Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization

Michael Weylandt

To Appear in the *Journal of Computational and Graphical Statistics*
JSM 2019: 2019-07-30 at 10:30am in CC-704

Department of Statistics, Rice University, Houston, TX USA

Acknowledgements

US NSF:

- GRFP-1842494, NeuroNex-1707400,
- DMS-1554821, 1264058

Co-Authors:

- John Nagorski (Rice)
- Genevera Allen (Rice)

Methods available in the `clustRviz` R package

github.com/DataSlingers/clustRviz

Paper to appear in *J. Computational and Graphical Statistics* (2019+)

Clustering

Clustering: identifying sub-populations in unlabelled data

Clustering

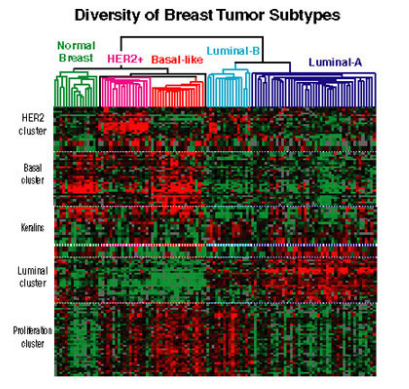
Clustering: identifying sub-populations in unlabelled data

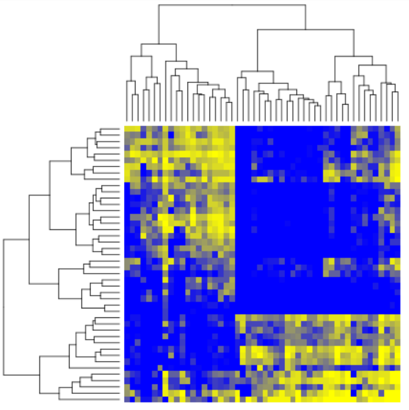
Example: breast cancer sub-typing & precision medicine

Clustering

Clustering: identifying sub-populations in unlabelled data

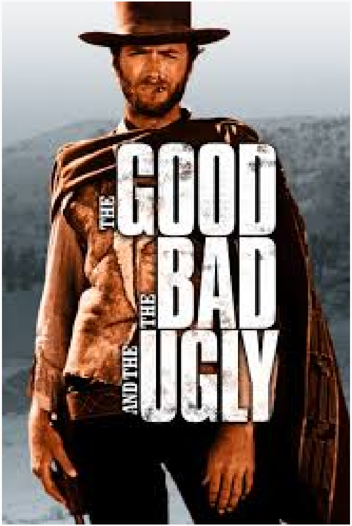
Example: breast cancer sub-typing & precision medicine





Clustering: The Good, The Bad, and The Ugly

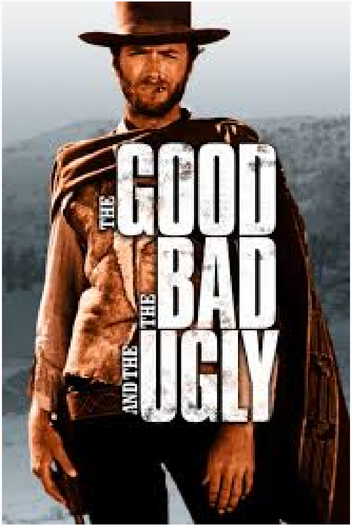
Existing methods for clustering:



Clustering: The Good, The Bad, and The Ugly

Existing methods for clustering:

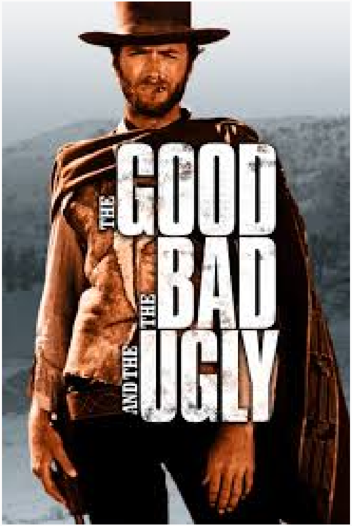
- K-Means
  - Good: Fast
  - Bad: Non-Convex
  - Ugly: How many clusters?



Clustering: The Good, The Bad, and The Ugly

Existing methods for clustering:

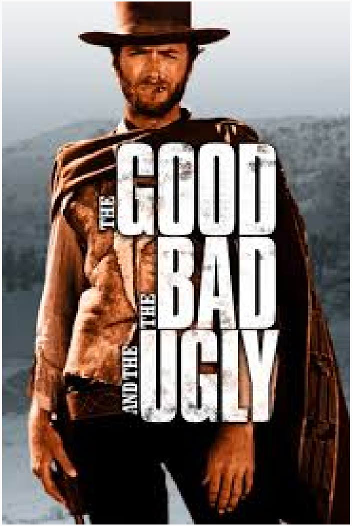
- K-Means
  - Good: Fast
  - Bad: Non-Convex
  - Ugly: How many clusters?
- Hierarchical Clustering
  - Good: Fast, nice visualizations
  - Bad: Many variants
  - Ugly: How many clusters?



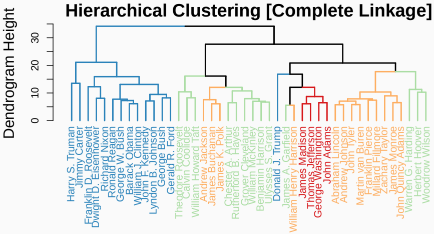
Clustering: The Good, The Bad, and The Ugly

Existing methods for clustering:

- K-Means
  - Good: Fast
  - Bad: Non-Convex
  - Ugly: How many clusters?
- Hierarchical Clustering
  - Good: Fast, nice visualizations
  - Bad: Many variants
  - Ugly: How many clusters?
- Others: spectral clustering, GMM+EM, DBSCAN, *etc.*



Consider the Humble Dendrogram



Dendrograms:

- Easily-understood, ubiquitous
- Show multiple clusterings simultaneously
- Give a sense of separation (ordinate)

Convex Clustering

Convex clustering (Hocking *et al.* 2011; Lindsten *et al.* 2011; Pelckmans *et al.* 2005):

$$\hat{\mathbf{U}} = \underset{\mathbf{U} \in \mathbb{R}^{n \times p}}{\arg\min} \frac{1}{2} \|\mathbf{X} - \mathbf{U}\|_F^2 + \lambda \sum_{\substack{i,j=1 \\ i \neq j}}^{n} w_{ij} \|\mathbf{U}_{i\cdot} - \mathbf{U}_{j\cdot}\|_q$$

Observations are clustered together if $\hat{\mathbf{U}}_{i\cdot} = \hat{\mathbf{U}}_{j\cdot}$

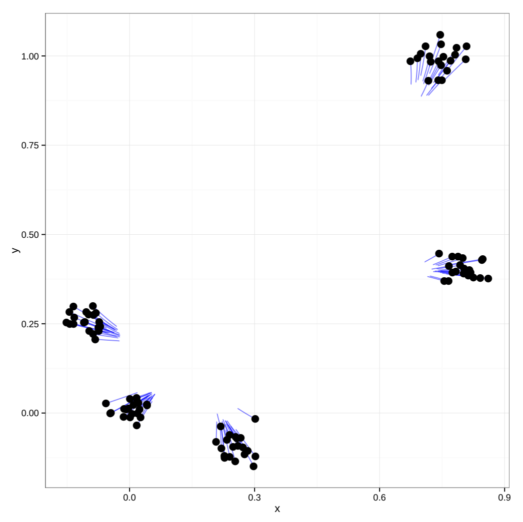
Estimated centroids $\hat{\mathbf{U}}$ are close to original data and fused together

Convexity implies:

- Global optimality + efficient algorithms
- Good statistical properties

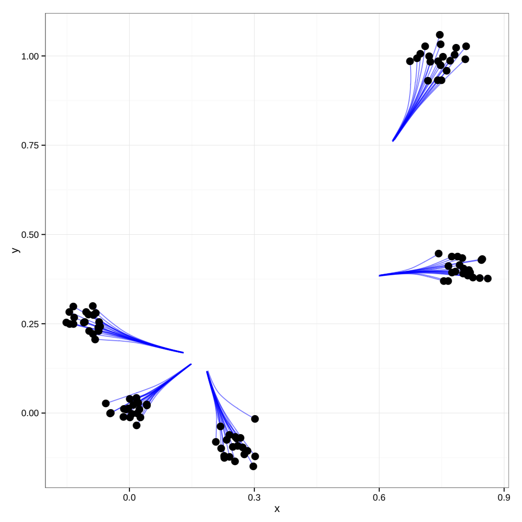
λ controls number of clusters smoothly

Convex Clustering: Solution Path



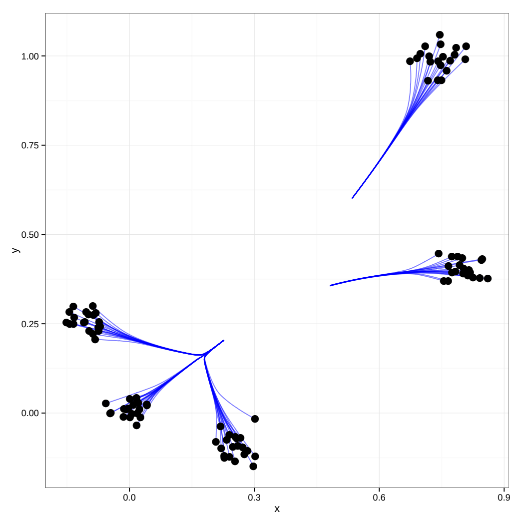
λ

Convex Clustering: Solution Path



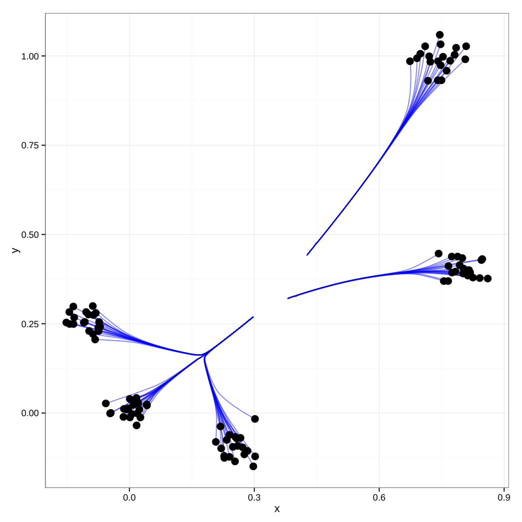
λ

Convex Clustering: Solution Path



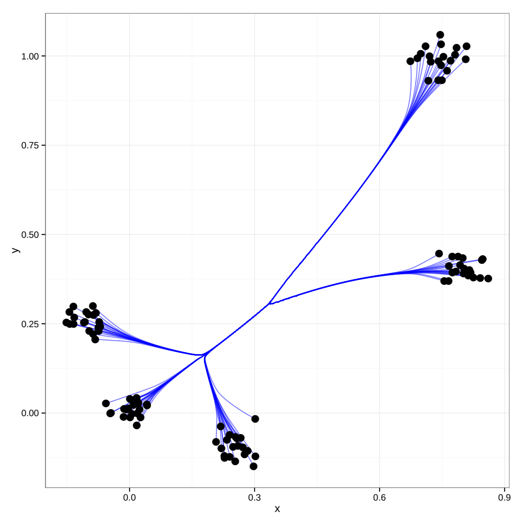
λ

Convex Clustering: Solution Path



λ

Convex Clustering: Solution Path



λ

Convex Clustering: Related Work

Related Work:

- Basic Framework: Hocking *et al.* (2011), Lindsten *et al.* (2011), and Pelckmans *et al.* (2005)
- Algorithms: Chen *et al.* (2015), Chi and Lange (2015), Ho *et al.* (2019), Panahi *et al.* (2017), and Sun *et al.* (2018)
- Two-Way Matrix / Bi-Clustering: Chi *et al.* (2017) and Weylandt (2019)
- Multi-Way Tensor / Co-Clustering: Chi *et al.* (2018)
- Consistency: Panahi *et al.* (2017), Radchenko and Mukherjee (2017), Tan and Witten (2015), and Zhu *et al.* (2014)
- Non-Convex Penalties: Marchetti and Zhou (2014), Pan *et al.* (2013), Shah and Koltun (2017), and Wu *et al.* (2016)
- Robustness: Wang *et al.* (2016)
- Feature Selection: Wang *et al.* (2018)
- Generalized Losses: Wang and Allen (2019+)

Convex Clustering: Related Work

Related Work:

- Basic Framework: Hocking *et al.* (2011), Lindsten *et al.* (2011), and Pelckmans *et al.* (2005)
- Algorithms: Chen *et al.* (2015), Chi and Lange (2015), Ho *et al.* (2019), Panahi *et al.* (2017), and Sun *et al.* (2018)
- Two-Way Matrix / Bi-Clustering: Chi *et al.* (2017) and Weylandt (2019)
- Multi-Way Tensor / Co-Clustering: Chi *et al.* (2018)
- Consistency: Panahi *et al.* (2017), Radchenko and Mukherjee (2017), Tan and Witten (2015), and Zhu *et al.* (2014)
- Non-Convex Penalties: Marchetti and Zhou (2014), Pan *et al.* (2013), Shah and Koltun (2017), and Wu *et al.* (2016)
- Robustness: Wang *et al.* (2016)
- Feature Selection: Wang *et al.* (2018)
- Generalized Losses: Wang and Allen (2019+)

Despite all this, relatively little adoption: speed, graphics, and software support

Convex Clustering: Splitting Algorithms

Simplified form:

$$\underset{\mathbf{U} \in \mathbb{R}^{n \times p}}{\arg\min} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\|_F^2 + \lambda \underbrace{\|\mathbf{D}\mathbf{U}\|_{\text{row},q}}_{P(\mathbf{D}\mathbf{U})}$$

Convex Clustering: Splitting Algorithms

Simplified form:

$$\underset{\mathbf{U}\in\mathbb{R}^{n\times p}}{\arg\min}\ \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_F^2+\lambda\underbrace{\|\mathbf{D}\mathbf{U}\|_{\text{row},q}}_{P(\mathbf{D}\mathbf{U})}$$

ADMM for Convex Clustering (Chi and Lange 2015; Weylandt *et al.* 2019+):

1. $\mathbf{U}^{(k+1)}=(\mathbf{I}+\rho\mathbf{D}^T\mathbf{D})^{-1}(\mathbf{X}+\mathbf{D}(\mathbf{V}^{(k)}-\mathbf{Z}^{(k)}))$
2. $\mathbf{V}^{(k+1)}=\text{prox}_{\lambda/\rho\,P(\cdot)}(\mathbf{D}\mathbf{U}^{(k+1)}+\mathbf{Z}^{(k)})$
3. $\mathbf{Z}^{(k+1)}=\mathbf{Z}^{(k)}+\rho(\mathbf{D}\mathbf{U}^{(k+1)}-\mathbf{V}^{(k+1)})$

Fastest general purpose solver for convex clustering, but still slow ...

Convex Clustering: Splitting Algorithms

Simplified form:

$$\underset{\mathbf{U} \in \mathbb{R}^{n \times p}}{\arg\min} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\|_F^2 + \lambda \underbrace{\|\mathbf{D}\mathbf{U}\|_{\text{row},q}}_{P(\mathbf{D}\mathbf{U})}$$

ADMM for Convex Clustering (Chi and Lange 2015; Weylandt *et al.* 2019+):

1. $\mathbf{U}^{(k+1)} = (\mathbf{I} + \rho \mathbf{D}^T\mathbf{D})^{-1}(\mathbf{X} + \mathbf{D}(\mathbf{V}^{(k)} - \mathbf{Z}^{(k)}))$
2. $\mathbf{V}^{(k+1)} = \text{prox}_{\lambda/\rho\, P(\cdot)}(\mathbf{D}\mathbf{U}^{(k+1)} + \mathbf{Z}^{(k)})$
3. $\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \rho(\mathbf{D}\mathbf{U}^{(k+1)} - \mathbf{V}^{(k+1)})$

Fastest general purpose solver for convex clustering, but still slow ...

- $\mathbf{D}$-matrix has $\binom{n}{2}$ rows but large nullspace

Convex Clustering: Splitting Algorithms

Simplified form:

$$\underset{\mathbf{U} \in \mathbb{R}^{n \times p}}{\arg\min} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\|_F^2 + \lambda \underbrace{\|\mathbf{D}\mathbf{U}\|_{\text{row},q}}_{P(\mathbf{D}\mathbf{U})}$$

ADMM for Convex Clustering (Chi and Lange 2015; Weylandt *et al.* 2019+):

1. $\mathbf{U}^{(k+1)} = (\mathbf{I} + \rho \mathbf{D}^T\mathbf{D})^{-1}(\mathbf{X} + \mathbf{D}(\mathbf{V}^{(k)} - \mathbf{Z}^{(k)}))$
2. $\mathbf{V}^{(k+1)} = \text{prox}_{\lambda/\rho\, P(\cdot)}(\mathbf{D}\mathbf{U}^{(k+1)} + \mathbf{Z}^{(k)})$
3. $\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \rho(\mathbf{D}\mathbf{U}^{(k+1)} - \mathbf{V}^{(k+1)})$

Fastest general purpose solver for convex clustering, but still slow …

- $\mathbf{D}$-matrix has $\binom{n}{2}$ rows but large nullspace
- Fusion penalty non-separable and induces no (computational) sparsity

Convex Clustering: Splitting Algorithms

Simplified form:

$$\underset{\mathbf{U}\in\mathbb{R}^{n\times p}}{\arg\min}\ \frac{1}{2}\|\mathbf{X}-\mathbf{U}\|_F^2 + \lambda \underbrace{\|\mathbf{D}\mathbf{U}\|_{\text{row},q}}_{P(\mathbf{D}\mathbf{U})}$$

ADMM for Convex Clustering (Chi and Lange 2015; Weylandt *et al.* 2019+):

1. $\mathbf{U}^{(k+1)} = (\mathbf{I} + \rho\mathbf{D}^T\mathbf{D})^{-1}(\mathbf{X} + \mathbf{D}(\mathbf{V}^{(k)} - \mathbf{Z}^{(k)}))$
2. $\mathbf{V}^{(k+1)} = \text{prox}_{\lambda/\rho\, P(\cdot)}(\mathbf{D}\mathbf{U}^{(k+1)} + \mathbf{Z}^{(k)})$
3. $\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \rho(\mathbf{D}\mathbf{U}^{(k+1)} - \mathbf{V}^{(k+1)})$

Fastest general purpose solver for convex clustering, but still slow ...

- $\mathbf{D}$-matrix has $\binom{n}{2}$ rows but large nullspace
- Fusion penalty non-separable and induces no (computational) sparsity
- $\mathcal{O}(n^2p)$ variables

Difficulties of Convex Clustering Optimization

Dendrogram recovery:

- Need to solve at (at least) $n-1$ different λ values
- Don't know what those are *a priori*

Difficulties of Convex Clustering Optimization

Dendrogram recovery:

- Need to solve at (at least) $n - 1$ different λ values
- Don't know what those are *a priori*

Grid search expensive

Not amenable to homotopy (path-following) algorithms

Local and Global Accuracy

Two (contrary?) aims:

- *Local* Accuracy: optimization convergence at all λ
- *Global* Accuracy: solution at dense λ grid

Local and Global Accuracy

Two (contrary?) aims:

- *Local* Accuracy: optimization convergence at all λ
- *Global* Accuracy: solution at dense λ grid

Standard optimization techniques give *local* accuracy

Relatively little consideration of *global* accuracy

Local and Global Accuracy

Two (contrary?) aims:

- *Local* Accuracy: optimization convergence at all λ
- *Global* Accuracy: solution at dense λ grid

Standard optimization techniques give *local* accuracy

Relatively little consideration of *global* accuracy

Global accuracy often more interesting: variable selection order, dendrograms, *etc.*

It’s Getting Hot In Here! Advantages of Warm Starting

Useful trick: *warm starts*!

If solving for grid of λ, use $\hat{\mathsf{U}}_{\lambda_{k-1}}$ to start algorithm for $\hat{\mathsf{U}}_{\lambda_k}$

It's Getting Hot In Here! Advantages of Warm Starting

Useful trick: *warm starts*!

If solving for grid of λ, use $\hat{\mathsf{U}}_{\lambda_{k-1}}$ to start algorithm for $\hat{\mathsf{U}}_{\lambda_k}$

Starting near solution reduces number of iterations needed to converge

It's Getting Hot In Here! Advantages of Warm Starting

Useful trick: *warm starts*!

If solving for grid of λ, use $\hat{\mathsf{U}}_{\lambda_{k-1}}$ to start algorithm for $\hat{\mathsf{U}}_{\lambda_k}$

Starting near solution reduces number of iterations needed to converge

Second-order benefit: algorithms have improved *local* convergence rates near solutions

Some Mild Heresy

Warm-Started ADMM:

- Initialize $l = 0$, $\lambda_l = \epsilon$, $\mathsf{V}^{(0)} = \mathsf{Z}^{(0)} = \mathsf{DX}$
- Repeat until $\|\mathsf{V}^{(k)}\| = 0$:
  - Repeat until convergence:
    (i) $\mathsf{U}^{(k+1)} = (\mathsf{I} + \mathsf{D}^T\mathsf{D})^{-1}\left(\mathsf{X} + \mathsf{D}^T(\mathsf{V}^{(k)} - \mathsf{Z}^{(k)})\right)$
    (ii) $\mathsf{V}^{(k+1)} = \text{prox}_{\lambda_l P(\cdot)}\left(\mathsf{DU}^{(k+1)} + \mathsf{Z}^{(k)}\right)$
    (iii) $\mathsf{Z}^{(k+1)} = \mathsf{Z}^{(k)} + \mathsf{DU}^{(k+1)} - \mathsf{V}^{(k+1)}$
    (iv) $k := k + 1$
  - Store $\hat{\mathsf{U}}_{\lambda_l} = \mathsf{U}^{(k)}$
  - Update regularization: $l := l + 1$; $\lambda_l := \lambda_{l-1} * t$
- Return $\{\hat{\mathsf{U}}_\lambda\}$ as the regularization path

Some Mild Heresy

CARP Algorithm:

- Initialize $l = 0$, $\lambda_l = \epsilon$, $\mathbf{V}^{(0)} = \mathbf{Z}^{(0)} = \mathbf{DX}$
- Repeat until $\|\mathbf{V}^{(k)}\| = 0$:
  - Do Once:
    (i) $\mathbf{U}^{(k+1)} = (\mathbf{I} + \mathbf{D}^T\mathbf{D})^{-1}\left(\mathbf{X} + \mathbf{D}^T(\mathbf{V}^{(k)} - \mathbf{Z}^{(k)})\right)$
    (ii) $\mathbf{V}^{(k+1)} = \text{prox}_{\lambda_l P(\cdot)}\left(\mathbf{DU}^{(k+1)} + \mathbf{Z}^{(k)}\right)$
    (iii) $\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \mathbf{DU}^{(k+1)} - \mathbf{V}^{(k+1)}$
    (iv) $k := k + 1$
  - Store $\hat{\mathbf{U}}_{\lambda_l} = \mathbf{U}^{(k)}$
  - Update regularization: $l := l + 1$; $\lambda_l := \lambda_{l-1} * t$
- Return $\{\hat{\mathbf{U}}_\lambda\}$ as the algorithmic regularization path

Some Mild Heresy

CARP Algorithm:

- Initialize $l = 0$, $\lambda_l = \epsilon$, $\mathbf{V}^{(0)} = \mathbf{Z}^{(0)} = \mathbf{DX}$
- Repeat until $\|\mathbf{V}^{(k)}\| = 0$:
  - Do Once:
    (i) $\mathbf{U}^{(k+1)} = (\mathbf{I} + \mathbf{D}^T\mathbf{D})^{-1}\left(\mathbf{X} + \mathbf{D}^T(\mathbf{V}^{(k)} - \mathbf{Z}^{(k)})\right)$
    (ii) $\mathbf{V}^{(k+1)} = \text{prox}_{\lambda_l P(\cdot)}\left(\mathbf{DU}^{(k+1)} + \mathbf{Z}^{(k)}\right)$
    (iii) $\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \mathbf{DU}^{(k+1)} - \mathbf{V}^{(k+1)}$
    (iv) $k := k + 1$
  - Store $\hat{\mathbf{U}}_{\lambda_l} = \mathbf{U}^{(k)}$
  - Update regularization: $l := l + 1$; $\lambda_l := \lambda_{l-1} * t$
- Return $\{\hat{\mathbf{U}}_\lambda\}$ as the algorithmic regularization path

CARP: Convex Clustering via Algorithmic Regularization Paths

Algorithmic Regularization

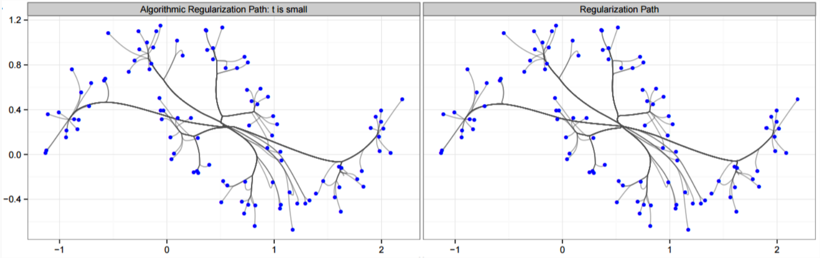
Algorithmic Regularization: Single Optimization Step then Update λ

Algorithmic Regularization

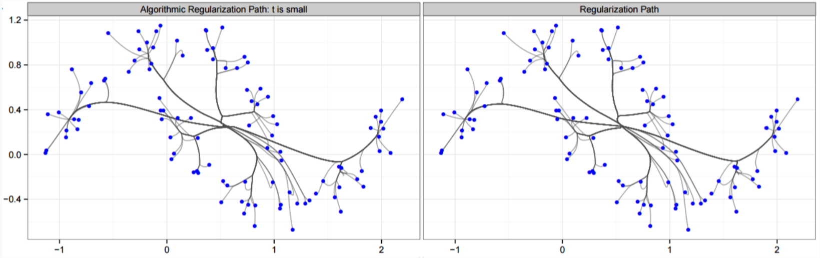
Algorithmic Regularization: Single Optimization Step then Update λ

Faster, but can it work?

Eppur si muove!



Eppur si muove!



Yes - it seems to work!

Intuition: Warm-starting at previous iteration gets "good enough" answer in one-step

Practical Advantages: Same number of iterations spent at **many** more $\lambda \implies$ finer grid!

A Convergence Theorem

Theorem (Informal): Global Recovery of Entire Path

As the step-size t goes to zero, CARP recovers the entire solution path (primal and dual):

$$\max\left\{\sup_{\lambda}\inf_{k}\left\|\mathbf{U}^{(k)} - \hat{\mathbf{U}}_{\lambda}\right\|, \sup_{k}\inf_{\lambda}\left\|\mathbf{U}^{(k)} - \hat{\mathbf{U}}_{\lambda}\right\|\right\} \xrightarrow{(t,\epsilon)\to(1,0)} 0$$

$$\max\left\{\sup_{\lambda}\inf_{k}\left\|\mathbf{Z}^{(k)} - \hat{\mathbf{Z}}_{\lambda}\right\|, \sup_{k}\inf_{\lambda}\left\|\mathbf{Z}^{(k)} - \hat{\mathbf{Z}}_{\lambda}\right\|\right\} \xrightarrow{(t,\epsilon)\to(1,0)} 0$$

A Convergence Theorem

Theorem (Informal): Global Recovery of Entire Path

As the step-size t goes to zero, CARP recovers the entire solution path (primal and dual):

$$\max\left\{\sup_{\lambda}\inf_{k}\left\|\mathrm{U}^{(k)} - \hat{\mathrm{U}}_\lambda\right\|, \sup_{k}\inf_{\lambda}\left\|\mathrm{U}^{(k)} - \hat{\mathrm{U}}_\lambda\right\|\right\} \xrightarrow{(t,\epsilon)\to(1,0)} 0$$

$$\max\left\{\sup_{\lambda}\inf_{k}\left\|\mathrm{Z}^{(k)} - \hat{\mathrm{Z}}_\lambda\right\|, \sup_{k}\inf_{\lambda}\left\|\mathrm{Z}^{(k)} - \hat{\mathrm{Z}}_\lambda\right\|\right\} \xrightarrow{(t,\epsilon)\to(1,0)} 0$$

Very strong convergence - global and local + primal and dual

A Convergence Theorem

Theorem (Informal): Global Recovery of Entire Path

As the step-size t goes to zero, CARP recovers the entire solution path (primal and dual):

$$\max\left\{\sup_{\lambda}\inf_{k}\left\|\mathsf{U}^{(k)}-\hat{\mathsf{U}}_{\lambda}\right\|,\sup_{k}\inf_{\lambda}\left\|\mathsf{U}^{(k)}-\hat{\mathsf{U}}_{\lambda}\right\|\right\}\xrightarrow{(t,\epsilon)\to(1,0)}0$$

$$\max\left\{\sup_{\lambda}\inf_{k}\left\|\mathsf{Z}^{(k)}-\hat{\mathsf{Z}}_{\lambda}\right\|,\sup_{k}\inf_{\lambda}\left\|\mathsf{Z}^{(k)}-\hat{\mathsf{Z}}_{\lambda}\right\|\right\}\xrightarrow{(t,\epsilon)\to(1,0)}0$$

Very strong convergence - global and local + primal and dual

The whole path and nothing but the path

Sketch of Proof

Key elements of proof:

- Problem is *strongly* convex (always) so ADMM converges linearly (Deng and Yin 2016)
- Solution path is *Lipschitz* so $\|\partial \hat{\mathbf{U}}_\lambda / \partial \lambda\|$ is bounded above

Proof Sketch:

Sketch of Proof

Key elements of proof:

- Problem is *strongly* convex (always) so ADMM converges linearly (Deng and Yin 2016)
- Solution path is *Lipschitz* so $\|\partial \hat{\mathbf{U}}_\lambda / \partial \lambda\|$ is bounded above

Proof Sketch:

- At initalization:

$$\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_\epsilon\| \leq L\epsilon \text{ by Lipschitz-ness of } \mathbf{U}_\lambda$$

Sketch of Proof

Key elements of proof:

- Problem is *strongly* convex (always) so ADMM converges linearly (Deng and Yin 2016)
- Solution path is *Lipschitz* so $\|\partial \hat{\mathbf{U}}_\lambda / \partial \lambda\|$ is bounded above

Proof Sketch:

- At initalization:

$$\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_\epsilon\| \leq L\epsilon \text{ by Lipschitz-ness of } \mathbf{U}_\lambda$$

- After one step:

$$\|\mathbf{U}^{(1)} - \hat{\mathbf{U}}_{t\epsilon}\| \leq c\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_{t\epsilon}\| \leq c\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_{\epsilon}\| + c\|\hat{\mathbf{U}}_\epsilon - \hat{\mathbf{U}}_{t\epsilon}\| \leq cL\epsilon + c(t-1)\epsilon$$

Sketch of Proof

Key elements of proof:

- Problem is *strongly* convex (always) so ADMM converges linearly (Deng and Yin 2016)
- Solution path is *Lipschitz* so $\|\partial \hat{\mathbf{U}}_\lambda / \partial \lambda\|$ is bounded above

Proof Sketch:

- At initalization:

$$\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_\epsilon\| \leq L\epsilon \text{ by Lipschitz-ness of } \mathbf{U}_\lambda$$

- After one step:

$$\|\mathbf{U}^{(1)} - \hat{\mathbf{U}}_{t\epsilon}\| \leq c\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_{t\epsilon}\| \leq c\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_\epsilon\| + c\|\hat{\mathbf{U}}_\epsilon - \hat{\mathbf{U}}_{t\epsilon}\| \leq cL\epsilon + c(t-1)\epsilon$$

- Iterating:

$$\|\mathbf{U}^{(k)} - \hat{\mathbf{U}}_{t^k\epsilon}\| \leq c^k L\epsilon + L(t-1)\epsilon t^k \sum_{i=1}^{k-1} \left(\frac{c}{t}\right)^i$$

Sketch of Proof

Key elements of proof:

- Problem is *strongly* convex (always) so ADMM converges linearly (Deng and Yin 2016)
- Solution path is *Lipschitz* so $\|\partial \hat{\mathbf{U}}_\lambda / \partial \lambda\|$ is bounded above

Proof Sketch:

- At initalization:

$$\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_\epsilon\| \leq L\epsilon \text{ by Lipschitz-ness of } \mathbf{U}_\lambda$$

- After one step:

$$\|\mathbf{U}^{(1)} - \hat{\mathbf{U}}_{t\epsilon}\| \leq c\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_{t\epsilon}\| \leq c\|\mathbf{U}^{(0)} - \hat{\mathbf{U}}_\epsilon\| + c\|\hat{\mathbf{U}}_\epsilon - \hat{\mathbf{U}}_{t\epsilon}\| \leq cL\epsilon + c(t-1)\epsilon$$

- Iterating:

$$\|\mathbf{U}^{(k)} - \hat{\mathbf{U}}_{t^k\epsilon}\| \leq c^k L\epsilon + L(t-1)\epsilon t^k \sum_{i=1}^{k-1} \left(\frac{c}{t}\right)^i$$

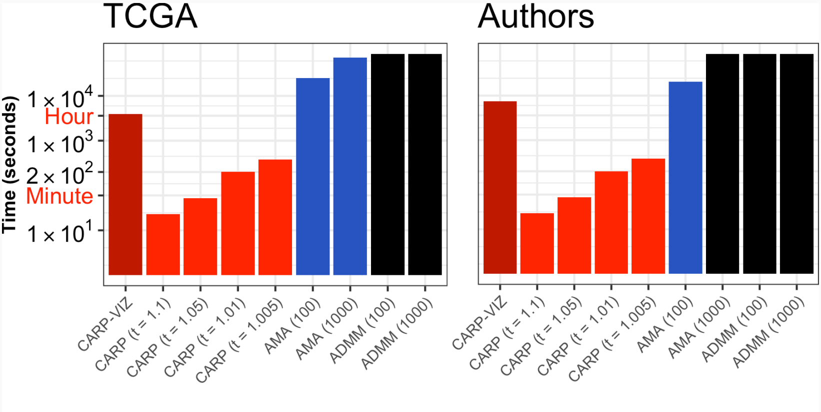
- Show this goes to zero for all k simultaneously as $t, \epsilon \to 0$

Authors and Genes

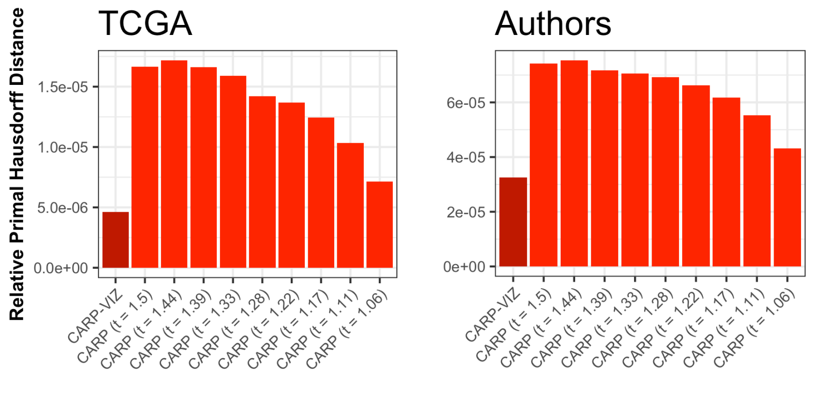
Two test data sets:

- Authors $\in \mathbb{R}^{841 \times 69}$: stop-word counts from 4 authors
- TCGA $\in \mathbb{R}^{438 \times 353}$: gene expression from 3 breast cancer subtypes

Authors and Genes



Authors and Genes



CBASS: Convex Bi-Clustering

Similar modification to Chi *et al.* (2017) or Weylandt (2019) yields

CBASS:
Convex BiClustering via Algorithmic Regularization with Small Steps

Also in `clustRviz`

Future Work

Where else can we use Algorithmic Regularization:

- Signal Approximation
- "Big *n* Problems"
- ℓ_2 (Tikhonov) Regularization

Future Work

Where else can we use Algorithmic Regularization:

- Signal Approximation
- "Big *n* Problems"
- ℓ_2 (Tikhonov) Regularization

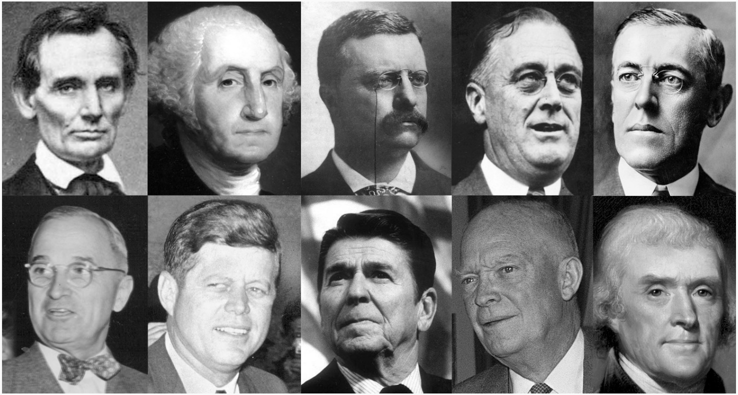
Extensions of Algorithmic Regularization:

- Inexact Updates
- Multi-Block
- Non-Strongly Convex
- "Nice" Non-Convex
- Stochastic / Parallel Updates

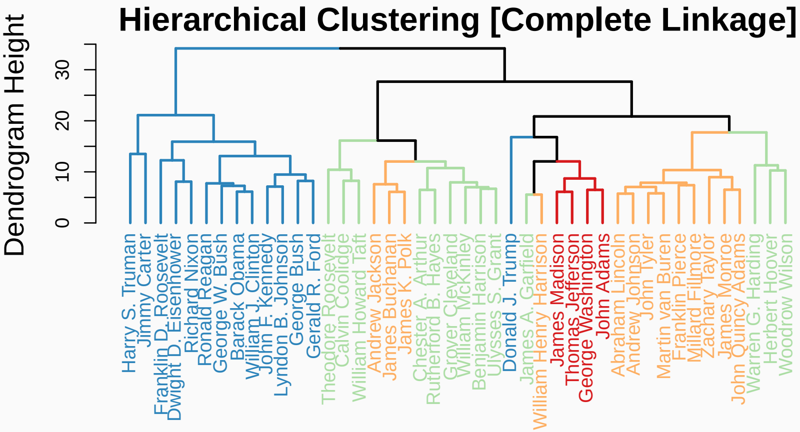
One Fun Thing

Presidential speeches data set ($n = 44, p = 75$):

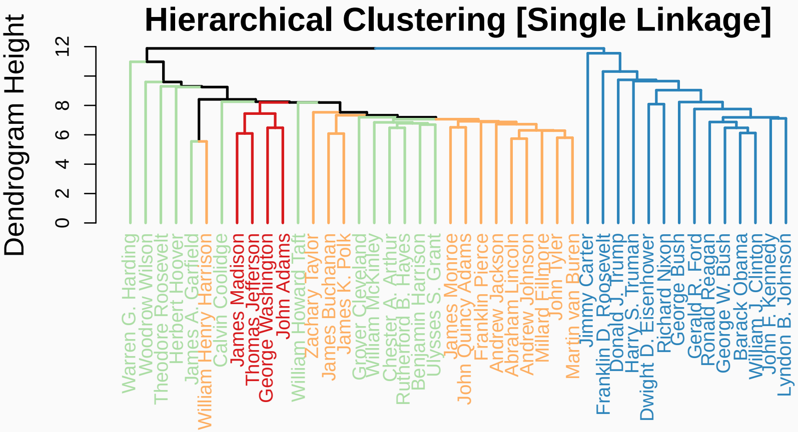
- Relative word frequency of top 75 words from inaugurations, State of the Union, and other famous speeches
- Words are stemmed and frequencies are log-transformed



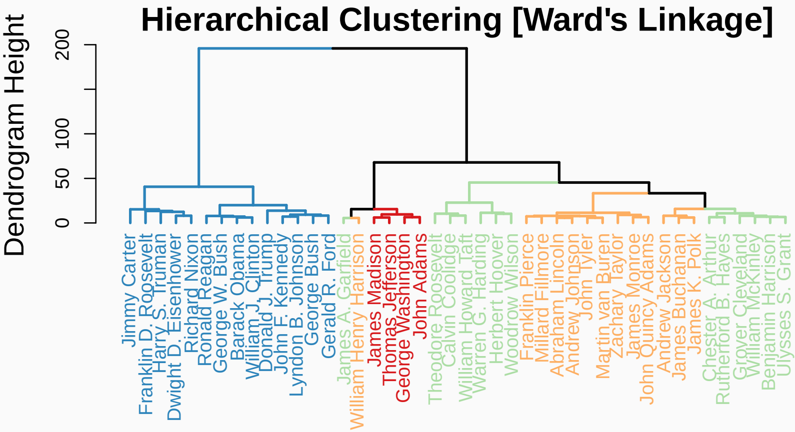
Dendrograms:



Dendrograms:

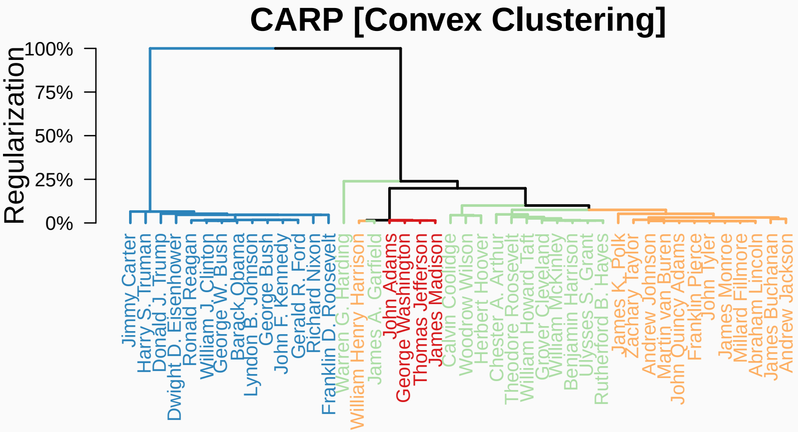


Dendrograms:



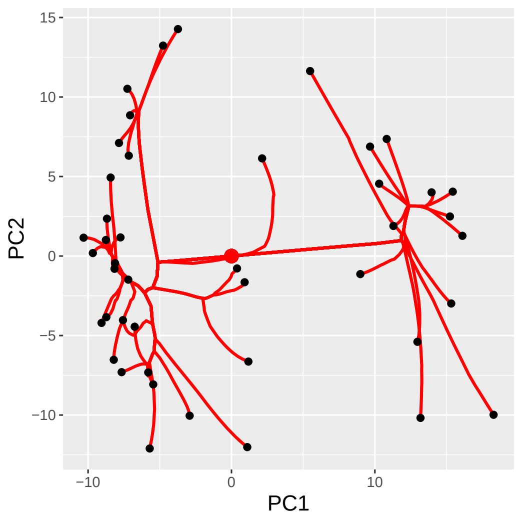
One Fun Thing

Dendrograms:



One Fun Thing

Paths:



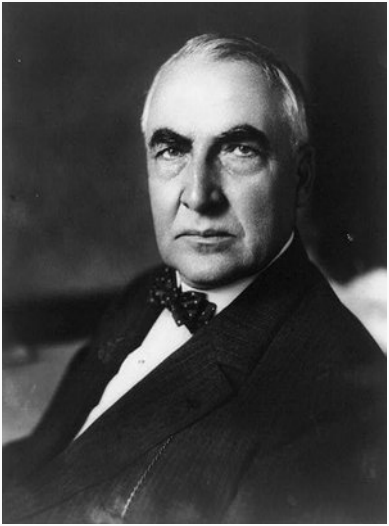
One Fun Thing

Outlier:

- Republican
- Known for pro-business and anti-immigration policies
- Tried to upend traditional alliances
- First president elected from previous (non-traditional) background
- Campaigned on return to past glories

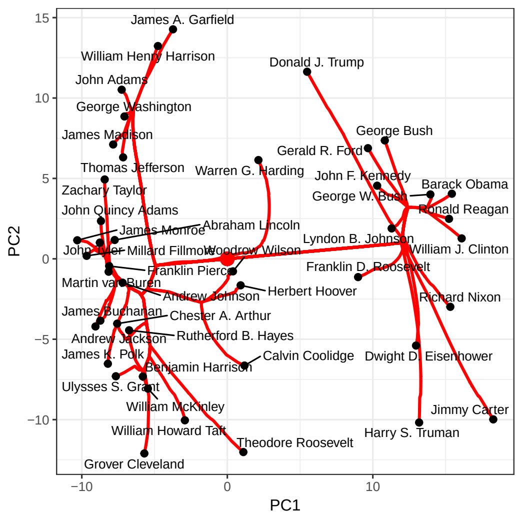
One Fun Thing

Outlier:

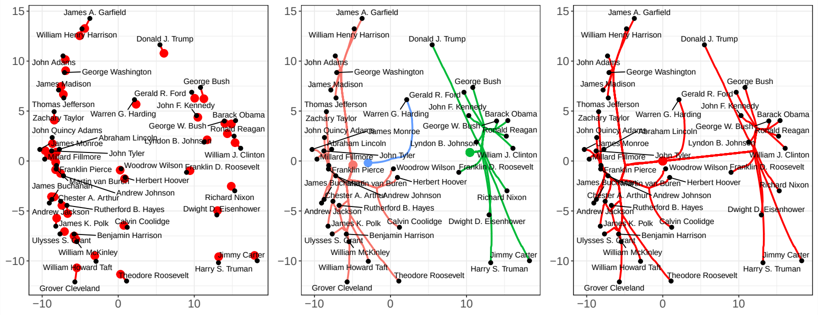


One Fun Thing

Paths:



Paths:


15
10
5
0
−5
−10
−10
0
10
James A. Garfield
William Henry Harrison
Donald J. Trump
John Adams
George Washington
James Madison
George Bush
Gerald R. Ford
Thomas Jefferson
John F. Kennedy
Warren G. Harding
Zachary Taylor
Barack Obama
George W. Bush
Ronald Reagan
John Quincy Adams
Abraham Lincoln
James Monroe
Lyndon B. Johnson
John Tyler
Millard Fillmore
William J. Clinton
Woodrow Wilson
Franklin D. Roosevelt
Franklin Pierce
Martin van Buren
Herbert Hoover
James Buchanan
Andrew Johnson
Richard Nixon
Chester A. Arthur
Rutherford B. Hayes
Dwight D. Eisenhower
Andrew Jackson
James K. Polk
Calvin Coolidge
Benjamin Harrison
Ulysses S. Grant
William McKinley
Jimmy Carter
William Howard Taft
Harry S. Truman
Theodore Roosevelt
Grover Cleveland

Conclusions

W., Nagorski, and Allen: “Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization .” *JCGS* 2019+.

github.com/DataSlingers/clustRviz

Thank you!

Conclusions

W., Nagorski, and Allen: "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization ." *JCGS* 2019+.

- Algorithmic Regularization Meta-Algorithm: One Optimization Step per λ

github.com/DataSlingers/clustRviz

Thank you!

Conclusions

W., Nagorski, and Allen: "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization ." *JCGS* 2019+.

- Algorithmic Regularization Meta-Algorithm: One Optimization Step per λ
    - Faster Computation & More Accurate Global Structure

github.com/DataSlingers/clustRviz

Thank you!

Conclusions

W., Nagorski, and Allen: "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization ." *JCGS* 2019+.

- Algorithmic Regularization Meta-Algorithm: One Optimization Step per λ
  - Faster Computation & More Accurate Global Structure
  - CARP and CBASS Algorithms for Clustering and Bi-Clustering

github.com/DataSlingers/clustRviz

Thank you!

Conclusions

W., Nagorski, and Allen: "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization ." *JCGS* 2019+.

- Algorithmic Regularization Meta-Algorithm: One Optimization Step per λ
    - Faster Computation & More Accurate Global Structure
    - CARP and CBASS Algorithms for Clustering and Bi-Clustering
    - First *global* convergence result for one-step schemes

github.com/DataSlingers/clustRviz

Thank you!

Conclusions

W., Nagorski, and Allen: "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization ." *JCGS* 2019+.

- Algorithmic Regularization Meta-Algorithm: One Optimization Step per λ
  - Faster Computation & More Accurate Global Structure
  - CARP and CBASS Algorithms for Clustering and Bi-Clustering
  - First *global* convergence result for one-step schemes
- Convex Clustering:

github.com/DataSlingers/clustRviz

Thank you!

Conclusions

W., Nagorski, and Allen: "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization ." *JCGS* 2019+.

- Algorithmic Regularization Meta-Algorithm: One Optimization Step per λ
  - Faster Computation & More Accurate Global Structure
  - CARP and CBASS Algorithms for Clustering and Bi-Clustering
  - First *global* convergence result for one-step schemes
- Convex Clustering:
  - Strong Statistical Guarantees

github.com/DataSlingers/clustRviz

Thank you!

Conclusions

W., Nagorski, and Allen: "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization ." *JCGS* 2019+.

- Algorithmic Regularization Meta-Algorithm: One Optimization Step per λ
  - Faster Computation & More Accurate Global Structure
  - CARP and CBASS Algorithms for Clustering and Bi-Clustering
  - First *global* convergence result for one-step schemes
- Convex Clustering:
  - Strong Statistical Guarantees
  - Fast(er) Computation + Dynamic & Iteractive Visualizations!

github.com/DataSlingers/clustRviz

Thank you!

References i

Chen, Gary K., Eric C. Chi, John Michael O. Ranola, and Kenneth Lange (2015). "Convex Clustering: An Attractive Alternative to Hierarchical Clustering". *PLOS Computational Biology* 11.5, e1004228.

Chi, Eric C., Genevera I. Allen, and Richard G. Baraniuk (2017). "Convex Biclustering". *Biometrics* 73.1, pp. 10–19.

Chi, Eric C., Brian R. Gaines, Will Wei Sun, Hua Zhou, and Jian Yang (2018). "Provable Convex Co-Clustering of Tensors". *ArXiv Pre-Print 1803.06518.*

Chi, Eric C. and Kenneth Lange (2015). "Splitting Methods for Convex Clustering". *Journal of Computational and Graphical Statistics* 24.4, pp. 994–1013.

Deng, Wei and Wotao Yin (2016). "On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers". *Journal of Scientific Computing* 66.3, pp. 889–916.

Ho, Nhat, Tianyi Lin, and Michael I. Jordan (2019). "Global Error Bounds and Linear Convergence for Gradient-Based Algorithms for Trend Filtering and ℓ_1-Convex Clustering". *ArXiv Pre-Print 1904.07462.*

References ii

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

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

Marchetti, Yuliya and Qing Zhou (2014). "Solution Path Clustering with Adaptive Concave Penalty". *Electronic Journal of Statistics* 8.1, pp. 1569–1603.

Pan, Wei, Xiaotong Shen, and Binghui Liu (2013). "Cluster Analysis: Unsupervised Learning via Supervised Learning with a Non-convex Penalty". *Journal of Machine Learning Research* 14, pp. 1865–1889.

Panahi, Ashkan, Devdatt Dubhashi, Fredrik D. Johansson, and Chiranjib Bhattacharyya (2017). "Clustering by Sum of Norms: Stochastic Incremental Algorithm, Convergence, and Cluster Recovery". In: *ICML:2017: Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Sydney, Australia: PMLR, pp. 2769–2777.

References iii

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

Radchenko, Peter and Gourab Mukherjee (2017). "Convex Clustering via ℓ_1 Fusion Penalization". *Journal of the Royal Statistical Society, Series B: Statistical Methodology* 79.5, pp. 1527–1546.

Shah, Sohil Atul and Vladlen Koltun (2017). "Robust continuous clustering". *Proceedings of the National Academy of Sciences of the United States* 114.37, pp. 9814–9819.

Sun, Defeng, Kim-Chuan Toh, and Yancheng Yuan (2018). "Convex Clustering: Model, Theoretical Guarantee and Efficient Algorithm". *ArXiv Pre-Print 1810.02677*.

Tan, Kean Ming and Daniela Witten (2015). "Statistical Properties of Convex Clustering". *Electronic Journal of Statistics* 9.2, pp. 2324–2347.

Wang, Binhuan, Yilong Zhang, Will Wei Sun, and Yixin Fang (2018). "Sparse Convex Clustering". *Journal of Computational and Graphical Statistics* 27.2, pp. 393–403.

Wang, Qi, Pinghua Gong, Shiyu Chang, Thomas S. Huang, and Jiayu Zhou (2016). "Robust Convex Clustering Analysis". In: *ICDM 2016: Proceedings of the 16th IEEE International Conference on Data Mining*. Ed. by Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu. Barcelona, Spain, pp. 1263–1268.

Weylandt, Michael (2019). "Splitting Methods for Convex Bi-Clustering and Co-Clustering". In: *DSW 2019: Proceedings of the 2nd IEEE Data Science Workshop*. Ed. by Georgios B. Giannakis, Geert Leus, and Antonio G. Marques. Minneapolis, Minnesota: IEEE.

Weylandt, Michael, John Nagorski, and Genevera I. Allen (2019+). "Dynamic Visualization and Fast Computation for Convex Clustering via Algorithmic Regularization". *Journal of Computational and Graphical Statistics*.

Wu, Chong, Sunghoon Kown, Xiaotong Shen, and Wei Pan (2016). "A New Algorithm and Theory for Penalized Regression-based Clustering". *Journal of Machine Learning Research* 17.188, pp. 1–25.

References v

Zhu, Changbo, Huan Xu, Chenlei Leng, and Shuicheng Yan (2014). “Convex Optimization Procedure for Clustering: Theoretical Revisit”. In: *NIPS 2014: Advances in Neural Information Processing Systems 27*. Ed. by Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Killian Q. Weinberger. Montréal, Canada: Curran Associates, Inc., pp. 1619–1627.