i < j \le n\}$ are usually defined to be distance-dependent, e.g. $w_{ij} = \exp(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2)$, in order to make the estimates of the centroids enjoy asymptotic consistency, in the manner of the adaptive lasso (Zou, 2006). The second term in (1) is a regularizer, putting a constraint on the number of distinct cluster centroids. If $\lambda = 0$, the minimum is attained when $\boldsymbol{U} = \boldsymbol{X}$, and each point $\boldsymbol{x}_i$ occupies a unique cluster $\boldsymbol{u}_i$. As λ increases, the cluster centroids begin to coalesce. For sufficiently large λ, all related points will coalesce into a single cluster.

The objective function $F_\lambda(\boldsymbol{U})$ is strictly convex and coercive, and hence problem (1) has a unique minimum for each value of λ. The well-known alternating direction method of multipliers (Boyd et al., 2011), abbreviated as ADMM, can be applied to solve problem (1). Specifically, the ADMM solves the equiva-

**Corresponding author: Tel. +353-1896-2726.
e-mail: `Mimi.Zhang@tcd.ie` (Mimi Zhang)

lent problem

$$\begin{aligned}\min_{\boldsymbol{U},\boldsymbol{Z}} \quad & \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\|_F^2+\lambda\sum_{i<j} w_{ij}\|\boldsymbol{z}_{ij}\|_q+\frac{\rho}{2}\sum_{i<j}\|\boldsymbol{u}_i-\boldsymbol{u}_j-\boldsymbol{z}_{ij}\|_2^2,\\ \text{subject to} \quad & \boldsymbol{u}_i-\boldsymbol{u}_j-\boldsymbol{z}_{ij}=0,\ \ 1\le i<j\le n.\end{aligned}$$

ρ is a nonnegative tuning parameter. Define a matrix $\boldsymbol{Z}\in\mathbb{R}^{p\times\frac{n(n-1)}{2}}$, with the $\boldsymbol{z}_{ij}$'s being its columns. The corresponding Lagrangian is

$$\begin{aligned}L_\rho(\boldsymbol{U},\boldsymbol{Z},\{\boldsymbol{v}_{ij}\}) =& \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{U}\|_F^2+\lambda\sum_{i<j} w_{ij}\|\boldsymbol{z}_{ij}\|_q\\ &+\frac{\rho}{2}\sum_{i<j}\|\boldsymbol{u}_i-\boldsymbol{u}_j-\boldsymbol{z}_{ij}\|_2^2+\sum_{i<j}\boldsymbol{v}_{ij}'(\boldsymbol{u}_i-\boldsymbol{u}_j-\boldsymbol{z}_{ij}),\end{aligned}$$

where each $\boldsymbol{v}_{ij}$ is a vector of Lagrangian multipliers. For $t=0,1,2,\ldots$, the $(t{+}1)$st iteration of the ADMM algorithm consists of the following 3 steps:

$$\begin{aligned}\boldsymbol{U}^{t+1} &\leftarrow \operatorname*{arg\,min}_{\boldsymbol{U}} L_\rho(\boldsymbol{U},\boldsymbol{Z}^t,\{\boldsymbol{v}_{ij}^t\});\\ \boldsymbol{Z}^{t+1} &\leftarrow \operatorname*{arg\,min}_{\boldsymbol{Z}} L_\rho(\boldsymbol{U}^{t+1},\boldsymbol{Z},\{\boldsymbol{v}_{ij}^t\});\\ \boldsymbol{v}_{ij}^{t+1} &\leftarrow \boldsymbol{v}_{ij}^t+\rho(\boldsymbol{u}_i^{t+1}-\boldsymbol{u}_j^{t+1}-\boldsymbol{z}_{ij}^{t+1}).\end{aligned}$$

For many problems of interest, the updates of $\boldsymbol{U}$ and $\boldsymbol{Z}$ can be evaluated by either an explicit formula or an efficient algorithm. Moreover, the minimization can be carried out separately in parallel. Further algorithmic developments can be found in Chi and Lange (2015), Hallac et al. (2015), Chen et al. (2015) and Han and Zhang (2016).

In this paper, we propose an extremely simple algorithm for problem (1) with $q=1$, utilizing the idea of forward stagewise. The forward-stagewise algorithm developed in Section 3 may take many more iterations than existing convex clustering algorithms. However, the computation of each iteration is exceptionally simple. Furthermore, compared with existing convex clustering algorithms, the forward-stagewise approach gives a smoother clustering path; that is, the clustering path does not have any split. Under the ℓ_1 norm penalty, problem (1) is a fused lasso (Tibshirani et al., 2005). The lasso has undergone intense study, and many of its properties cast the lasso in a favorable light. The equivalence between the (limiting) forward-stagewise and lasso paths lends credibility to forward stagewise as a regularized estimator. At a high level, forward stagewise, in spite of being simple, can produce estimates that stand alongside those defined by the relatively sophisticated optimization problem (Tibshirani, 2015).

The rest of this paper is organized as follows. Section 2 briefly discusses related work from the literature, focusing on the theoretical aspect. Section 3 presents the forward-stagewise algorithm and its properties. Section 4 discusses how to deal with noise and outliers. In Section 5, we conduct a detailed empirical analysis of our approach. Finally, the paper concludes in Section 6 with a summary of new insights and recommendations for future research.

2. Related Work

Theoretical properties of convex clustering have been investigated for the particular case $w_{ij}=1$ (Zhu et al., 2014; Tan and Witten, 2015; Radchenko and Mukherjee, 2017). Zhu et al. (2014) show that if all samples are drawn from two clusters, each being a cube, then problem (1) is guaranteed to successfully recover the cluster membership provided that the distance between the two cubes is greater than a threshold; the threshold depends on the cube size and the ratio between the numbers of the samples in each cluster. Tan and Witten (2015) give the range of the tuning parameter λ such that convex clustering yields a non-trivial solution with more than one cluster. They also provide an unbiased estimator of the degrees of freedom and establish finite sample bounds for the prediction error (assuming that the elements in the difference matrix $\boldsymbol{X}-\boldsymbol{U}$ are independent sub-Gaussian random variables). Radchenko and Mukherjee (2017) develop a computationally efficient sample merging procedure (for producing a path of solutions) and an equivalent sample splitting procedure (which can recover all the corresponding cluster splits by solving a sequence of maximization problems). From the sample splitting procedure, Radchenko and Mukherjee (2017) further develop a population splitting procedure, showing that under some very mild regularity conditions the sample splitting procedure consistently estimates its population analog. On the basis of new perspectives gained from the population splitting procedure, they propose a post-processing modification of the original sample merging procedure by keeping only the merges with significant empirical sizes.

Wang et al. (2016) and Sui et al. (2018) point out that the Euclidean metric treats each feature equally, and hence the performance of a convex clustering algorithm can deteriorate significantly in the presence of outlier features. Wang et al. (2016) assume that the data matrix can be decomposed into two parts: $\boldsymbol{X}=\boldsymbol{U}+\boldsymbol{Q}$, where the clustering component $\boldsymbol{U}$ captures the clustering structure, while the robust component $\boldsymbol{Q}$ identifies the outlier features in the data. The convex clustering is performed on the purified data $(\boldsymbol{X}-\boldsymbol{Q})$. The objective function is

$$\min_{\boldsymbol{U},\boldsymbol{Q}} \ \frac{1}{2}\|(\boldsymbol{X}-\boldsymbol{Q})-\boldsymbol{U}\|_F^2+\lambda_1\sum_{i<j}w_{ij}\|\boldsymbol{u}_i-\boldsymbol{u}_j\|_q+\lambda_2\|\boldsymbol{Q}\|_{2,1},$$

where the group-lasso penalty is used to regularize $\boldsymbol{Q}$ and achieve the desired column-wise sparsity. $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$ norm, i.e., the sum of ℓ_2 norm. If a feature is useful, the corresponding column in $\boldsymbol{Q}$ will be zero; if a feature is outlier, the corresponding column in $\boldsymbol{Q}$ will be non-zero for all elements. Wang et al. (2016) iteratively fix one of the $\{\boldsymbol{U},\ \boldsymbol{Q}\}$ matrices and optimize w.r.t. the other, until certain convergence criteria are achieved. Sui et al. (2018) modify problem (1) by introducing a positive definite matrix $\boldsymbol{Q}$:

$$\begin{aligned}\min_{\boldsymbol{U},\boldsymbol{Q}} \quad & \frac{1}{2}\sum_{i=1}^n(\boldsymbol{x}_i-\boldsymbol{u}_i)'\boldsymbol{Q}(\boldsymbol{x}_i-\boldsymbol{u}_i)+\lambda\sum_{i<j}w_{ij}\|\boldsymbol{u}_i-\boldsymbol{u}_j\|_q,\\ \text{subject to} \quad & \operatorname{logdet}(\boldsymbol{Q})\ge 0.\end{aligned}$$

The structure of the matrix $\boldsymbol{Q}$ shows which features are more congruent with the cluster assignment. In particular, when $\boldsymbol{Q}$ is diagonal, the larger diagonal values of $\boldsymbol{Q}$ correspond to the features that are of higher relevance or of lower noise corruptions. To solve the above optimization problem, Sui et al. (2018) propose an alternating procedure that alternates between minimizing over $\boldsymbol{U}$ and minimizing over $\boldsymbol{Q}$. More specifically, for a fixed $\boldsymbol{Q}$, the optimization problem can be solved in the ADMM framework, while for a fixed $\boldsymbol{U}$, the optimization problem admits a closed-form solution.

Define $\hat{\boldsymbol{U}}(\lambda)' = [\hat{\boldsymbol{u}}_1(\lambda),\dots,\hat{\boldsymbol{u}}_n(\lambda)]$, where $\hat{\boldsymbol{U}}(\lambda) = \arg\min_{\boldsymbol{U}} F_\lambda(\boldsymbol{U})$. A convex clustering algorithm usually starts with $\lambda = 0$, corresponding to the hierarchical agglomerative clustering. However, for existing algorithms, the solution path to (1) generally can split: $\hat{\boldsymbol{u}}_i(\lambda_1) = \hat{\boldsymbol{u}}_j(\lambda_1)$, yet $\hat{\boldsymbol{u}}_i(\lambda_2) \neq \hat{\boldsymbol{u}}_j(\lambda_2)$ for certain $\lambda_2 > \lambda_1$. Indeed, this phenomenon is ubiquitous in the lasso regression, where a coefficient path can pass through zero. The splitting of the solution path in convex clustering implies that a merged cluster then splits into two sub-clusters. If this is the case, the corresponding agglomerative structure is no longer a tree and difficult to interpret. Hocking et al. (2011) propose different algorithms for the three types of norms: $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$. They prove that, when $w_{ij} = 1$ and $q = 1$, the solution path contains no split, and hence the inferred structure is a tree. Chiquet et al. (2017) prove that, for any ℓ_q norm, the solution path contains no split if $w_{ij} = n_i \times n_j$, where n_i is the size of cluster $\boldsymbol{u}_i$. However, for such weights, the recovered tree structure is often unbalanced: two centroids initially close to one another at $\lambda = 0$ fuse relatively late in the path of solutions. Distance-dependent weights should ensure that close neighbors fuse quickly. Hence, Chiquet et al. (2017) further prove that, for the ℓ_1 norm, the solution path contains no split if $w_{ij} = n_i \times n_j \times h(|\bar{\boldsymbol{x}}_{c(i)} - \bar{\boldsymbol{x}}_{c(j)}|)$, where $h(\cdot)$ is a decreasing positive function, and $\bar{\boldsymbol{x}}_{c(i)}$ is the average of the objects in cluster $c(i)$ – the cluster that object $\boldsymbol{x}_i$ belongs to.

The forward-stagewise strategy for regression is a "slow learning" method. However, it turns out that this "slow fitting" is a form of regularization and can present considerable benefits in terms of the generalization error of the fitted models. Efron et al. (2004) show that the sequence of forward-stagewise estimates and the solution path of the lasso are both piece-wise linear. Hastie et al. (2007) show that the infinitesimal forward stagewise fits a monotone version of the lasso, which optimally reduces at each step the loss function for a given increase in the arc length of the coefficient path. Even in situations where stagewise estimates differ from lasso solutions, the former estimates can still perform competitively with the latter. By modern computational resources, forward stagewise is computationally cheap: to trace out a path of regularized estimates, we repeat very simple iterations which could be trivially parallelized. Inspired by the stable feature of forward-stagewise solution paths, this paper proposes the forward-stagewise strategy for clustering, with the aim of obtaining merging-only clustering paths.

3. Forward-Stagewise Clustering

When $q = 1$, the objective function $F_\lambda(\boldsymbol{U})$ is separable, and hence the minimization can be carried out separately in parallel for each dimension. We might let the generic vector $\boldsymbol{x}$ represent an arbitrary column of $\boldsymbol{X}$, and $\boldsymbol{u}$ the corresponding column of $\boldsymbol{U}$. Solving (1) amounts to solving p minimization problems of the following form:

$$\hat{\boldsymbol{u}}_\lambda = \underset{\boldsymbol{u}\in\mathbb{R}^n}{\arg\min}\ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_2^2 + \lambda\sum_{i<j} w_{ij}|u_i - u_j|. \tag{2}$$

Define a (sparse) matrix $\boldsymbol{D} \in \mathbb{R}^{\frac{n(n-1)}{2}\times n}$:

$$\boldsymbol{D} = \begin{pmatrix} w_{12} & -w_{12} & 0 & 0 & 0 & \cdots & 0 & 0 \\ w_{13} & 0 & -w_{13} & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ w_{1n} & 0 & 0 & 0 & 0 & \cdots & 0 & -w_{1n} \\ 0 & w_{23} & -w_{23} & 0 & 0 & \cdots & 0 & 0 \\ 0 & w_{24} & 0 & -w_{24} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & w_{2n} & 0 & 0 & 0 & \cdots & 0 & -w_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & w_{(n-1)n} & -w_{(n-1)n} \end{pmatrix}.$$

Then problem (2) can be rewritten as

$$\hat{\boldsymbol{u}}_\lambda = \underset{\boldsymbol{u}\in\mathbb{R}^n}{\arg\min}\ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{u}\|_1, \tag{3}$$

which is a generalized lasso problem (Tibshirani and Taylor, 2011). Problem (3) is difficult to analyze directly because the nondifferentiable ℓ_1 penalty is composed with a linear transformation of $\boldsymbol{u}$. Hence, we turn to the dual problem of (3) and recover the primal solution $\hat{\boldsymbol{u}}_\lambda$ from the dual solution.

Problem (3) is equivalent to

$$\min_{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^{\frac{n(n-1)}{2}}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_2^2 + \lambda\|\boldsymbol{z}\|_1, \quad \text{subject to } \boldsymbol{D}\boldsymbol{u}-\boldsymbol{z} = 0,$$

for which the dual problem is

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}\in\mathbb{R}^{\frac{n(n-1)}{2}}}{\arg\min}\ f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{D}'\boldsymbol{\beta}\|_2^2, \quad \text{subject to } g(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_\infty \leq \lambda. \tag{4}$$

While (3) is strictly convex, its dual problem (4) is not strictly convex, since $\boldsymbol{D}$ is not of full rank. Therefore, the solution to (4) is not unique. It is easy to prove that the primal and dual solutions satisfy the relationship

$$\hat{\boldsymbol{u}}_\lambda = \boldsymbol{x} - \boldsymbol{D}'\hat{\boldsymbol{\beta}}_\lambda. \tag{5}$$

Equation (5) implies that, given a dual solution, the corresponding primal solution can be uniquely determined.

Tibshirani (2015) develops a general stagewise algorithm, in Algorithm 1, for problems of the form:

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}), \quad \text{subject to } g(\boldsymbol{\beta}) \leq \lambda,$$

with the assumptions that $f(\cdot)$ and $g(\cdot)$ are convex, and $f(\cdot)$ is differentiable. Algorithm 1 can be treated as gradient descent:

Algorithm 1 General Stagewise Algorithm

```
1: Fix $\varepsilon > 0$ and $\lambda = 0$.
2: Initialize $\boldsymbol{\beta}_0 = \hat{\boldsymbol{\beta}}_0 = \mathbf{0}$.
3: for $k = 1, 2, 3, \ldots,$ do
4:     $\Delta \in \arg\min_{z} \langle \nabla f(\boldsymbol{\beta}_{k-1}),\, \boldsymbol{z}\rangle$ subject to $g(\boldsymbol{z}) \le \varepsilon$;
5:     $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \Delta$;
6: end for
```

$f(\boldsymbol{\beta}_{k-1})$ decreases fastest if one goes from $\boldsymbol{\beta}_{k-1}$ in the direction $-\nabla f(\boldsymbol{\beta}_{k-1})$. However, we want to counterbalance the greediness by forcing the increment $g(\boldsymbol{\beta}_k) - g(\boldsymbol{\beta}_{k-1})$ to be small. If $g(\cdot)$ satisfies the triangle inequality (e.g., when $g(\cdot)$ is a norm), then the increase in the value of $g(\cdot)$ between successive iterates is bounded by ε:

$$g(\boldsymbol{\beta}_k) - g(\boldsymbol{\beta}_{k-1}) = g(\boldsymbol{\beta}_{k-1} + \Delta) - g(\boldsymbol{\beta}_{k-1}) \le g(\Delta) \le \varepsilon.$$

To obtain the solution path of the primal problem (3) for various values of λ, we can apply Algorithm 1 on the dual problem (4) to obtain the dual solution path $\{\boldsymbol{\beta}_k\}$. Then the primal solution path $\{\boldsymbol{u}_k\}$ can be readily determined from $\{\boldsymbol{\beta}_k\}$ via (5); the λ value for $\boldsymbol{u}_k$ is $\|\boldsymbol{\beta}_k\|_\infty$. The idea is represented succinctly by forward-stagewise clustering in Algorithm 2, which is explained as follows. For problem (4), we have

Algorithm 2 Forward-Stagewise Clustering

```
1: Fix $\varepsilon > 0$.
2: Initialize $\boldsymbol{\beta}_0 = \mathbf{0}$ and $\boldsymbol{u}_0 = \boldsymbol{x}$.
3: for $k = 1, 2, 3, \ldots,$ do
4:     $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \varepsilon \times \mathrm{sign}(\boldsymbol{D}\boldsymbol{u}_{k-1})$;
5:     $\boldsymbol{u}_k = \boldsymbol{x} - \boldsymbol{D}'\boldsymbol{\beta}_k$;
6: end for
```

$$\nabla f(\boldsymbol{\beta}_{k-1}) = \boldsymbol{D}(\boldsymbol{D}'\boldsymbol{\beta}_{k-1} - \boldsymbol{x}) = -\boldsymbol{D}\boldsymbol{u}_{k-1},$$

where we have utilized the relationship (5). Hence, step 4 in Algorithm 1 reduces to

$$\Delta \in \arg\max_{z} \langle \boldsymbol{D}\boldsymbol{u}_{k-1},\, \boldsymbol{z}\rangle, \quad \text{subject to } \|\boldsymbol{z}\|_\infty \le \varepsilon$$

It is easy to prove that: if $[\boldsymbol{D}\boldsymbol{u}_{k-1}]_i < 0$, then $\Delta_i = -\varepsilon$; if $[\boldsymbol{D}\boldsymbol{u}_{k-1}]_i > 0$, then $\Delta_i = \varepsilon$. If $[\boldsymbol{D}\boldsymbol{u}_{k-1}]_i = 0$, we set $\Delta_i = 0$, which makes the dual solution unique. Hence, step 5 in Algorithm 1 becomes $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \varepsilon \times \mathrm{sign}(\boldsymbol{D}\boldsymbol{u}_{k-1})$. Here, $\mathrm{sign}(\cdot)$ is to be interpreted componentwise with the convention $\mathrm{sign}(0) = 0$. Step 5 in Algorithm 2 calculates the corresponding primal solution.

The computational load of Algorithm 2 depends mainly on the two matrix multiplications: one by $\boldsymbol{D}$ and one by $\boldsymbol{D}'$. Since $\boldsymbol{D}$ is sparse, the computation of steps 4 and 5 is cheap. It may seem that the memory requirement for the matrix $\boldsymbol{D}$ is high, and the algorithm will deplete all available memory when dealing with big data. However, as $\boldsymbol{D}$ has a particular structure, it can be parsimoniously stored in the form of a vector $(w_{12}, w_{13}, \ldots, w_{(n-1)n})$, which reduces the storage from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. Finally, assuming $\hat{\boldsymbol{u}}$ is the limit of the sequence $\{\boldsymbol{u}_k\}$, the number of iterations until convergence is bounded by $\mathcal{O}(\max\{\frac{|u_i - \hat{u}_i|}{\varepsilon \times \min\{w_{ij} > 0 : 1 \le j \le n\}} : 1 \le i \le n\})$.

The primal update can be expressed succinctly from steps 4 and 5 of Algorithm 2:

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} - \varepsilon \times \boldsymbol{D}'\mathrm{sign}(\boldsymbol{D}\boldsymbol{u}_{k-1}).$$

Write $\boldsymbol{D}' = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_\eta]$, where $\eta = \frac{n(n-1)}{2}$ and $\boldsymbol{d}_i'$ ($i = 1, \ldots, \eta$) is the ith row of $\boldsymbol{D}$. The above primal update is equivalent to

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} - \varepsilon \times \sum_{i=1}^{\eta} \mathrm{sign}(\boldsymbol{d}_i'\boldsymbol{u}_{k-1})\boldsymbol{d}_i. \tag{6}$$

If $\boldsymbol{d}_i'\boldsymbol{u}_{k-1} > 0$ (i.e., the ith row of $\boldsymbol{D}$ is active), then the algorithm adds $-\varepsilon\boldsymbol{d}_i$ to $\boldsymbol{u}_{k-1}$ in forming $\boldsymbol{u}_k$. Likewise, if $\boldsymbol{d}_i'\boldsymbol{u}_{k-1} < 0$, then the algorithm adds $\varepsilon\boldsymbol{d}_i$ to $\boldsymbol{u}_{k-1}$. Hence, Equation (6) implies that Algorithm 2 iteratively shrinks along directions opposite to the active rows of $\boldsymbol{D}$.

3.1. Identical Weights

It is intuitive that the solution path $\{\boldsymbol{u}_k\}$ is piecewise linear. We below prove that, when the weights $\{w_{ij}\}$ are identical, the forward-stagewise clustering path has no splits. Without loss of generality, assume $w_{ij} = 1$ for $1 \le i < j \le n$. Then we only need to prove that $|\boldsymbol{d}_l'\boldsymbol{u}_k| \le |\boldsymbol{d}_l'\boldsymbol{u}_{k-1}|$, $\forall\ l = 1, \ldots, \eta$. From Equation (6) we have

$$\boldsymbol{d}_l'\boldsymbol{u}_k = \boldsymbol{d}_l'\boldsymbol{u}_{k-1} - \varepsilon \times \sum_{i=1}^{\eta} \mathrm{sign}(\boldsymbol{d}_i'\boldsymbol{u}_{k-1})\boldsymbol{d}_l'\boldsymbol{d}_i. \tag{7}$$

We illustrate the proof through the case $\boldsymbol{d}_l = \boldsymbol{d}_1 = (1, -1, 0, \ldots, 0)'$. When $\boldsymbol{d}_l = \boldsymbol{d}_1$, most of the products $\boldsymbol{d}_1'\boldsymbol{d}_i$ are zero, except when $\boldsymbol{d}_i \in \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{2n-3}\}$. Hence we have

$$\begin{aligned} u_k^1 - u_k^2 = & u_{k-1}^1 - u_{k-1}^2 - \mathrm{sign}(u_{k-1}^1 - u_{k-1}^2) \times 2\varepsilon \\ & - \varepsilon[\sum_{i=3}^{n} \mathrm{sign}(u_{k-1}^1 - u_{k-1}^i) - \sum_{i=3}^{n} \mathrm{sign}(u_{k-1}^2 - u_{k-1}^i)]. \end{aligned}$$

If $u_{k-1}^1 > u_{k-1}^2$, then $\sum_{i=3}^n \mathrm{sign}(u_{k-1}^1 - u_{k-1}^i) \ge \sum_{i=3}^n \mathrm{sign}(u_{k-1}^2 - u_{k-1}^i)$ and

$$0 < u_k^1 - u_k^2 \le u_{k-1}^1 - u_{k-1}^2 - 2\varepsilon < u_{k-1}^1 - u_{k-1}^2,$$

given that ε is sufficiently small. Likewise, if $u_{k-1}^1 < u_{k-1}^2$, then $\sum_{i=3}^n \mathrm{sign}(u_{k-1}^1 - u_{k-1}^i) \le \sum_{i=3}^n \mathrm{sign}(u_{k-1}^2 - u_{k-1}^i)$ and

$$0 > u_k^1 - u_k^2 \ge u_{k-1}^1 - u_{k-1}^2 + 2\varepsilon > u_{k-1}^1 - u_{k-1}^2.$$

If $u_{k-1}^1 = u_{k-1}^2$, then $u_k^1 = u_k^2$. Combining all the three cases, we have $|\boldsymbol{d}_1'\boldsymbol{u}_k| \le |\boldsymbol{d}_1'\boldsymbol{u}_{k-1}|$. Therefore, all the points will be grouped into one cluster, and the clustering path does not have any split.

3.2. Non-identical Weights

In the general case when the weights are non-identical, Equation (7) for $\boldsymbol{d}_l = \boldsymbol{d}_1 = (w_{12}, -w_{12}, 0, \dots, 0)'$ is equivalent to

$$u_k^1 - u_k^2 = u_{k-1}^1 - u_{k-1}^2 - \text{sign}(u_{k-1}^1 - u_{k-1}^2) \times 2w_{12}\varepsilon \\ - \varepsilon[\sum_{i=3}^n \text{sign}(u_{k-1}^1 - u_{k-1}^i)w_{1i} - \sum_{i=3}^n \text{sign}(u_{k-1}^2 - u_{k-1}^i)w_{2i}].$$

Define $\Psi_{12}^{k-1} = \sum_{i=3}^n \text{sign}(u_{k-1}^1 - u_{k-1}^i)w_{1i} - \sum_{i=3}^n \text{sign}(u_{k-1}^2 - u_{k-1}^i)w_{2i}$, and hence Equation (7) can be further simplified:

$$u_k^1 - u_k^2 = u_{k-1}^1 - u_{k-1}^2 - \varepsilon\left[\text{sign}(u_{k-1}^1 - u_{k-1}^2) \times 2w_{12} + \Psi_{12}^{k-1}\right].$$

If $u_{k-1}^1 > u_{k-1}^2$, it is not necessary that $\Psi_{12}^{k-1} \geq 0$. Specifically, with ε being small enough, we have

(1) if $-2w_{12} < \Psi_{12}^{k-1} < 2w_{12}$, then $|u_k^1 - u_k^2| < |u_{k-1}^1 - u_{k-1}^2|$;

(2) if $\Psi_{12}^{k-1} \geq 2w_{12}$ and $u_{k-1}^1 < u_{k-1}^2$, then $u_k^1 - u_k^2 \leq u_{k-1}^1 - u_{k-1}^2 < 0$;

(3) if $\Psi_{12}^{k-1} \leq -2w_{12}$ and $u_{k-1}^1 > u_{k-1}^2$, then $u_k^1 - u_k^2 \geq u_{k-1}^1 - u_{k-1}^2 > 0$;

(4) if $\Psi_{12}^{k-1} \geq 2w_{12}$ and $u_{k-1}^1 > u_{k-1}^2$, or if $\Psi_{12}^{k-1} \leq -2w_{12}$ and $u_{k-1}^1 < u_{k-1}^2$, then $|u_k^1 - u_k^2| < |u_{k-1}^1 - u_{k-1}^2|$.

The above cases imply that (1) the distance $|u_k^1 - u_k^2|$ could fluctuate up and down due to the changes of the order of the data $\{u_{k-1}^1, u_{k-1}^2, \dots, u_{k-1}^n\}$; (2) the elements in the final $\boldsymbol{u}_k$ may not be identical. Equation (6) implies that the number of different elements in the final $\boldsymbol{u}_k$ depends on the structure of the matrix $\boldsymbol{D}$. Specifically, we can define an undirected graph G from $\boldsymbol{D}$, for which the nodes are the n data points $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$, and the edges are $\{(i,j) : w_{ij} > 0\}$. If the graph is fully connected, the final $\boldsymbol{u}_k$ will have identical elements; otherwise, if the graph contains m isolated sub-graphs, the final $\boldsymbol{u}_k$ will have m distinct elements.

We now prove that, if $u_{k-2}^1 \neq u_{k-2}^2$ and $u_{k-1}^1 = u_{k-1}^2$, then $u_s^1 = u_s^2$ for any $s \geq k$, which means that the forward-stagewise clustering path cannot have any split. Define $\boldsymbol{t}_i = \text{sign}(\boldsymbol{d}_i)$ and π_i: $\boldsymbol{d}_i = \pi_i \text{sign}(\boldsymbol{d}_i)$. Equation (6) can be written as

$$\begin{pmatrix} u_k^1 \\ u_k^2 \\ \vdots \end{pmatrix} = \begin{pmatrix} u_{k-1}^1 \\ u_{k-1}^2 \\ \vdots \end{pmatrix} \\ - \varepsilon \sum_{i=1}^{\eta} \pi_i \begin{pmatrix} t_i^1 \\ t_i^2 \\ \vdots \end{pmatrix} \text{sign}(\begin{pmatrix} t_i^1 & t_i^2 & \cdots \end{pmatrix} \begin{pmatrix} u_{k-1}^1 \\ u_{k-1}^2 \\ \vdots \end{pmatrix}).$$

Define

$$\begin{pmatrix} \vartheta^1 \\ \vartheta^2 \\ \vdots \end{pmatrix} = \varepsilon \sum_{i=1}^{\eta} \pi_i \begin{pmatrix} t_i^1 \\ t_i^2 \\ \vdots \end{pmatrix} \text{sign}(\begin{pmatrix} t_i^1 & t_i^2 & \cdots \end{pmatrix} \begin{pmatrix} u_{k-2}^1 \\ u_{k-2}^2 \\ \vdots \end{pmatrix}).$$

We have

$$\begin{pmatrix} u_{k-1}^1 \\ u_{k-1}^2 \\ \vdots \end{pmatrix} = \begin{pmatrix} u_{k-2}^1 \\ u_{k-2}^2 \\ \vdots \end{pmatrix} - \begin{pmatrix} \vartheta^1 \\ \vartheta^2 \\ \vdots \end{pmatrix}.$$

We might assume that the difference $\xi = |u_{k-2}^1 - u_{k-2}^2|$ is small enough such that $\text{sign}(u_{k-2}^1 - u_{k-2}^i) = \text{sign}(u_{k-2}^2 - u_{k-2}^i)$ for any $i = 3, \dots, n$. Now that $u_{k-1}^1 = u_{k-1}^2$, we have $\text{sign}(u_{k-1}^1 - u_{k-1}^i) = \text{sign}(u_{k-2}^1 - u_{k-2}^i)$ and $\text{sign}(u_{k-1}^2 - u_{k-1}^i) = \text{sign}(u_{k-2}^2 - u_{k-2}^i)$ for any $i = 3, \dots, n$. Therefore, we have

$$\begin{pmatrix} u_k^1 \\ u_k^2 \\ \vdots \end{pmatrix} = \begin{pmatrix} u_{k-1}^1 \\ u_{k-1}^2 \\ \vdots \end{pmatrix} - \begin{pmatrix} \vartheta^1 \\ \vartheta^2 \\ \vdots \end{pmatrix},$$

and

$$|u_k^1 - u_k^2| = |u_{k-1}^1 - \vartheta^1 - u_{k-1}^2 + \vartheta^2| = |\vartheta^1 - \vartheta^2|.$$

Recall that $u_{k-2}^1 - \vartheta^1 = u_{k-1}^1 = u_{k-1}^2 = u_{k-2}^2 - \vartheta^2$, and hence $|\vartheta^1 - \vartheta^2| = |u_{k-2}^1 - u_{k-2}^2| = \xi$. Therefore, $|u_k^1 - u_k^2| = \xi$ and $\text{sign}(u_k^1 - u_k^i) = \text{sign}(u_k^2 - u_k^i)$ for any $i = 3, \dots, n$. By analogy, we have $|u_s^1 - u_s^2| = \xi$ or $u_s^1 - u_s^2 = 0$ for any $s \geq k$. Therefore, $\{|u_s^1 - u_s^2| : s \geq k\}$ fluctuates within a narrow range ξ, and we can claim that forward-stagewise clustering paths cannot split.

4. Noise and Outliers

Convex clustering can be guided by external information through appropriately defined weights $\{w_{ij} : 1 \leq i < j \leq n\}$, and hence is more robust to noise and outliers. For example, we can define data-dependent weights in the following manner:

$$w_{ij} = I(\boldsymbol{x}_j \text{ is among the k nearest neighbors of } \boldsymbol{x}_i \text{ or vice versa}) \\ \times \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2), \quad 1 \leq i < j \leq n.$$

The adaptive weights indicate that, if a sample $\boldsymbol{x}_i$ is far away from its neighbours, then w_{ij} will be small for all j $(1 \leq j \neq i \leq n)$; when minimizing (2), the centroid $\boldsymbol{u}_i$ will be distant from the other centroids.

We can further introduce adaptive weights into the 1st term in (2):

$$\min_{\boldsymbol{u} \in \mathbb{R}^n} \frac{1}{2}\|\boldsymbol{\Lambda}(\boldsymbol{x} - \boldsymbol{u})\|_2^2 + \lambda \sum_{i<j} w_{ij}|u_i - u_j| \\ = \min_{\boldsymbol{u} \in \mathbb{R}^n} \frac{1}{2}\|\tilde{\boldsymbol{x}} - \boldsymbol{\Lambda}\boldsymbol{u}\|_2^2 + \lambda \sum_{i<j} w_{ij}|u_i - u_j|, \tag{8}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{\Lambda}\boldsymbol{x}$. For example, $\boldsymbol{\Lambda}$ can be a diagonal matrix with the ith diagonal element being

$$\sum_{\substack{j=1,\dots,n \\ j \neq i}} \exp(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2) I(\boldsymbol{x}_j \text{ is among the k nearest neighbors of } \boldsymbol{x}_i).$$

If a sample $\boldsymbol{x}_i$ is far away from its neighbours, then the ith diagonal element will be large and force the centroid $\boldsymbol{u}_i$ being adjacent to $\boldsymbol{x}_i$.

The forward-stagewise clustering algorithm for problem (8) is still extremely simple. Particularly, it can be proved that the primal and dual solutions satisfy the relationship $\boldsymbol{u}_\lambda = (\boldsymbol{\Lambda}'\boldsymbol{\Lambda})^{+}(\boldsymbol{\Lambda}'\tilde{\boldsymbol{x}} - \boldsymbol{D}'\boldsymbol{\beta}_\lambda)$, where $(\boldsymbol{\Lambda}'\boldsymbol{\Lambda})^{+}$ denotes the Moore-Penrose inverse of $\boldsymbol{\Lambda}'\boldsymbol{\Lambda}$. If $\boldsymbol{\Lambda}$ is a diagonal matrix, then the Moore-Penrose inverse reduces to matrix inverse and can be quickly calculated. The algorithm for problem (8) is only one more line than Algorithm 2:

```
1: Calculate the Moore-Penrose inverse $\boldsymbol{M} = (\boldsymbol{\Lambda}'\boldsymbol{\Lambda})^{+}$.
2: Fix $\varepsilon > 0$.
3: Initialize $\boldsymbol{\beta}_0 = \boldsymbol{0}$ and $\boldsymbol{u}_0 = \boldsymbol{M}\boldsymbol{\Lambda}'\tilde{\boldsymbol{x}}$.
4: for $k = 1, 2, 3, \ldots,$ do
5:     $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \varepsilon \times \mathrm{sign}(\boldsymbol{D}\boldsymbol{u}_{k-1})$;
6:     $\boldsymbol{u}_k = \boldsymbol{M}(\boldsymbol{\Lambda}'\tilde{\boldsymbol{x}} - \boldsymbol{D}'\boldsymbol{\beta}_k)$;
7: end for
```

5. Numerical Experiments

We first give two simple clustering examples in two dimensions to demonstrate the feasibility of forward-stagewise clustering. The weights are defined as $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2)$, and ε is set to be 0.001. Data in Figure 1 are 150 points falling

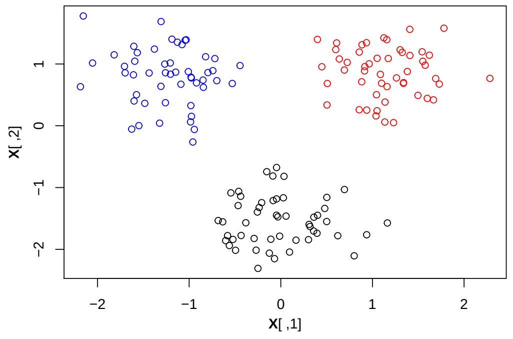

Fig. 1: Scatter plot of the simulated data, with each color representing one cluster.

in three clusters of 50 points each. The points are generated from three Gaussian distributions, with one distribution representing one cluster. Figure 2 plots the solution path for each dimension. Figure 3 represents the 3-D view of the solution path profile in which the x-axis and y-axis are for locating the data points, and the z-axis is for indicating the λ value. Figure 3 gives a clear view of the corresponding agglomerative tree structure and shows that the proposed algorithm can correctly identify the underlying three clusters.

Figures 4-6 illustrate another example where the data are in the form of two interlocking half-moons composed of 50 points each. Both the 2-D solution path plots in Figure 5 and the 3-D solution path plot in Figure 6 indicate that the proposed algorithm can correctly group the data into two clusters.

The preceding two examples give some additional insights into forward-stagewise clustering. Firstly, as is expected, there is no split in any of the solution paths. Therefore, each clustering path corresponds to an agglomerative tree. Secondly,

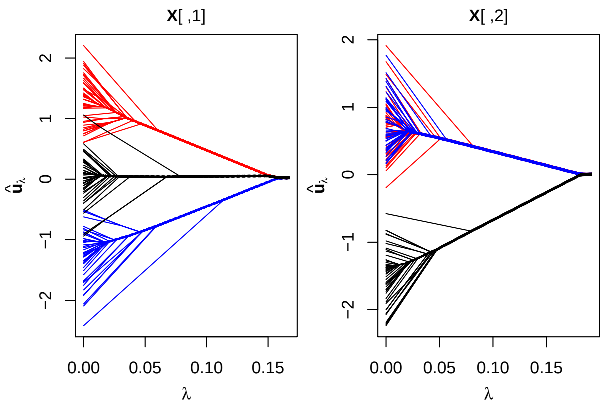

Fig. 2: Left: solution path for the first dimension. Right: solution path for the second dimension.

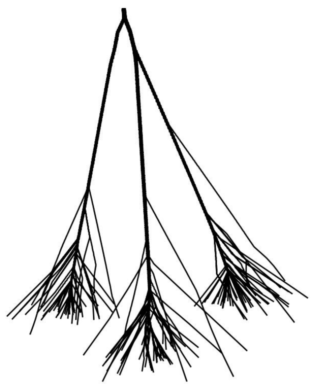

Fig. 3: 3-D plot of the solution path profile. With the λ value (vertical axis) increasing, all data points are grouped into one cluster.

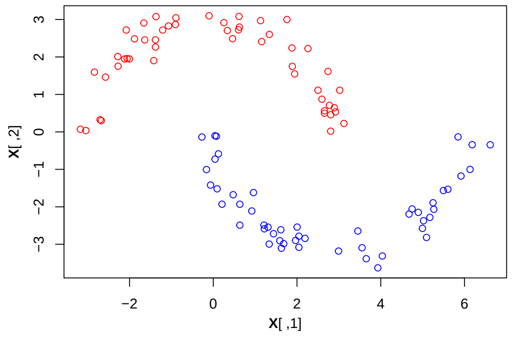

Fig. 4: Scatter plot of the simulated data, with each color representing one cluster.

if a cluster can be correctly identified (that is, no point from this cluster is wrongly grouped into another cluster), then there should exist at least one coordinate on which the projections of the data from this cluster are separated from the projects of the other data. For example, the left panel of Figure 2 shows that, on the first coordinate, the projections of the data from the red and blue clusters can be separated. The right panel of Figure 2 shows that, on the second coordinate, the projections of the data from the black cluster can be separated from the projections of the other data. Likewise, in the second example, Figure 5 shows that the red and blue clusters can be separated in terms

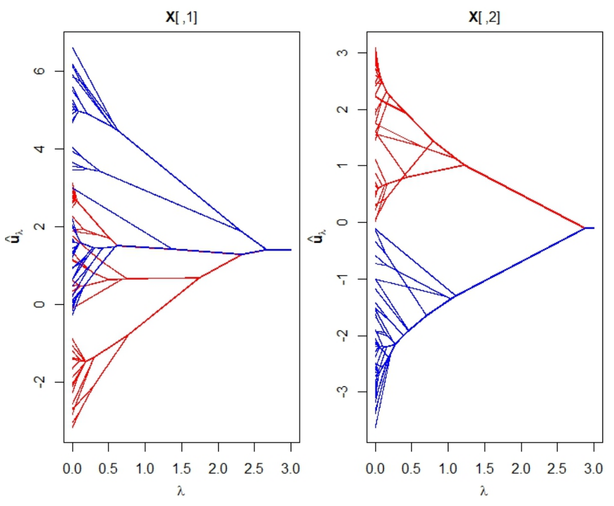

Fig. 5: Left: solution path for the first dimension. Right: solution path for the second dimension.

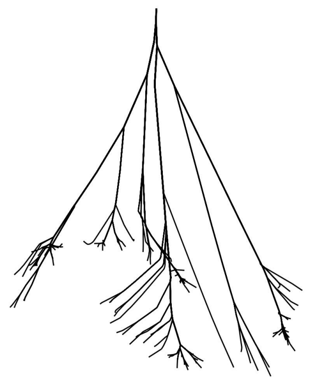

Fig. 6: 3-D plot of the solution path profile. With the λ value (vertical axis) increasing, all data points are grouped into one cluster.

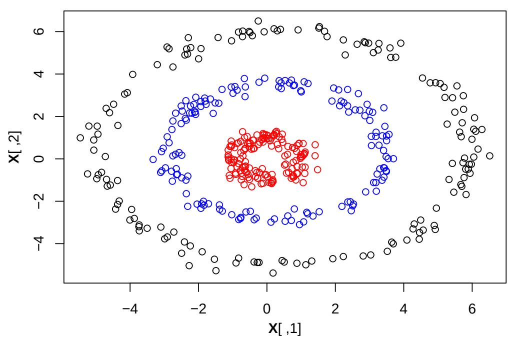

Fig. 7: Scatter plot of the simulated data, with each color representing one cluster.

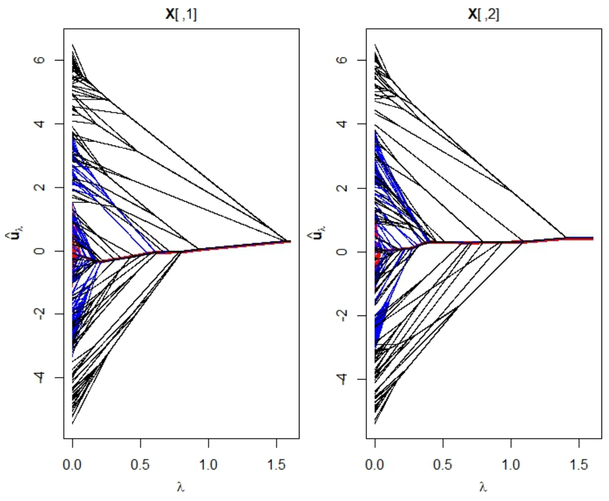

Fig. 8: Forward-stagewise clustering path when $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2)$.

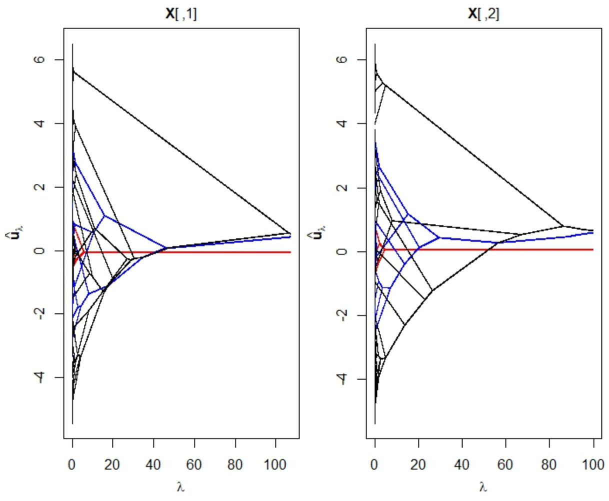

Fig. 9: Forward-stagewise clustering path when $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2) \times I(\boldsymbol{x}_j$ is among the 10 nearest neighbors of $\boldsymbol{x}_i$ or vice versa).

of the second coordinate. Consequently, in the above two examples, forward-stagewise clustering can successfully identify the underlying clusters before finally merging them.

In the case where there exists at least one cluster whose projection on any coordinate cannot be separated from the projects of the other clusters, we suggest taking either a data-level strategy or a graph-level strategy. The data-level strategy aims to project/transform the original data into a new space in which the clusters are separable. For example, as with spectral clustering, one can apply forward-stagewise clustering on the eigenvectors of the unnormalized graph Laplacian. The graph-level strategy attempts to control the clustering path through the edge set $\{(i,j): w_{ij} > 0\}$ of the graph G. Hocking et al. (2011) and Lindsten et al. (2011) point out that the choice of the weights can significantly affect the clustering path and propose to define the weights as $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2) \times I(\boldsymbol{x}_j$ is among the k nearest neighbors of $\boldsymbol{x}_i$ or vice versa), which means that a node (data point) in the graph G is directly connected only to its k nearest neighbors. If the graph contains several isolated communities (no edge connecting any two of the communities), then the points within each community will finally be grouped into one cluster, but the communities will not be merged. Figures 7-9 illustrate another example in which the data are 450 points falling in three non-convex clusters of 150 points each. The clustering path in Figure 8 corresponds to $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2)$, while the clustering path in Figure 9 corresponds to $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2) \times I(j$ is among the 10 nearest neighbors of i or vice versa). It is clear that the graph-level strategy makes forward-stagewise clustering successfully identify the underlying three clusters.

As a final example, we apply forward-stagewise clustering to the classical Iris data, which can be downloaded from the UCI machine learning repository (Dheeru and Karra Taniskidou, 2017). The dataset contains three classes of 50 instances each, where each class refers to a type of iris plant. The weights are defined as $w_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2) \times I(j$ is among the 5 nearest neighbors of i or vice versa$)$. Figure 10 shows the clustering path profile. It can be inferred from Figure 10 that the blue and black clusters (respectively representing Iris-versicolor and Iris-virginica) are non-separable w.r.t. any coordinate. If we set the number of clusters to be 3, then 14 Iris-virginica specimens will be misclassified as Iris-versicolor for an overall error rate of 14/150=0.093. The Rand index is 0.892, higher than the results obtained by Hocking et al. (2011).

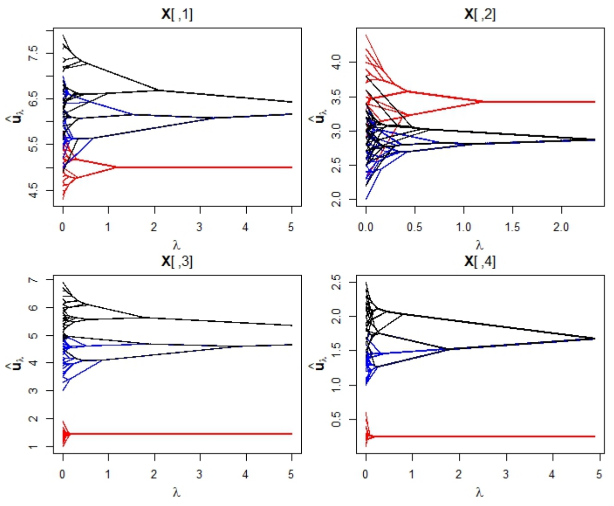

Fig. 10: Clustering of the Iris data. Each panel plots the clustering path for one coordinate.

6. Conclusion

This paper investigated the feasibility and competence of the forward-stagewise strategy for convex clustering. We proved that, under the ℓ_1 norm penalty, forward-stagewise solution paths cannot have any split, irrespective of the values of the weights $\{w_{ij}\}$. The numerical studies performed here further demonstrate that forward-stagewise clustering has exceptional performance in that it can correctly identify the underlying clusters. Moreover, two alternative approaches were suggested for forward-stagewise clustering to deal with the case when clusters are non-convex.

Forward-stagewise clustering can be readily modified to solve problems of the more general form: $\min_{\boldsymbol{u}\in\mathbb{R}^n} f(\boldsymbol{u}) + \lambda\|\boldsymbol{D}\boldsymbol{u}\|_1$, where $f(\boldsymbol{u})$ is formulated to address different issues (e.g., robust to outlier features). We leave to future work the investigation of the possibility of combining forward-stagewise clustering with real-time processing and the analysis of big data.

References

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 3, 1–122.

Chen, G.K., Chi, E.C., Ranola, J.M.O., Lange, K., 2015. Convex clustering: An attractive alternative to hierarchical clustering. PLOS Computational Biology 11, 1–31.

Chi, E.C., Lange, K., 2015. Splitting methods for convex clustering. Journal of Computational and Graphical Statistics 24, 994–1013.

Chiquet, J., Gutierrez, P., Rigaill, G., 2017. Fast tree inference with weighted fusion penalties. Journal of Computational and Graphical Statistics 26, 205–216.

Dheeru, D., Karra Taniskidou, E., 2017. UCI machine learning repository. URL: `http://archive.ics.uci.edu/ml`.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. The Annals of Statistics 32, 407–499.

Hallac, D., Leskovec, J., Boyd, S., 2015. Network lasso: Clustering and optimization in large graphs, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 387–396.

Han, L., Zhang, Y., 2016. Reduction techniques for graph-based convex clustering, in: 13th AAAI Conference on Artificial Intelligence.

Hastie, T., Taylor, J., Tibshirani, R., Walther, G., 2007. Forward stagewise regression and the monotone lasso. Electronic Journal of Statistics 1, 1–29.

Hocking, T.D., Joulin, A., Bach, F., Vert, J.P., 2011. Clusterpath: An algorithm for clustering using convex fusion penalties, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, pp. 745–752.

Lindsten, F., Ohlsson, H., Ljung, L., 2011. Just Relax and Come Clustering!: A Convexification of k-Means Clustering. Technical Report. Linkping University, The Institute of Technology.

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

Sui, X.L., Xu, L., Qian, X., Liu, T., 2018. Convex clustering with metric learning. Pattern Recognition 81, 575 – 584.

Tan, K.M., Witten, D., 2015. Statistical properties of convex clustering. Electronic journal of statistics 9, 23242347.

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

Tibshirani, R.J., 2015. A general framework for fast stagewise algorithms. Journal of Machine Learning Research 16, 2543–2588.

Tibshirani, R.J., Taylor, J., 2011. The solution path of the generalized lasso. The Annals of Statistics 39, 1335–1371.

Wang, Q., Gong, P., Chang, S., Huang, T.S., Zhou, J., 2016. Robust convex clustering analysis, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1263–1268.

Zhu, C., Xu, H., Leng, C., Yan, S., 2014. Convex optimization procedure for clustering: Theoretical revisit, in: Advances in Neural Information Processing Systems 27, pp. 1619–1627.

Zou, H., 2006. The adaptive lasso and its oracle properties. Journal of the American Statistical Association 101, 1418–1429.