Inference for Dependent Data with Cluster Learning

Jianfei Cao, Chicago Booth

joint with

Christian Hansen, Chicago Booth
Damian Kozbur, UZH
Lucciano Villacorta, Bank of Chile

December 31, 2019

Introduction

- Failing to account for dependence leads to invalid inference
- e.g. linear model

$$y_i = x_i'\beta + \varepsilon_i$$

asymptotic variance of $\widehat{\beta}$ depends on $\frac{1}{n}\sum_i \sum_j E[x_i x_j' \varepsilon_i \varepsilon_j]$

Introduction

- Failing to account for dependence leads to invalid inference

- e.g. linear model

$$y_i = x_i'\beta + \varepsilon_i$$

  asymptotic variance of $\widehat{\beta}$ depends on $\frac{1}{n}\sum_i \sum_j E[x_i x_j' \varepsilon_i \varepsilon_j]$

- Group-based inference: Given clustering $\mathcal{C} = \{C_g\}_{g=1}^{G}$
  - Cluster Covariance Estimator (**CCE**)
  - Ibragimov and Mueller (2010, **IM**)
  - Canay, Romano, and Shaikh (2017, **CRS**)
  - ...

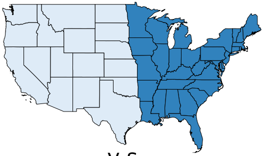
- Focus on a few large groups (small G)

Practical Issues

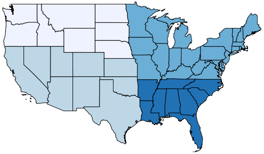
- Choice of clustering often ad-hoc

Two tuning parameters

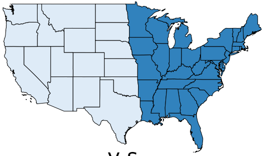
Number of groups G



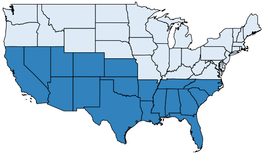
v.s.



G partitions



v.s.

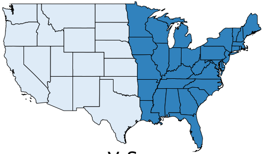


Practical Issues

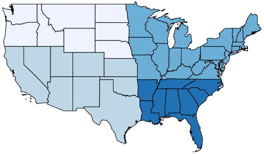
- Choice of clustering often ad-hoc

Two tuning parameters

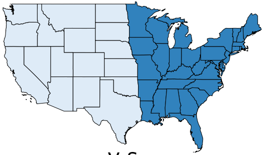
Number of groups G



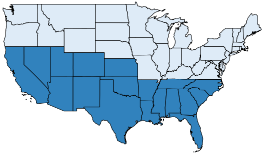
v.s.



G partitions



v.s.



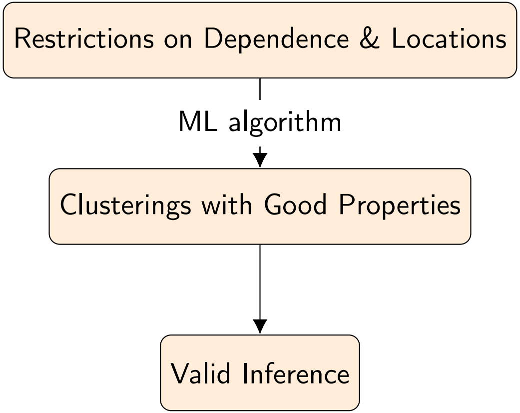
- Goal: data-driven methods to make these choices
  - Use Unsupervised Learning from ML to form partitions given G
  - Use simulation to choose G based on inferential properties

This paper

- Difficulty: Recovering the "true" clustering is hard

This paper

- Difficulty: Recovering the “true” clustering is hard
- Idea: Find clusterings with good properties



Proposed Inferential Methods

Generate $\{\mathcal{C}^{(G)}\}_{G=2}^{\bar{G}}$ by *k*-medoids

↓

Fit a parametric covariance model to scores

↓

Select $\mathcal{C}^*$ by simulated size & power

↓

Perform cluster-based inference using $\mathcal{C}^*$

Proposed Inferential Methods

Generate $\{\mathcal{C}^{(G)}\}_{G=2}^{\bar{G}}$ by *k*-medoids

↓

Fit a parametric covariance model to scores

↓

Select $\mathcal{C}^*$ by simulated size & power

↓

Perform cluster-based inference using $\mathcal{C}^*$

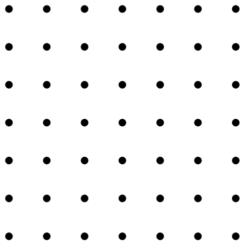
IV simulation on $H_0 : \beta_1 = 0$

Method	Median	MAD	Size	Power
White	0.006	0.21	0.31	0.90
S-HAC	0.006	0.21	0.15	0.89
Our method	0.014	0.74	0.06	0.81

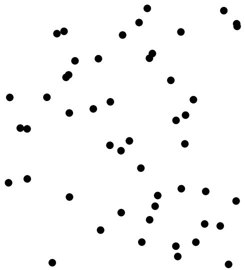
Setup

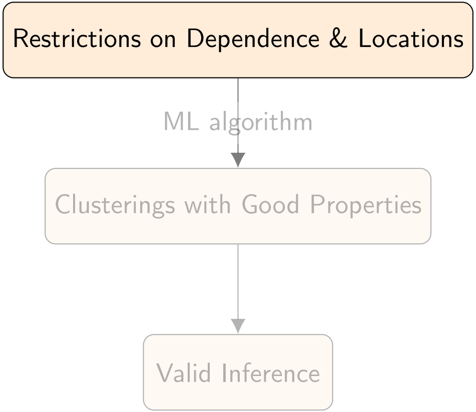
Data:

$$\{\{Z_{i,n}\}_{i=1}^{n}, (\mathrm{X}_n, d_n)\}_{n\geq 1}$$



or

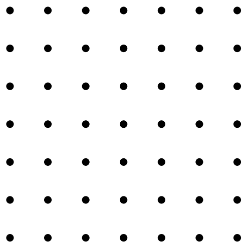




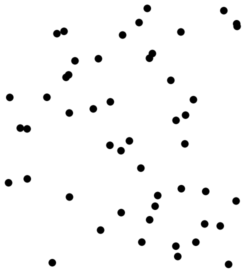
Setup

Data:

$$\{\{Z_{i,n}\}_{i=1}^{n}, (\mathrm{X}_n, d_n)\}_{n\geq 1}$$



or



Restrictions on Dependence & Locations

ML algorithm

Clusterings with Good Properties

Valid Inference

Condition 1 (Mixing)

Dependence between $Z_{i,n}$ and $Z_{j,n}$ decays sufficiently fast for large $d_n(i,j)$ and $Z_{i,n}$ have sufficient finite moments

Restrictions on Locations

Condition 2

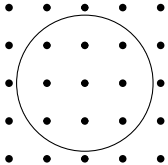
- *(Ahlfors Regularity)* $\exists C, \delta$, *s.t.* $\forall n \geq 1, x \in \mathrm{X}_n, r > 0$

$$|B_{\mathrm{X}_n,r}(x)| \approx Cr^{\delta}$$

- *(Approximate Convexity)* $\forall x, y \in \mathrm{X}_n$ *and* $\lambda \in [0,1]$, $\exists z \in \mathrm{X}_n$ *s.t.*

$$z \approx \lambda x + (1-\lambda) y$$

Ahlfors Regularity:



$$|B_{\mathrm{X}_n,r}(x)| \approx \pi r^2 / h^2$$
$$\Downarrow$$
$$\delta = 2,\ \ C \propto 1/h^2$$

Restrictions on Locations

Condition 2

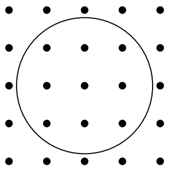
- *(Ahlfors Regularity)* $\exists C, \delta$, *s.t.* $\forall n \geq 1, x \in \mathsf{X}_n, r > 0$

$$|B_{\mathsf{X}_n, r}(x)| \approx Cr^{\delta}$$

- *(Approximate Convexity)* $\forall x, y \in \mathsf{X}_n$ *and* $\lambda \in [0,1]$, $\exists z \in \mathsf{X}_n$ *s.t.*

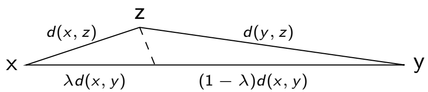
$$z \approx \lambda x + (1-\lambda) y$$

Ahlfors Regularity:



$$|B_{\mathsf{X}_n, r}(x)| \approx \pi r^2 / h^2$$
$$\Downarrow$$
$$\delta = 2, \;\; C \propto 1/h^2$$

Approximate Convexity:



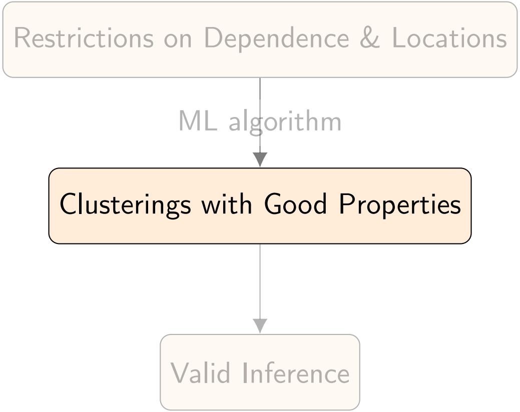
Conditions on $\{\mathcal{C}_n\}_{n\geq 1}$ with G groups

- **Group Balance**:

$$\liminf_{n\to\infty} \min_{C\in\mathcal{C}_n} \frac{|C|}{n} > 0$$

- **Small Boundaries**: $\exists r_n \to \infty$ s.t.

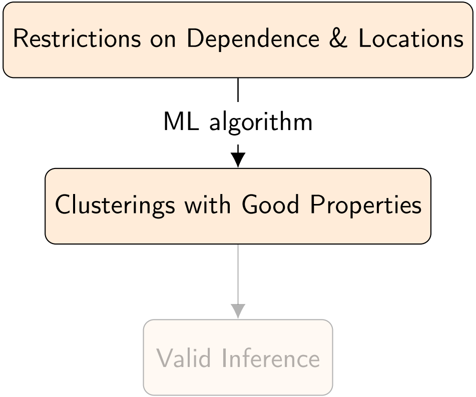
$$\max_{C\in\mathcal{C}_n} |\{x \in C : d(x, X\backslash C) \leq r_n\}| = o(n)$$



Results on Clustering Algorithm

k-medoids: iterate

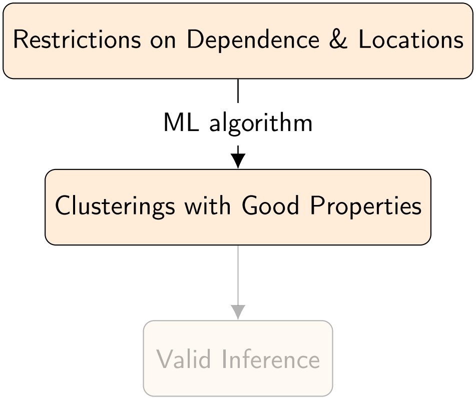
1. Given k centers, assign each point to the closest center
2. Given k clusters, find the center that minimizes sum of distances by swapping



Results on Clustering Algorithm

k-medoids: iterate

1. Given k centers, assign each point to the closest center
2. Given k clusters, find the center that minimizes sum of distances by swapping



Proposition 1

Under Ahlfors Regularity & Approximate Convexity (Condition 2), k-medoids implies Group Balance & Small Boundaries

Main Results

Key (sufficient) requirements for IM & CRS:

Proposition 2

Under Mixing, Group Balance, and Small Boundaries,

$$\begin{bmatrix} \sigma_{n,C_1}^{-1} \sum_{i \in C_1} Z_{i,n} \\ \vdots \\ \sigma_{n,C_G}^{-1} \sum_{i \in C_G} Z_{i,n} \end{bmatrix} \to_d N(0, I_G), \quad \textit{with} \quad \sigma_{n,C}^2 = Var\left[\sum_{i \in C} Z_{i,n}\right].$$

Main Results

Key (sufficient) requirements for IM & CRS:

Proposition 2

Under Mixing, Group Balance, and Small Boundaries,

$$\begin{bmatrix} \sigma_{n,C_1}^{-1} \sum_{i \in C_1} Z_{i,n} \\ \vdots \\ \sigma_{n,C_G}^{-1} \sum_{i \in C_G} Z_{i,n} \end{bmatrix} \to_d N(0, I_G), \quad \textit{with} \quad \sigma_{n,C}^2 = Var\left[\sum_{i \in C} Z_{i,n}\right].$$

Theorem

Under Condition 1 and 2 (and regularity conditions), IM or CRS with a selected clustering has asymptotically correct size:

$$\sup_{\mathcal{C} \in \{\mathcal{C}^{(G)}\}_{G=2}^{\bar{G}}} |E_{P_n}[\phi(\mathcal{C})] - \alpha| \to 0$$

Conclusion

This paper:

- Conditions for well-behaved clustering algorithm
- Formal conditions of valid inference with learned clustering
- Choice of G and partition based on (heuristic) size-power tradeoff

Conclusion

This paper:

- Conditions for well-behaved clustering algorithm
- Formal conditions of valid inference with learned clustering
- Choice of G and partition based on (heuristic) size-power tradeoff

A harder question:

- So far d is assumed to be known
- Would be interesting to know how d can be learned

Thanks!