Published in final edited form as:
J Comput Graph Stat. 2020 ; 29(1): 87–96. doi:10.1080/10618600.2019.1629943.

Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization

Michael Weylandt[1], **John Nagorski**[1], **Genevera I. Allen**[1,2,3,4]

[1]Department of Statistics, Rice University

[2]Department of Computer Science, Rice University

[3]Department of Electrical and Computer Engineering, Rice University

[4]Neurological Research Institute, Baylor College of Medicine

Abstract

Convex clustering is a promising new approach to the classical problem of clustering, combining strong performance in empirical studies with rigorous theoretical foundations. Despite these advantages, convex clustering has not been widely adopted, due to its computationally intensive nature and its lack of compelling visualizations. To address these impediments, we introduce Algorithmic Regularization, an innovative technique for obtaining high-quality estimates of regularization paths using an iterative one-step approximation scheme. We justify our approach with a novel theoretical result, guaranteeing global convergence of the approximate path to the exact solution under easily-checked non-data-dependent assumptions. The application of algorithmic regularization to convex clustering yields the Convex Clustering via Algorithmic Regularization Paths (CARP) algorithm for computing the clustering solution path. On example data sets from genomics and text analysis, CARP delivers over a 100-fold speed-up over existing methods, while attaining a finer approximation grid than standard methods. Furthermore, CARP enables improved visualization of clustering solutions: the fine solution grid returned by CARP can be used to construct a convex clustering-based dendrogram, as well as forming the basis of a dynamic path-wise visualization based on modern web technologies. Our methods are implemented in the open-source R package clustRviz, available at https://github.com/DataSlingers/clustRviz.

Keywords

Clustering; Convex Clustering; Optimization; Algorithmic Regularization; Visualization; Dendrograms

Supplementary Materials

Supplementary materials published alongside the online version of this article contain a more detailed derivation of Algorithms 1 and 2, a detailed proof of Theorem 1, a discussion of our CBASS algorithm for convex bi-clustering, numerical comparisons of CARP to existing clustering approaches on real and synthetic data sets, details about the back-tracking, post-processing, and dendrogram strategies used in our CARP-VIZ algorithm and clustRviz software, and a discussion of additional related work.

1 Introduction

Clustering, the task of identifying meaningful sub-populations in unlabelled data, is a fundamental problem in applied statistics, with applications as varied as cancer subtyping, market segmentation, and topic modeling of text documents. A wide range of methods for clustering have been proposed and we do not attempt to make a full accounting here, instead referring the reader to the recent book of Hennig et al. (2015). Perhaps the most popular clustering method, however, is hierarchical clustering (Ward, 1963). Hierarchical clustering derives its popularity from an intuitive formulation, efficient computation, and powerful visualizations. Dendrogram plots, which display the family of clustering solutions simultaneously, provide an easily-understood summary of the global structure of the data, allowing the analyst to visually examine the nested group structure of the data. Despite its popularity, hierarchical clustering has several limitations: it is highly sensitive to the choice of distance metric and linkage used; it is a heuristic algorithm which lacks optimality guarantees; and the conditions under which hierarchical clustering recovers the true clustering are unknown.

To address these limitations, several authors have recently studied a convex formulation of clustering (Pelckmans et al., 2005; Hocking et al., 2011; Lindsten et al., 2011; Chi and Lange, 2015). This convex formulation provides guarantees of global optimality of a clustering solution and allows analysis of its theoretical properties (Tan and Witten, 2015; Zhu et al., 2014; Radchenko and Mukherjee, 2017; Chi and Steinerberger, 2018). Despite these advantages, convex clustering has not yet achieved widespread popularity, due to its computationally intensive nature and lack of dendrogram-based visualizations. In this paper, we address these problems with a efficient algorithm for computing convex clustering solutions with sufficient precision to construct interpretable accurate dendrograms and dynamic path-wise visualizations, thereby making convex clustering a practical tool for applied data analysis.

Our main theoretical contribution is the concept of Algorithmic Regularization, a novel computationally efficient approach for obtaining high-quality approximation of regularization paths. We provide a theoretical justification for our proposed approach, showing that we can obtain a high-quality approximation simultaneously at all values of the regularization parameter. While we focus on the convex clustering problem, our proposed approach can be applied to a much wider range of problems arising in statistical learning.

Using algorithmic regularization, we make two methodological contributions related to clustering: first, we propose an efficient algorithm, CARP, for computing convex clustering solutions, CARP is typically over one-hundred times faster than existing approaches, while simultaneously computing a much finer set of solutions than commonly used in practice. Secondly, we propose new visualization strategies for convex clustering based on CARP: a new dendrogram construction based on convex clustering paths and a novel “ path-wise” visualization, which provides more information about the structure of the estimated clusters. We hope that, thanks to our proposed computational and visualization strategies, convex clustering will become a viable tool for exploratory data analysis, CARP and our proposed visualizations are implemented in our clustRviz R package, available at https://github.com/DataSlingers/clustRviz.

The remainder of this paper is organized as follows: Section 2 reviews convex clustering in more detail and discusses the difficulties entailed in producing dendrograms from convex clustering. Section 3 introduces the concept of "Algorithmic Regularization," uses it to develop the CARP clustering algorithm, and gives theoretical guarantees of global convergence. Section 4 compares CARP with existing approaches for convex clustering, demonstrating its impressive computational and statistical performance on several data sets. Section 5 describes several novel visualizations made possible by the CARP algorithms in the context of an extended text-mining example. Finally, Section 6 concludes the paper with a discussion and proposes possible future directions for investigation.

2 Convex Clustering and Dendrograms

We seek to represent the convex clustering solution path as a dendrogram, and in this section, discuss both the theoretical conditions and computational considerations for this task. We first review the basic properties of convex clustering in Section 2.1 and then discuss dendrogram construction from convex clustering in Section 2.2.

2.1 Convex Clustering

Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ denote a data matrix, consisting of n observations (rows of the matrix) in p dimensions. The convex clustering problem, first discussed by Pelckmans et al. (2005) and later explored by Hocking et al. (2011) and Lindsten et al. (2011), is a convex relaxation of the general clustering problem:

$$\underset{\boldsymbol{U} \in \mathbb{R}^{n \times p}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\|_F^2 \quad \text{subject to} \quad \sum_{1 \leq i < j \leq n} 1_{\boldsymbol{U}_{i\cdot} \neq \boldsymbol{U}_{j\cdot}} \leq t,$$

where $\boldsymbol{U}_{i\cdot}$ is the i^{th} row of $\boldsymbol{U}$. Replacing the non-convex indicator function with an ℓ_q-norm of the difference, we obtain the convex clustering problem:

$$\boldsymbol{U}_\lambda = \underset{\boldsymbol{U} \in \mathbb{R}^{n \times p}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\|_F^2 + \lambda\left(\sum_{1 \leq i < j \leq n} w_{ij}\|\boldsymbol{U}_{i\cdot} - \boldsymbol{U}_{j\cdot}\|_q\right). \tag{1}$$

Note that we have included non-negative fusion weights $\{w_{ij}\}$ in our convex relaxation. We will say more about the computational and statistical roles played by these weights below.

The squared Frobenius norm loss function favors solutions which minimize the Euclidean distance between observations and their estimated centroids, while the fusion penalty term, $p(\boldsymbol{U}; \boldsymbol{w}, q)$, encourages the differences in columns of $\boldsymbol{U}_\lambda$ to be shrunk to zero. We interpret the solution $\boldsymbol{U}_\lambda$ as a matrix of cluster centroids, where each observation $\boldsymbol{X}_{i\cdot}$ belongs to a cluster with centroid $(\boldsymbol{U}_\lambda)_{i\cdot}$. For sufficiently large values of λ, the columns of $\boldsymbol{U}_\lambda$ will be shrunk together by the penalty term. We say that points with the same centroid belong to the same cluster; that is, the observations $\boldsymbol{X}_{i\cdot}$ and $\boldsymbol{X}_{j\cdot}$ are assigned to the same cluster if $(\boldsymbol{U}_\lambda)_{i\cdot} = (\boldsymbol{U}_\lambda)_{j\cdot}$.

A major advantage of convex clustering is that the solution $\boldsymbol{U}_\lambda$ smoothly interpolates between clustering solutions, yielding a continuous path of solutions indexed by λ. At $\lambda = 0$, $\boldsymbol{U}_{\lambda=0} = \boldsymbol{X}$, resulting in a solution of n distinct clusters, with each observation as the centroid of its own cluster. As λ is increased, the fusion penalty encourages the columns of $\boldsymbol{U}_\lambda$ to merge together, inducing a clustering behavior. Finally, when λ is large, all columns of $\boldsymbol{U}_\lambda$ are fully merged, yielding a single cluster centroid equal to the grand mean of the columns of X. Thus, the penalty parameter λ determines both the number of clusters and the cluster assignments.

The choice of the fusion weights $\boldsymbol{w} \in \mathbb{R}^{\binom{n}{2}}_{\geq 0}$ has a large effect on the statistical accuracy and computational efficiency of convex clustering. When uniform weights are used, convex clustering has a close connection to single-linkage hierarchical clustering, as shown by Tan and Witten (2015). More commonly, weights inversely proportional to the distances between observations are used, and have been empirically demonstrated to yield superior performance (Hocking et al., 2011; Chi and Lange, 2015; Chi et al., 2017). Furthermore, setting many of the weights to zero dramatically reduces the computational cost associated with computing the convex clustering solution. We typically prefer using the rotation-invariant ℓ_2-norm in the fusion penalty ($q = 2$), but one could employ ℓ_1 or ℓ_∞ norms as well.

By formulating clustering as a convex problem, it becomes possible to analyze its theoretical properties using standard techniques from the highdimensional statistics literature. Tan and Witten (2015) show a form of prediction consistency and derive an unbiased estimator of the effective degrees of freedom associated with the solution. Zhu et al. (2014) give sufficient conditions for exact cluster recovery at a fixed value of λ (“sparsistency”). Like us, Radchenko and Mukherjee (2017) are interested in properties of the entire solution path and give conditions under which convex clustering solutions (1) asymptotically yield the true dendrogram.

2.2 Constructing Dendrograms from Convex Clustering Paths

In this paper, we propose to represent the convex clustering solution path as a dendrogram, an example of which is shown in Figure 1. The convex clustering dendrogram is interpreted in much the same way as the classical dendrogram. As individual observations or groups of observations are fused together by the fusion penalty, they are denoted by merges in the tree structure. The height of the merge in the tree structure is given by the value of the regularization parameter, λ, or more precisely $\log(\lambda)$, at which the fusion occurred. Thus, observations that fuse at small values of λ are denoted by merges at the bottom of the dendrogram structure. As with hierarchical clustering, one can cut the dendrogram horizontally at a specific height to yield the associated clusters, and we can interpret the tree height at which merges occur as indicative of the similarity between groups. Before proceeding, however, it is natural to ask whether it is even possible to represent the convex clustering solution path as a dendrogram. This simple question turns out to have a rather subtle answer.

There are two possible impediments to finding the desired dendrogram representation: i) it may be impossible to represent the exact solution path as a dendrogram; and ii) it may be unrealistic to compute the solution path with enough precision to form a dendrogram. We first consider the question of whether the exact solution path admits a dendrogram representation. It is easy to observe that the exact solution path can only be written as a dendrogram if it is agglomerative, *i.e.*, if the solution path consists of only fusions and no fissions. Hocking et al. (2011) showed that if an ℓ_1-norm is used for the fusion penalty and the weights are uniform then the solution path is agglomerative, but their analysis does not generalize to arbitrary norms or arbitrary weight schemes. Chi and Steinerberger (2018) showed that it is possible to select weights to yield a agglomerative path, but their analysis applies only to a specific weighting scheme. In general, for an arbitrary data-driven choice of weights, there is no theoretical guarantee that the solution path will be agglomerative. In our experience, however, the solution path is agglomerative except in pathological situations.

Assuming that the exact solution path is agglomerative, we still must determine whether the solution path, *as calculated*, is agglomerative. While, in theory, this poses no additional challenge as the solution path is a continuous function of λ, in practice this poses a nearly insurmountable computational challenge. To construct a dendrogram, we require the exact values of λ at which each fusion occurs. Since these values of λ are not known *a priori*, we are faced with a double burden: we must identify a critical set of values of λ and solve the convex clustering problem (1) at those values. There are two widely-used approaches to finding the critical set of λ's, path-wise algorithms and grid search, but, as we will show below, neither approach is sufficient in this case.

Path-wise algorithms, such as those proposed for the Lasso (Osborne et al., 2000; Efron et al., 2004) or generalized Lasso (Tibshirani and Taylor, 2011) problems, compute the entire solution path exactly, identifying each value of λ at which a sparsity event, equivalent to a fusion in our case, occurs. These algorithms are typically based on piece-wise linearity of the underlying solution path, which allows for smooth interpolation between sparsity events. Rosset and Zhu (2007) studied the conditions under which solution paths are piece-wise linear and hence under which path-wise algorithms can be developed. It is easy to verify that these conditions do not hold with our preferred ℓ_2-norm fusion penalty, so the solution path is not piece-wise linear and hence a path-wise algorithm cannot be employed for the convex clustering problem (1). We note that several authors have proposed path-wise algorithms which can theoretically handle non-piece-wise-linear paths, but require solving an ordinary differential equation exactly (Wu, 2011; Zhou and Wu, 2014; Xiao et al., 2015); in practice, these methods are computationally intensive and only approximate the path at a series of grid points, similar to the iterative methods we discuss next. For the ℓ_1-norm fusion case, it is possible to apply path-wise algorithms for weights with specific graphical structures (see the discussion in Hocking et al. (2011) or the examples considered in Tibshirani and Taylor (2011)), but not for arbitrary graph structures with arbitrary weights, again eliminating the possibility of a general-purpose path-wise algorithm.

Since there exists no exact path-wise algorithm, we might instead compute the convex clustering solution at a series of discrete points corresponding to a regular grid of λ's. As we will show in Section 4, however, this strategy is still computationally burdensome even using state-of-the-art algorithms and warm-start techniques. For example, the fastest algorithm considered by Chi and Lange (2015), an Accelerated Alternating Minimization Algorithm, takes 6.87 hours and 19.81 hours to compute the solution path at 100 and 1000 regularly spaced λ's, respectively, on a relatively small data set of dimension $n = 438$ and $p = 353$. Furthermore, a grid of 100 or 1000 λ's does not give us the value of λ at which each fusion event occurs, which we need to construct a dendrogram. Even if one wants to construct a dendrogram using the order in which each fusion event occurs and their associated approximate values of λ, computing the path along a grid of 100 or 1000 λ's only uniquely resolves 11.4% or 37.44% respectively of the fusion events needed to construct a dendrogram. (See Table 1 in Section 4 for complete results.)

In general, the computational cost of performing convex clustering is so high that it precludes its use as a practical tool for clustering and exploratory data analysis. Further, computing the entire convex clustering solution path with fine enough precision to construct a dendrogram is an all but insurmountable task given existing computational algorithms for convex clustering. We seek to address this problem in this paper, using a novel computational technique which provides a fine grid of high-quality estimates of the regularization path. We introduce our approach and the clustering algorithm it suggests, CARP, in the next section.

3 CARP: Convex Clustering via Algorithmic Regularization Paths

We now turn our attention to efficiently computing solutions to the convex clustering problem (1) for a fine grid of λ, with a goal of dendrogram construction. Like many problems in the "loss + penalty" form, the convex clustering problem is particularly amenable to operator-splitting schemes such as the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). In statistical learning, we are often interested in the solution to a regularized estimation problem at a large number of values of λ. In this context, the performance of ADMMs is increased further by the use of "warm-starts:" if the ADMM is initialized near the solution, usually the solution at the previous value of λ, only a few iterations are typically required to obtain a solution which is accurate up to the statistical uncertainty inherent in the problem.

We propose an extreme version of this approach which we call *Algorithmic Regularization*. Instead of running the ADMM to convergence, we take only a single ADMM step, after which we move to the next value of λ. By taking only a single ADMM step, we can significantly reduce the computational cost associated with estimating a regularization path. We can then use these computational savings to solve for a much finer grid of λ's than we would typically use if employing a standard scheme. In essence, Algorithmic Regularization allows us to exchange computing an exact solution for a small set of λ's for calculating a highly accurate approximation at a large set of λ's. Usefully, we can now use a λ grid with sufficiently fine resolution that we can fully capture the desired dendrogram structure in a reasonable amount of time.

Chi and Lange (2015) first considered the use of the ADMM to solve the convex clustering problem (1). To apply the ADMM, we introduce $\boldsymbol{D}$, the directed difference matrix used to calculate to the pairwise differences of rows of $\boldsymbol{U}$, and an auxiliary variable $\boldsymbol{V}$, corresponding to the matrix of between-observation differences:

$$\underset{\substack{\boldsymbol{U} \in \mathbb{R}^{n \times p} \\ \boldsymbol{V} \in \mathbb{R}^{\binom{n}{2} \times p}}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{U}\|_F^2 + \lambda \underbrace{\sum_{k=1}^{\binom{n}{2}} w_k \|\boldsymbol{V}_{k.}\|_q}_{P(\boldsymbol{V}; \boldsymbol{w}, q)} \text{subject to } \boldsymbol{V} = \boldsymbol{D}\boldsymbol{U}.$$

Applying the ADMM with warm-starts to the above, we obtain the following algorithm:

Algorithm 1 Warm-Started ADMM for the Convex Clustering Problem (1)

```
Initialize $l = 0$, $\lambda_l = \epsilon$, $\boldsymbol{V}^{(0)} = \boldsymbol{Z}^{(0)} = \boldsymbol{D}\boldsymbol{X}$
Repeat until $\|\boldsymbol{V}^{(k)}\| = 0$:
    • Repeat until convergence:
        (i) $\boldsymbol{U}^{(K+1)} = \boldsymbol{L}^{-T} L^{-1} (\boldsymbol{X} + \boldsymbol{D}^T (\boldsymbol{V}^{(k)} - \boldsymbol{Z}^{(k)}))$
        (ii) $\boldsymbol{V}^{(k+1)} = \text{prox}_{\lambda_l p(\cdot; \boldsymbol{w}, q)} (\boldsymbol{D}\boldsymbol{U}^{(k+1)} + \boldsymbol{Z}^{(k)})$
        (iii) $\boldsymbol{Z}^{(k+1)} = \boldsymbol{Z}^{(k)} + \boldsymbol{D}\boldsymbol{U}^{(k+1)} - \boldsymbol{V}^{(k+1)}$
        (iv) $k := k + 1$
    • Store $\boldsymbol{U}_{\lambda_l} = \boldsymbol{U}^{(k)}$
    • Update regularization: $l := l + 1$ ; $\lambda_l := \lambda_{l-1} * t$
Return $\{\boldsymbol{U}_\lambda\}$ as the regularization path
```

where $\boldsymbol{Z}$ is the dual variable with the same dimensions as $\boldsymbol{V}$, $\boldsymbol{L}$ is Cholesky factorization of $\boldsymbol{I} + \boldsymbol{D}^T \boldsymbol{D}$, and $\text{prox}_{f(\cdot)}(\boldsymbol{x}) = \text{argmin}_{\boldsymbol{z}} \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + f(\boldsymbol{z})$ is the proximal mapping of a general function f. Note that, if sparse weights are used, the corresponding rows of $\boldsymbol{D}$,$\boldsymbol{V}$, and $\boldsymbol{Z}$ may be omitted, yielding more efficient updates. Additionally, note that, because we use a multiplicative update for λ, we must initialize at $\lambda = \epsilon$, for some small ϵ, rather than at $\lambda = 0$. A derivation and more detailed statement of this algorithm are given in Section A of the Supplementary Materials.

We now take Algorithm 1 as the basis for our extreme early stopping strategy of Algorithmic Regularization. Removing the the inner loop, we obtain the following scheme, which we refer to as CARP–**C**onvex Clustering via **A**lgorithmic **R**egularization **P**aths:

Algorithm 2 CARP: Algorithmic Regularization for the Convex Clustering Problem (1)

```
Initialize $k = 0$, $\gamma^{(k)} = \epsilon$, $\boldsymbol{V}^{(0)} = \boldsymbol{Z}^{(0)} = \boldsymbol{D}\boldsymbol{X}$
Repeat until $\|\boldsymbol{V}^{(k)}\| = 0$ :
    (i) $\boldsymbol{U}^{(k+1)} = \boldsymbol{L}^{-T} L^{-1} (\boldsymbol{X} + \boldsymbol{D}^T (\boldsymbol{V}^{(k)} - \boldsymbol{Z}^{(k)}))$
```

Algorithm 2 CARP: Algorithmic Regularization for the Convex Clustering Problem (1)

```
    (ii) $\boldsymbol{V}^{(k+1)} = \text{prox}_{\gamma^{(k)}\text{p}(\cdot;\boldsymbol{w},q)}\,(\boldsymbol{DU}^{(k+1)} + \boldsymbol{Z}^{(k)})$
    (iii) $\boldsymbol{Z}^{(k+1)} = \boldsymbol{Z}^{(k)} + \boldsymbol{DU}^{(k+1)} - \boldsymbol{V}^{(k+1)}$
    (iv) $k := k + 1,\ \gamma^{(k)} = \gamma^{(k-1)} * t$
Return $\{\boldsymbol{U}^{(k)}\}$ as the CARP path.
```

The fundamental difference between Algorithm 1 and Algorithm 2 is that Algorithm 2 does not have an “inner loop” in which ADMM iterates are repeated until convergence to the exact solution for a fixed value of the regularization parameter. As such, the CARP iterates $\{\boldsymbol{U}^{(k)}\}$ are not exact solutions to the convex clustering problem (1) for any value of λ, though they are typically accurate approximations in a sense that Theorem 1 below makes precise. Our notation reflects this distinction and replaces λ_l with $\gamma^{(K)}$ in the $\boldsymbol{V}$-update to avoid suggesting any false equivalence. A more detailed formulation of the CARP algorithm is given in Section A of the Supplementary Materials.

The role of the step-size parameter t in Algorithm 2 is particularly important in understanding CARP. The step size t controls the fineness of the $\{\gamma^{(K)}\}$ grid used internally by CARP and, as such, serves as a *computational* tuning parameter controlling how well the CARP path approximates the true convex clustering path. Decreasing t therefore has benefits for both *local* and *global* accuracy of the CARP path: a smaller value of t yields an approximate solution path which has a finer set of grid points $\{\gamma^{(K)}\}$ (improved global accuracy) and more accurate approximations $\{\boldsymbol{U}^{(k)}\}$ at each of those grid points (improved local accuracy). This is in contrast to standard approaches where the user has to pre-specify the $\{\lambda_l\}$ grid used and the stopping tolerance of the iterative algorithm to strike a balance between local accuracy and global accuracy. As we will see in Section 4, the Algorithmic Regularization strategy of replacing an iterative algorithm with a one-step approximation thereof allows us to improve both local and global accuracy at a fraction of the cost of competing methods.

While this may all seem rather fishy, the following theorem shows that, in the limit of small changes to the regularization level (*i.e.*, $(t,\epsilon)\rightarrow(1,0)$), there is indeed no loss in accuracy induced by the one-step approximation. In fact, we are able to show a very strong form of convergence, so-called *Hausdorff* convergence, in both the primal and dual variables. Hausdorff convergence implies two different convergence results hold simultaneously for both the primal and dual variables. The first, $\sup_{\lambda}\inf_{k}\left\|\boldsymbol{U}^{(k)} - \boldsymbol{U}_{\lambda}\right\| \rightarrow 0$, implies every convex clustering solution will be recovered by CARP as $(t,\epsilon)\rightarrow(1,0)$. The second, $\sup_{k}\inf_{\lambda}\left\|\boldsymbol{U}^{(k)} - \boldsymbol{U}_{\lambda}\right\| \rightarrow 0$, implies that any clustering produced by CARP as $(t,\epsilon)\rightarrow(1,0)$ is a valid convex clustering solution for some λ. More memorably, Theorem 1 shows that asymptotically CARP produces “the whole regularization path and nothing but the regularization path:”

Theorem 1. *As* $(t,\epsilon)\rightarrow(1,0)$ *, where t is the multiplicative step-size update and* ϵ *is the initial regularization level, the primal and dual* CARP *paths converge to the primal and dual convex clustering paths in the Hausdorff metric: that is,*

$$d_H(\{\boldsymbol{U}^{(k)}\},\{\boldsymbol{U}_\lambda\}) \equiv \max\left\{\sup_\lambda \inf_k \left\|\boldsymbol{U}^{(k)}-\boldsymbol{U}_\lambda\right\|, \sup_k \inf_\lambda \left\|\boldsymbol{U}^{(k)}-\boldsymbol{U}_\lambda\right\|\right\} \xrightarrow{(t,\epsilon)\rightarrow(1,0)} 0$$

$$d_H(\{\boldsymbol{Z}^{(k)}\},\{\boldsymbol{Z}_\lambda\}) \equiv \max\left\{\sup_\lambda \inf_k \left\|\boldsymbol{Z}^{(k)}-\boldsymbol{Z}_\lambda\right\|, \sup_k \inf_\lambda \left\|\boldsymbol{Z}^{(k)}-\boldsymbol{Z}_\lambda\right\|\right\} \xrightarrow{(t,\epsilon)\rightarrow(1,0)} 0$$

where $\boldsymbol{U}^{(k)}$,$\boldsymbol{Z}^{(k)}$ *are the values of the* K^{th} CARP *iterate and* $\boldsymbol{U}_\lambda$, $\boldsymbol{Z}_\lambda$ *are the exact solutions to the convex clustering problem (1) and its dual at* λ.

A full proof of Theorem 1 is given in Section B of the Supplementary Materials, but we highlight the three essential elements here: i) we obtain a high-quality initialization at the first step by setting $\boldsymbol{U}^{(0)}=\boldsymbol{X}$ which is the exact solution at $\lambda=0$ ($\boldsymbol{U}_{\lambda=0}=\boldsymbol{X}$); ii) the convex clustering problem (1) is strongly convex due to the squared Frobenius norm loss, so the ADMM converges quickly (linearly); and iii) the solution path is Lipschitz as a function of λ, so $\boldsymbol{U}^{\lambda}$ does not vary too quickly. Putting these together, we show that CARP can "track" the exact solution path closely, with the approximation error at each step decreasing at a faster rate than the exact solution changes. We emphasize that both strong convexity and Lipschitz solution paths are features of the optimization problem, not the specific data, and are easily checked in practice. A careful reading of the proof of Theorem 1 will reveal that our analysis applies to a much wider class of problems than convex clustering. We consider the application of algorithmic regularization to the closely related problem of convex bi-clustering (Chi et al., 2017) in Section C of the Supplementary Materials, where we develop the CBASS (**C**onvex **B**i-Clustering via **A**lgorithmic Regularization with **S**mall **S**teps) algorithm, but we leave examination of the more general phenomenon of Algorithmic Regularization to future work.

While Theorem 1 implies that a sufficiently small choice of step size t allows for exact dendrogram recovery, in practice it is often challenging to select t sufficiently small without requiring excessive computation. Instead, we take a small, but not infinitesimal, value of t and add a back-tracking step to ensure that fusions necessary for dendrogram construction are exactly identified. We refer to the back-tracking version of CARP as CARP–VIZ, for reasons which will be clarified in Section 4. Furthermore, a post-processing step can be used to find fusions that back-tracking is unable to isolate. Details of the back-tracking and post-processing rules, as implemented in clustRviz, are given in Section E of the Supplementary Materials.

Algorithmic Regularization, as used here, was first discussed in Hu et al. (2016) without theoretical justification and was successfully applied to unmixing problems in hyperspectral imaging by Drumetz et al. (2017). We emphasize that, while grounded in standard optimization techniques, algorithmic regularization takes a different perspective than other commonly-used computational approaches, more akin to function approximation than standard optimization. The algorithmic regularization perspective is principally concerned with recovering the overall structure of the solution path than with obtaining the most accurate solution possible at a fixed value of λ. As such, our Theorem 1 is of a different character than similar results appearing in the optimization literature, making a claim of global path-wise convergence rather than local point-wise convergence. This "holistic" viewpoint is necessary to recover dendrograms, a major goal of this paper, but has also recently been found useful for choosing tuning parameters in penalized regression problems (Chichignoud et al., 2016).

3.1 Related Work

Several authors have considered path approximation algorithms not unlike CARP, often in the context of boosting algorithms. Rosset et al. (2004), Zhao and Yu (2007), and Friedman (2012) all consider iterative algorithms which approximate solution paths of regularized estimators. Of these, the approach of Zhao and Yu (2007), who consider a path approximation algorithm for the Lasso, is most similar to our own. Assuming strong convexity, their BLasso algorithm exactly recovers the lasso solution path as the step-size goes to zero. Their algorithm can be viewed as an application of algorithmic regularization to greedy coordinate descent with an additional back-tracking step to help isolate events of interest (variables entering or leaving the active set). Our algorithmic regularization strategy is simpler than their approach, as it does not require the back-tracking step, and can be applied to more general penalty functions.

Clarkson (2010) and Giesen et al. (2012) consider the problem of obtaining approximate solutions for a set of parameterized problems subject to a simplex constraint, though their approach still requires running an optimization step until approximate convergence at each step, Building on this, Tibshirani (2015) proposes a general framework for constructing "stagewise" solution paths, which can be interpreted as an application of algorithmic regularization to the Frank-Wolfe algorithm (Jaggi, 2013). He shows that the stagewise estimators achieve the optimal objective value as the step-size is taken to zero; if a strong convexity assumption is added, it is not difficult to extend his Theorem 2 to recover a result similar to our Theorem 1. Our framework is more general than his, as we do not require require the gradient of the loss function to be Lipschitz and we admit more general regularizers.

4 Numerical and Timing Comparisons

Having introduced CARP and given some theoretical justification for its use, we now consider its performance on representative data sets from text analysis and genomics. As we will show, CARP achieves the superior clustering performance of convex clustering at a small fraction of the computational cost. Throughout this section, we use two example data sets: TCGA and Authors. The TCGA data set ($n = 438$, $p = 353$) contains log-transformed Level III RPKM gene expression levels for 438 breast-cancer patients from The Cancer Genome Atlas Network (2012). The Authors data set ($n = 841$, $p = 69$) consists of word counts from texts written by four popular English-language authors (Austen, London, Shakespeare, and Milton). For all comparisons, we use clustRviz's default sparse Gaussian kernel weighting scheme described in the package documentation. For timing comparisons, the Accelerated ADMM and AMA proposed by Chi and Lange (2015) and implemented in their cvxclustr package was used; our clustRviz package was used for CARP and CARP-VIZ. All comparisons were run on a 2013 iMac with a 3.2 GHz Intel i5 processor and 16 GB of 1600 MHz DDR3 memory.

While Theorem 1 strictly only applies for asymptotically small values of t, CARP paths are high-quality approximations of the exact convex clustering solution path, even at moderate values of t. We assess accuracy of the CARP paths by considering the normalized relative Hausdorff distance between the primal CARP path ($\boldsymbol{U}^{(k)}$) and the exact solution ($\boldsymbol{U}_\lambda$)

$$d_H(\{\boldsymbol{U}^{(k)}\}, \{\boldsymbol{U}_\lambda\}) = \frac{\max\left\{\sup_\lambda \inf_k \left\|\boldsymbol{U}^{(k)} - \boldsymbol{U}_\lambda\right\|_2, \sup_k \inf_\lambda \left\|\boldsymbol{U}^{(k)} - \boldsymbol{U}_\lambda\right\|_2\right\}}{n^* p^* \|\boldsymbol{D}\boldsymbol{X}\|_{2,\infty}}$$

where $\|\cdot\|_{2,\infty}$ is the maximum of the ℓ_2-norms of the rows of a matrix. (We include the normalization constants in the denominator so that our distance measure does not depend on the size or numerical scale of the data.) In order to calculate the Hausdorff distance, the CARP path with a very small step-size t was used in lieu of the exact solution $\{\boldsymbol{U}_\lambda\}$. As can be seen in Figure 2, the CARP path is highly accurate even at moderate values of t and converges quickly to the exact solution as $t \to 1$. CARP–VIZ, which uses an adaptive choice of t to isolate each individual fusion, performs even better than the fixed step-size CARP, attaining a very accurate approximation of the true clustering path.

Even though they are highly accurate, CARP paths are relatively cheap to compute. In Figure 3, we compare the computational cost of CARP with the algorithms proposed in Chi and Lange (2015). As shown in Figure 3, CARP significantly outperforms the Accelerated AMA and ADMM algorithms. At large step-sizes (t = 1.1,1.05), CARP terminates in less than a minute and, even at finer grid-sizes (t = 1.01,1.005), CARP takes only a few minutes to run. The CARP-VIZ variant takes significantly longer than standard CARP, though it still outperforms the AMA, taking about an hour for TCGA, rather than the six and a half hours required to solve the AMA at 100 grid points. This improvement in computational performance is even more remarkable when we note that CARP and CARP-VIZ produce a fine grid of solutions by default: on the TCGA data, CARP with t = 1.01 produces over 2047 distinct grid points in under five minutes.

The fine grid of solutions returned by CARP and CARP-VIZ result in much improved dendrogram recovery, as measured by the fraction of unique clustering assignments each method returns. Table 1 shows recovery results for CARP (at several values of t), CARP–VIZ, and standard fixed-grid methods. It is clear that back-tracking employed by CARP–VIZ is necessary for exact dendrogram recovery and that CARP–VIZ should be used if the exact dendrogram recovery is required for visualization. Even with moderate step-sizes (t = 1.1,1.05), however, CARP is still able to estimate the dendrogram far more accurately and more rapidly than standard iterative methods, making it a useful alternative for exploratory work.

On the TCGA data, CARP with $t = 1.05$ is able to recover the dendrogram with the same accuracy in a minute that the AMA attains in six and a half hours. With $t = 1.01$, CARP recovers the dendrogram more accurately in under five minutes than the AMA does in nineteen hours (1000 grid points), CARP achieves these improvements by using its computation efficiently: while a standard optimization algorithm may spend several hundred iterations at a single value of the regularization parameter, CARP only spends a single iteration. By reducing the number of iterations at each grid point, it can take examine a much finer grid in less time. This trade-off is particularly well-suited for our goal of dendrogram recovery, which requires a fine grid of solutions to assess the order of fusions, but does not depend on the exact values of the estimated centroids. While not a primary focus of this paper, the high-quality dendrogram estimation allowed by CARP translates into improved statistical performance as well. We compare the statistical performance of CARP with other clustering methods in Section D of the Supplementary Materials.

5 Visualization of CARP Results

In this section, we discuss visualization of convex clustering results, emphasizing the role that CARP can play in exploratory data analysis. The visualizations illustrated in this section can all be produced using our clustRviz R package. Throughout this section, we will use the Presidents data set, ($n = 44$, $p = 75$) which contains log-transformed word counts of the 75 most variable words taken from the aggregated major speeches (primarily Inaugural and State of the Union Addresses) of the 44 U.S. presidents through mid-2018. (We consolidate the two non-consecutive terms of Grover Cleveland.)

We begin by considering a dendrogram representation of this data, as shown in Figure 4. For each dendrogram, we have colored the observations by historical period: Founding Fathers, pre-Civil War, pre-World War II, and modern. Given the evolution of the English language and the changing political concerns of these periods, we would expect clustering methods to group the presidents according to historical period. With three exceptions, CARP clearly identifies the four historical periods, with the modern period being particularly well-separated. The performance of hierarchical clustering is highly sensitive to the choice of linkage: Ward's linkage (Ward, 1963) does almost as well as CARP, but does not clearly separate the pre-Civil War and pre-World War II periods. Single linkage correctly identifies the modern period, but otherwise does not separate the pre-modern presidents. Complete linkage performs the worst, clustering Donald Trump with the Founding Fathers, Garfield, and Harrison instead of with other modern presidents. We note that Harrison is consistently misclustered by all methods considered: we believe this is due to the fact he died thirty-one days into his first term and did not leave a lengthy textual record.

Beyond allowing accurate dendrogram construction, the CARP paths are themselves interesting to visualize. By plotting the path traced by the CARP iterates $\boldsymbol{U}^{(k)}$, we can observe exactly how CARP forms clusters from a given data set. For dimension reduction, we typically plot the projection of $\boldsymbol{U}^{(k)}$ onto the principal components of $\boldsymbol{X}$, though clustRviz allows visualization of the raw features as well. Unlike the CARP-dendrogram, the path plot allows examination of the structure of the estimated clusters and not just their membership. By displaying the original observations on the path plot, we also enable comparison of the estimated centroids with the original data. Modern web technologies allow us to display these path plots dynamically, forming a movie with each CARP iterate as a separate frame. The fine solution grid returned by CARP is especially relevant here, enabling us to construct movies in which the observations move smoothly. We have found that the smoothness of the movie is a useful heuristic to asses whether a small enough step-size t was used: if the paths "jump" conspicuously from one frame to the next, one should consider re-running CARP with a smaller step-size.

Figure 5 shows three frames of such a movie: in each frame, on the left side, we see the Founding Fathers cluster being merged to the other pre-modern presidents, while on the right, we see a clear cluster of modern presidents. A closer examination of the central frame reveals additional information not visible in the CARP-dendrogram: Harding, the last president to join the pre-modern cluster, is an outlier lying between two clusters rather than far to one side.

Because both the dendrogram and path visualizations are indexed by the regularization level, $\gamma^{(k)}$, it is possible to display them in a "linked" fashion, highlighting clusterings on the dendrogram as they are fused in the path plot. Particularly when rendered dynamically, this combination gives the best of both visualizations, combining the global structure visible in the dendrogram with the structural information visible in the path plot. An example of this "linked" visualization is shown in Figure 6.

6 Discussion

We have introduced Algorithmic Regularization, an iterative one-step approximation scheme which can be used to efficiently obtain high-quality approximations of regularization paths. Algorithmic regularization focuses on accurate reconstruction of an entire regularization path and is particularly useful when for obtaining path-wise information, such as a dendrogram or the order in which variables leave the active set in sparse regression. We have focused on the application of the ADMM to convex clustering, but the technique of iterative one step-approximations can be applied to any problem which lacks an efficient algorithm. We believe that algorithmic regularization can be fruitfully applied to a broader range of statistical learning problems and expect that similar computational improvements can be achieved for other difficult optimization problems.

Theorem 1 is a novel *global* convergence result, guaranteeing high-quality approximation at each point of the exact solution path. Despite this, there are still many open questions in the analysis of algorithmic regularization. We are particularly interested in determining optimal convergence rates for global path-wise approximation problems and showing that algorithmic regularization can attain those rates. Our proof of Theorem 1 depends on the strong convexity of the convex clustering problem to ensure linear convergence of the underlying ADMM steps. It would be interesting to explore the interplay between algorithmic regularization and optimization schemes which are linearly convergent without strong convexity, as this may extend the applicability of algorithmic regularization even further.

Using algorithmic regularization, we have introduced the CARP and CBASS algorithms for convex clustering and bi-clustering. On moderately sized problems, CARP and CBASS reduce the time necessary to obtain high-quality regularization paths from several hours to only a few minutes, typically attaining over one-hundred-fold improvements over existing algorithms. Because CARP and CBASS return solutions at a fine grid of the regularization parameter, they can be used to construct accurate convex (bi-)clustering dendrograms, particularly if the back-tracking CARP–VIZ and CBASS–VIZ variants are employed. Additionally, the fine-grained CARP and CBASS solution paths allow for path-wise dynamic visualizations, allowing the analyst to observe exactly how the estimated clusters are formed and structured.

We anticipate that the computational and visualization techniques proposed in this paper will make convex clustering and bi-clustering an attractive option for applied data analysis. Both CARP and CBASS, as well as the proposed visualizations, are implemented in our clustRviz software, available at https://github.com/DataSlingers/clustRviz.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

We thank Eric Chi for helpful discussions about both convex clustering and algorithmic regularization. MW acknowledges support from the NSF Graduate Research Fellowship Program under grant number 1450681. GA acknowledges support from NSF DMS-1554821, NSF NeuroNex-1707400, and NSF DMS-1264058. JN acknowledges support from NSF DMS-124058, NSF DMS-1554821, and the National Institutes of Health National Cancer Institute T32 Training program in Biostatistics for Cancer Research, Grant Number: CA096520.

MW and JN jointly developed the clustRviz software. MW is responsible for the content and proof of Theorem 1 and prepared the manuscript. JN performed initial experiments and developed the back-tracking and post-processing schemes. GA supervised the research and edited the final manuscript.

References

Boyd Stephen, Parikh Neal, Chu Eric, Peleato Borja, and Eckstein Jonathan (2011). “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. Foundations and Trends® in Machine Learning 31, pp. 1–122. doi: 10.1561/2200000016.

Chi Eric C., Allen Genevera I., and Baraniuk Richard G. (2017). “Convex Biclustering”. Biometrics 731, pp. 10–19. doi: 10.1111/biom.12540. [PubMed: 27163413]

Chi Eric C. and Lange Kenneth (2015). “Splitting Methods for Convex Clustering”. Journal of Computational and Graphical Statistics 244, pp. 994–1013. doi: 10.1080/10618600.2014.948181. [PubMed: 27087770]

Chi Eric C. and Steinerberger Stefan (2018). “Recovering Trees with Convex Clustering”. ArXiv Pre-Print 1806.11096. url: https://arxiv.org/abs/1806.11096.

Chichignoud Michael, Lederer Johannes, and Wainwright Martin J. (2016). “A Practical Scheme and Fast Algorithm to Tune the Lasso With Optimality Guarantees”. Journal of Machine Learning Research 17231, pp. 1–20. url: http://jmlr.org/papers/v17/15-605.html.

Clarkson Kenneth L. (2010). “Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm”. ACM Transactions on Algorithms (TALG) 64, 63:1–63:30. doi: 10.1145/1824777.1824783.

Drumetz L, Tochon G, Veganzones MA, Chanussot J, and Jutten C (2017). “Improved Local Spectral Unmixing of Hyperspectral Data using an Algorithmic Regularization Path for Collaborative Sparse Regression”. ICASSP 2017: Proceedings of the 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing Ed. by Adali Tulay and Saber Elia. New Orleans, Louisiana: IEEE, pp. 6190–6194. doi:10.1109/ICASSP.2017.7953346.

Efron Bradley, Hastie Trevor, Johnstone lain, and Tibshirani Robert (2004). "Least Angle Regression". Annals of Statistics 322, pp. 407–451. doi:10.1214/009053604000000067.

Friedman Jerome H. (2012). "Fast Sparse Regression and Classification". International Journal of Forecasting 283, pp. 722–738. doi:10.1016/j.ijforecast.2012.05.001.

Giesen Joachim, Jaggi Martin, and Laue Sören (2012). "Approximating Parameterized Convex Optimization Problems". ACM Transactions on Algorithms (TALG) 91, 10:1–10:17. doi: 10.1145/2390176.2390186.

Hennig Christian, Meila Marina, Murtagh Fionn, and Rocci Roberto, eds. (2015). Handbook of Cluster Analysis Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, isbn: 978-1-466-55188-6. doi: 10.1201/b19706.

Hocking Toby Dylan, Joulin Armand, Bach Francis, and Vert Jean-Philippe (2011). "Clusterpath: An Algorithm for Clustering using Convex Fusion Penalties". ICML 2011: Proceedings of the 28th International Conference on Machine Learning Ed. by Getoor Lise and Scheffer Tobias. Bellevue, Washington, USA: ACM, pp. 745–752. isbn: 978-1-4503-0619-5. url: http://www.icml-2011.org/papers/419_icmlpaper.pdf.

Hu Yue, Chi Eric C., and Allen Genevera I. (2016). "ADMM Algorithmic Regularization Paths for Sparse Statistical Machine Learning". Splitting Methods in Communication and imaging, Science, and Engineering. Ed. by Glowinski Roland, Osher Stanley J., and Yin Wotao. Springer Chap. 13, pp. 433–449. doi: 10.1007/978-3-319-41589-5_13.

Jaggi Martin (2013). "Revisiting Frank-Wolfe: Projection-Free Convex Optimization". ICML 2013: Proceedings of the 30th International Conference on Machine Learning Ed. by Dasgupta Sanjoy and McAllester David. Atlanta, Georgia: PMLR, pp. 427–435. url: http://proceedings.mlr.press/v28/jaggi13.html.

Lindsten Fredrik, Ohlsson Henrik, and Ljung Lennart (2011). "Clustering using sum-of-norms regularization: With application to particle filter output computation". SSP 2011: Proceedings of the 2011 IEEE Statistical Signal Processing Workshop Ed. by Djuric Petar M.. Nice, France: Curran Associates, Inc., pp. 201–204. doi: 10.1109/SSP.2011.5967659.

Osborne Michael R., Presnell Brett, and Turlach Berwin A. (2000). "On the LASSO and its Dual". Journal of Computational and Graphical Statistics 92, pp. 319–337. doi: 10.1080/10618600.2000.10474883.

Pelckmans Kristiaan, de Brabanter Joseph, de Moor Bart, and Suykens Johan (2005). "Convex Clustering Shrinkage". PASCAL Workshop on Statistics and Optimization of Clustering.

Radchenko Peter and Mukherjee Gourab (2017). "Convex Clustering via `1 Fusion Penalization". Journal of the Royal Statistical Society, Series B: Statistical Methodology 795, pp. 1527–1546. doi: 10.1111/rssb.12226.

Rosset Saharon and Zhu Ji (2007). "Piecewise Linear Regularized Solution Paths". Annals of Statistics 353, pp. 1012–1030. doi:10.1214/009053606000001370.

Rosset Saharon, Zhu Ji, and Hastie Trevor (2004). "Boosting as a Regularized Path to a Maximum Margin Classifier". Journal of Machine Learning Research 5, pp. 941–973. url: http://www.jmlr.org/papers/v5/rosset04a.html.

Tan Kean Ming and Witten Daniela (2015). "Statistical Properties of Convex Clustering". Electronic Journal of Statistics 92, pp. 2324–2347. doi: 10.1214/15-EJS1074. [PubMed: 27617051]

The Cancer Genome Atlas Network (2012). "Comprehensive Molecular Portraits of Human Breast Tumours". Nature490, pp. 61–70. doi: 10.1038/nature11412. [PubMed: 23000897]

Tibshirani Ryan J. (2015). "A General Framework for Fast Stagewise Algorithms". Journal of Machine Learning Research 16, pp. 2543–2588. url: http://www.jmlr.org/papers/v16/tibshirani15a.html.

Tibshirani Ryan J. and Taylor Jonathan (2011). "The Solution Path of the Generalized Lasso". Annals of Statistics 393, pp. 1335–1371. doi:10.1214/11-AOS878.

Ward Jr., Joe H. (1963). "Hierarchical Grouping to Optimize an Objective Function". Journal of the American Statistical Association 58301, pp. 236–244. doi: 10.1080/01621459.1963.10500845.

Wu Yichao (2011). “An Ordinary Differential Equation-Based Solution Path Algorithm”. Journal of Nonparametric Statistics 231, pp. 185–199. doi:10.1080/10485252.2010.490584. [PubMed: 21532936]

Xiao Wei, Wu Yichao, and Zhou Hua (2015). “ConvexLAR: An Extension of Least Angle Regression”. Journal of Computational and Graphical Statistics 243, pp. 603–626. doi: 10.1080/10618600.2014.962700. [PubMed: 27114697]

Zhao Peng and Yu Bin (2007). “Stagewise Lasso”. Journal of Machine Learning Research 8, pp. 2701–2726. url: http://www.jmlr.org/papers/v8/zhao07a.html.

Zhou Hua and Wu Yichao (2014). “A Generic Path Algorithm for Regularized Statistical Estimation”. Journal of the American Statistical Association 109506, pp. 686–699. doi: 10.1080/01621459.2013.864166. [PubMed: 25242834]

Zhu Changbo, Xu Huan, Leng Chenlei, and Yan Shuicheng (2014). “Convex Optimization Procedure for Clustering: Theoretical Revisit”. NIPS 2014: Advances in Neural Information Processing Systems 27 Ed. by Ghahramani Zoubin, Welling Max, Cortes Corinna, Lawrence Neil D., and Weinberger Killian Q.. Montréal, Canada: Curran Associates, Inc., pp. 1619–1627. url: https://papers.nips.cc/paper/5307-convex-optimization-procedure-for-clustering-theoretical-revisit.

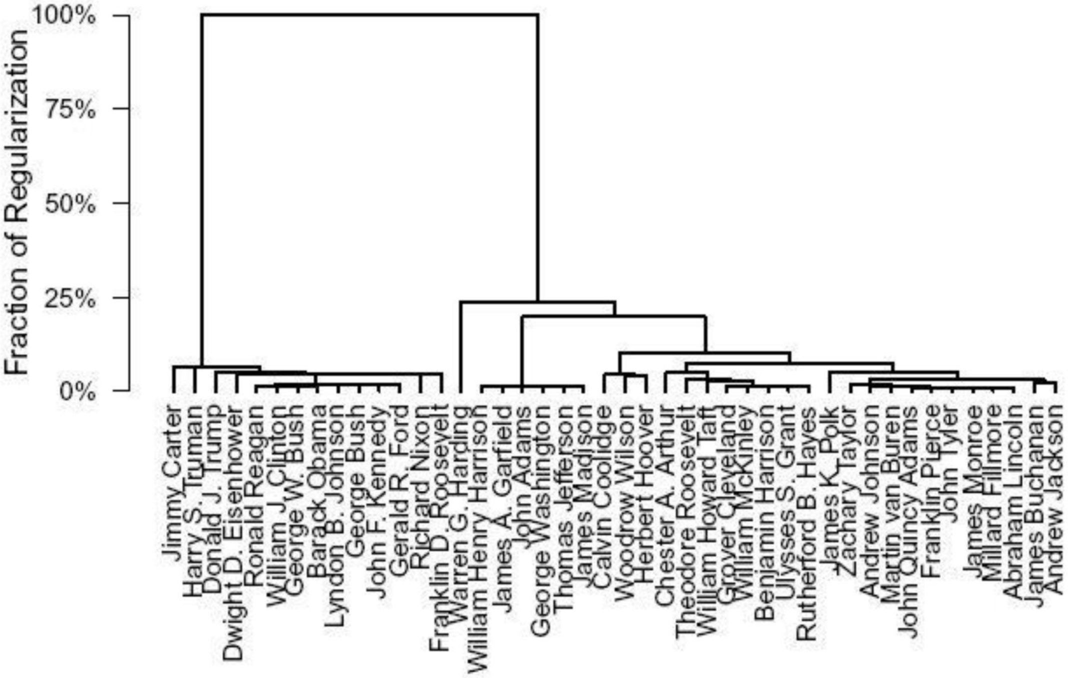

Fig. 1.

A convex clustering dendrogram, displaying the 44 U.S. presidents. The interpretation of this dendrogram is discussed in more detail in Section 5.

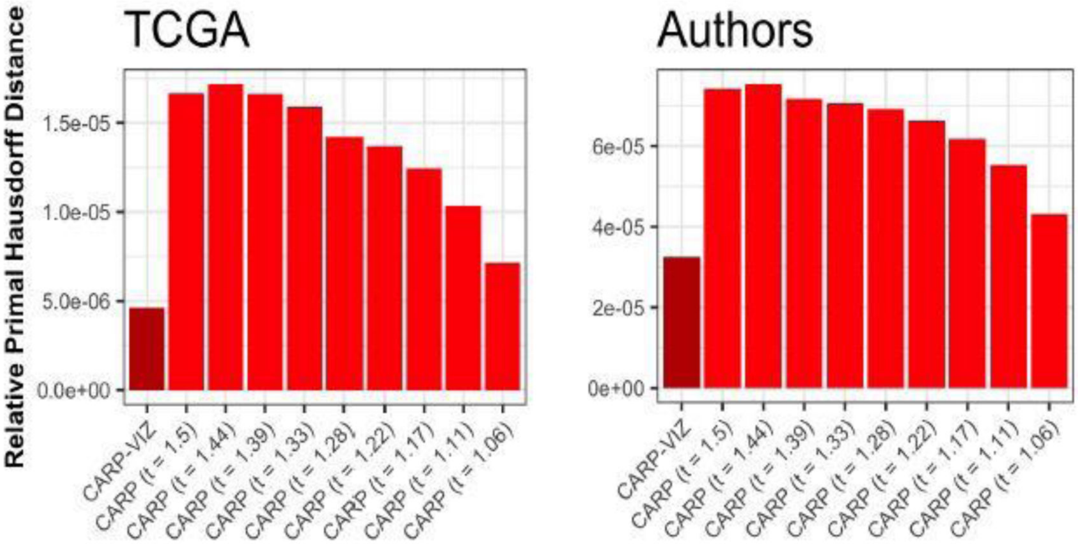

Fig. 2.

Normalized Relative Primal Hausdorff Distance between the exact convex clustering solution and CARP Paths for various values of *t*. As *t* decreases, the CARP Paths converge to the exact convex clustering solution path, consistent with Theorem 1.

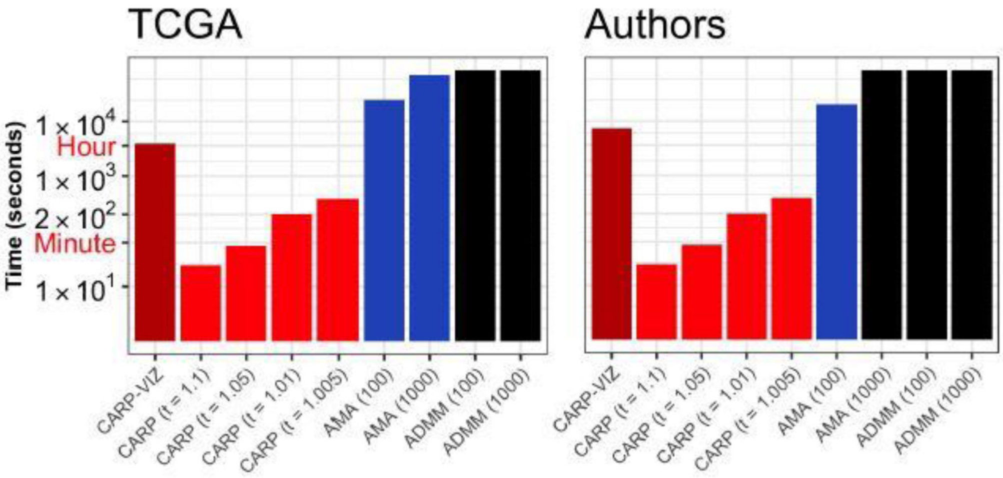

Fig. 3.

Time required to compute clustering solution path (logarithmic scale). CARP produces high-quality path approximations in a fraction of the time of standard iterative algorithms. Timings in black indicate calculations that took more than 24 hours to complete.

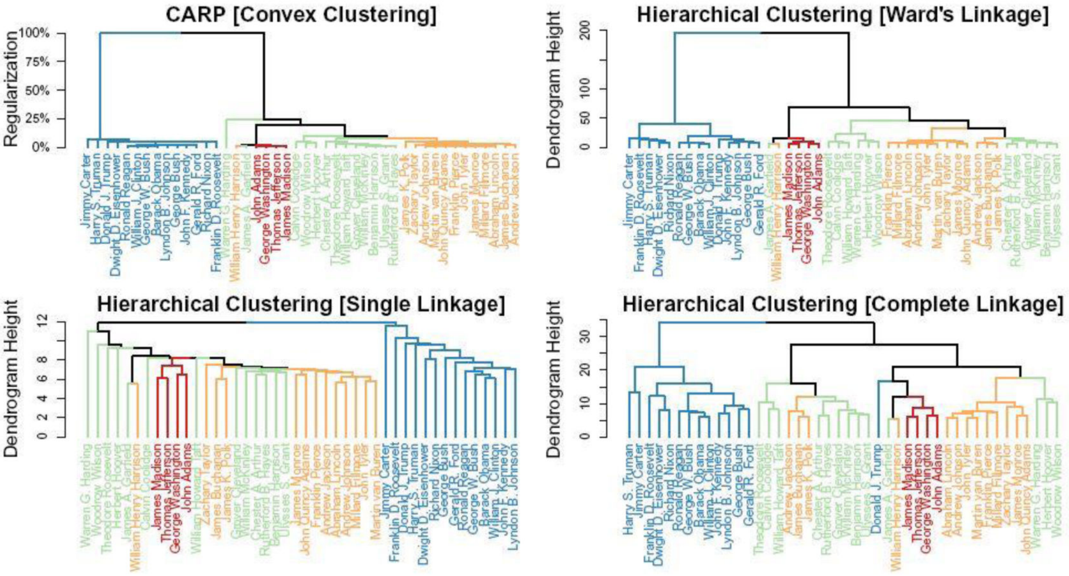

Fig. 4.

Comparison of CARP (top left) and Euclidean-distance Hierarchical Clustering dendrograms on the data. The presidents are colored according to historical period: Founding Fathers (pink, 1789–1817, Washington to Madison, *n* = 4); pre-Civil War (gold, 1817-1869, Monroe to Johnson, *n* = 13); pre-World War II (teal, 1869-1933, Grant to Hoover, *n* = 13); and modern (purple, 1933-present, F.D. Roosevelt to Trump, *n* = 14). We consider Johnson to be a pre-Civil War president as he ascended to the presidency following the assassination of Lincoln rather than being directly elected.

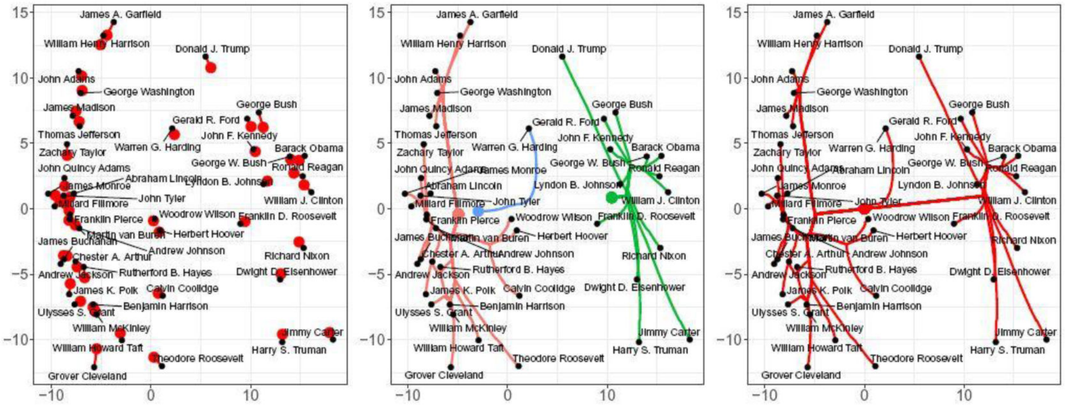

Fig. 5.

Direct visualization of the solution paths produced by CARP on the Presidents data, corresponding to unclustered (left), partially clustered (middle), and fully clustered (right) solutions. In each panel, the clusters of pre-modern and modern presidents are clearly visible, as is the outlier status of Harding.

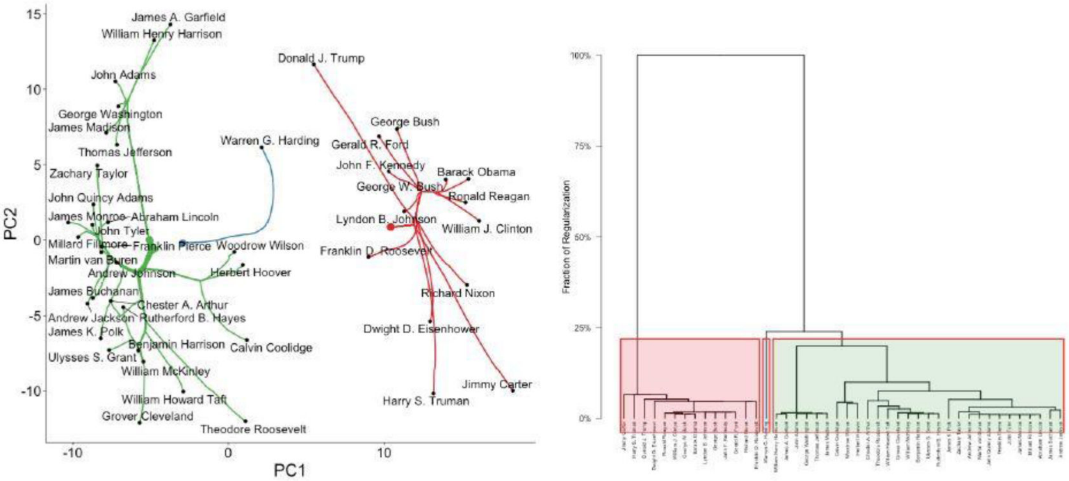

Fig. 6.

Linked simultaneous visualization of the CARP dendrogram and path plots. As clusters are formed in the path plot (left), they are highlighted on the dendrogram (right). The clusters of pre-modern and modern presidents are clearly visible, as is the outlier status of Warren G. Harding.

Table 1

Proportion of dendrogram recovered by CARP, CARP–VIZ, and standard fixed-grid methods. The back-tracking employed by CARP–VIZ is necessary for exact dendrogram recovery, but fixed step-size CARP is still able to recover the dendrogram more accurately than standard fixed-grid approaches.

Method	**TCGA (n = 438, p = 353)**	**Authors (n = 841, p = 69)**
CARP (t = 1.1)	5.93%	4.40%
CARP (t = 1.05)	11.18%	8.09%
CARP (t = 1.01)	41.55%	22.71%
CARP (t = 1.005)	60.27%	30.56%
CARP–VIZ	**100%**	**100%**
100-Point Grid	11.42%	5.00%
1000-Point Grid	37.44%	—