Beyond Hartigan Consistency

*Merge Distortion Metric* for Hierarchical Clustering

Justin Eldridge, Mikhail Belkin, Yusu Wang
The Ohio State University

July 4, 2015
The goal of clustering:
Identify structure in data by grouping it into clusters
The goal of clustering: Identify structure in data by grouping it into *clusters*.

**Assumption:** data is drawn from some *density*. 
Through clustering we hope to recover the *structure* of the density.
Through clustering we hope to recover the \textit{structure} of the density.

0. What do we mean, precisely, by \textit{structure}?
Through clustering we hope to recover the *structure* of the density.

0. What do we mean, precisely, by *structure*?

1. What *properties* ensure that an algorithm captures it?
Through clustering we hope to recover the *structure* of the density.

0. What do we mean, precisely, by *structure*?

1. What *properties* ensure that an algorithm captures it?

2. How *close* is a clustering to the ideal?
Through clustering we hope to recover the *structure* of the density.

0. What do we mean, precisely, by *structure*?

1. What properties ensure that an algorithm captures it?

2. How close is a clustering to the ideal?

3. Do algorithms with these properties exist?
In this talk...

0. What do we mean, precisely, by structure?
   ▶ The density cluster tree

1. What properties ensure that an algorithm captures it?

2. How close is a clustering to the ideal?

3. Do algorithms with these properties exist?
In this talk...

0. What do we mean, precisely, by structure?
   - The density cluster tree

1. What properties ensure that an algorithm captures it?
   - Previously: Hartigan consistency

2. How close is a clustering to the ideal?

3. Do algorithms with these properties exist?
In this talk...

0. What do we mean, precisely, by structure?
   - The density cluster tree

1. What properties ensure that an algorithm captures it?
   - Previously: Hartigan consistency (insufficient)

2. How close is a clustering to the ideal?

3. Do algorithms with these properties exist?
In this talk...

0. What do we mean, precisely, by structure?
   ▶ The density cluster tree

1. What properties ensure that an algorithm captures it?
   ▶ Previously: Hartigan consistency (insufficient)
   ▶ Introduce: Minimality and Separation

2. How close is a clustering to the ideal?

3. Do algorithms with these properties exist?
In this talk...

0. What do we mean, precisely, by *structure*?
   - The *density cluster tree*

1. What *properties* ensure that an algorithm captures it?
   - Previously: Hartigan consistency (insufficient)
   - Introduce: *Minimality* and *Separation*

2. How *close* is a clustering to the ideal?
   - Previously: Ø

3. Do algorithms with these properties *exist*?
In this talk...

0. What do we mean, precisely, by structure?
   - The density cluster tree

1. What properties ensure that an algorithm captures it?
   - Previously: Hartigan consistency (insufficient)
   - Introduce: Minimality and Separation

2. How close is a clustering to the ideal?
   - Previously: ∅
   - Introduce: Merge distortion metric

3. Do algorithms with these properties exist?
In this talk...

0. What do we mean, precisely, by *structure*?
   - The *density cluster tree*

1. What **properties** ensure that an algorithm captures it?
   - Previously: Hartigan consistency (insufficient)
   - Introduce: *Minimality* and *Separation*

2. How *close* is a clustering to the ideal?
   - Previously: ∅
   - Introduce: *Merge distortion metric*
   - Show: Convergence ⇔ Minimality + Separation

3. Do algorithms with these properties **exist**?
In this talk...

0. What do we mean, precisely, by *structure*?
   ▶ The *density cluster tree*

1. What *properties* ensure that an algorithm captures it?
   ▶ Previously: Hartigan consistency (insufficient)
   ▶ Introduce: *Minimality* and *Separation*

2. How *close* is a clustering to the ideal?
   ▶ Previously: ∅
   ▶ Introduce: *Merge distortion metric*
   ▶ Show: Convergence ⇔ Minimality + Separation

3. Do algorithms with these properties *exist*?
   ▶ Yes. 😊
What *structure* do we wish to recover?

A *cluster* of a density is a *region of high probability*.\(^1\)

\(^1\)Hartigan (1981), Wishart (1969)...
High-density clusters

Connected components of \( \{ f \geq \lambda_1 \} \)?
High-density clusters

Connected components of \( \{ f \geq \lambda_2 \} \)?
High-density clusters

Connected components of \( \{ f \geq \lambda_3 \} \)?
High-density clusters

A *cluster* is a connected component of \( \{ f \geq \lambda \} \) for any \( \lambda > 0 \).
A hierarchy of clusters

Clusters from higher levels nest within clusters from lower levels.
The density cluster tree

This gives rise to a tree structure called the *density cluster tree*.

\[ C_f(\lambda) = \text{connected components of } \{ f \geq \lambda \} \]
What *structure* do we wish to recover?

This *density cluster tree* is what we hope to recover from data.

\[ C_f(\lambda) = \text{connected components of } \{ f \geq \lambda \} \]
Recovering the *density cluster tree* from data

Draw $X_n \sim f$. Algorithm produces a collection of *empirical clusters*.
Recovering the *density cluster tree* from data

These clusters have hierarchical structure.
Recovering the *density cluster tree* from data

Can represent each cluster as a node in a tree.
Recovering the *density cluster tree* from data

In this talk, we’ll omit the redundant labels for clarity.
Recovering the *density cluster tree* from data

In this talk, we’ll omit the redundant labels for clarity.

\[ C_f \]

\[ \hat{C}_{f,n} \]
Recovering the *density cluster tree* from data

The *height* of a node is the density of lowest point it contains.
Recovering the *density cluster tree* from data

The *height* of a node is the density of lowest point it contains.
Recovering the *density cluster tree* from data

**Goal:** As $n \to \infty$, the empirical tree should resemble the true tree.
Recovering the *density cluster tree* from data

**Goal:** As $n \to \infty$, the empirical tree should resemble the true tree.
Recovering the \textit{density cluster tree} from data

**Goal:** As \( n \to \infty \), the empirical tree should resemble the true tree.
1. What properties ensure that an algorithm captures the density cluster tree?
1. What properties ensure that an algorithm captures the density cluster tree?

1. What properties ensure that an algorithm captures the density cluster tree?

- Informally: Clusters which are disjoint in the true tree should be separated in the empirical tree.
Hartigan Consistency

Let $A$ and $B$ be any disjoint ideal clusters.
Hartigan Consistency

Find $A_n :=$ the smallest *empirical cluster* containing $A \cap X_n$. 
Hartigan Consistency

Find $A_n := \text{the smallest empirical cluster containing } A \cap X_n$. 
Hartigan Consistency

Find $B_n :=$ the smallest *empirical cluster* containing $B \cap X_n$. 

---

$B_0 := \emptyset$ and $B_n := B_{n-1} \cup \{z \in X_n \mid D(z, \hat{C}_f, \mu_n) < 2\}$, where $D(z, \hat{C}_f, \mu_n)$ is the distance of $z$ from the empirical distribution $\hat{C}_f, \mu_n$.
Hartigan Consistency

Find $B_n :=$ the smallest *empirical cluster* containing $B \cap X_n$. 

$\mathcal{C}_f \quad \hat{\mathcal{C}}_{f,n}$
Hartigan Consistency

**Hartigan consistency:** As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 
Hartigan Consistency

_Hartigan consistency:_ As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 

$$C_f \quad \hat{C}_{f,n}$$
**Hartigan Consistency**

*Hartigan consistency:* As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 

![Diagram](image-url)
Hartigan Consistency

**Hartigan consistency:** As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 
Hartigan Consistency

Hartigan consistency: As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 
Hartigan consistency: As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 

$C_f$ 

$\hat{C}_{f,n}$
Hartigan Consistency

**Hartigan consistency**: As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 
Hartigan Consistency

**Hartigan consistency:** As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 
Hartigan Consistency

**Hartigan consistency**: As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 
Hartigan Consistency

**Hartigan consistency**: As $n \to \infty$, $\Pr(A_n \text{ is disjoint from } B_n) \to 1$. 
1. What properties ensure that an algorithm captures the *density cluster tree*?
1. What **properties** ensure that an algorithm captures the *density cluster tree*?
   - Hartigan consistency
1. What properties ensure that an algorithm captures the *density cluster tree*?
   - Hartigan consistency
   - We’ll see shortly that Hartigan consistency is insufficient
1. What properties ensure that an algorithm captures the *density cluster tree*?
   - *Hartigan consistency*
   - We’ll see shortly that Hartigan consistency is *insufficient*
   - But it is still a *desirable* property of an algorithm...
1. What properties ensure that an algorithm captures the *density cluster tree*?
   - *Hartigan consistency*
   - We’ll see shortly that Hartigan consistency is *insufficient*
   - But it is still a *desirable* property of an algorithm...

2. How close is a clustering to the ideal density cluster tree?
1. What properties ensure that an algorithm captures the density cluster tree?
   ▶ Hartigan consistency
   ▶ We’ll see shortly that Hartigan consistency is insufficient
   ▶ But it is still a desirable property of an algorithm...

2. How close is a clustering to the ideal density cluster tree?
   ▶ Hartigan consistency is a limit property: doesn’t quantify distance to true tree.
1. What **properties** ensure that an algorithm captures the *density cluster tree*?
   
   - *Hartigan consistency*
   - We’ll see shortly that Hartigan consistency is **insufficient**
   - But it is still a *desirable* property of an algorithm...

2. How **close** is a clustering to the ideal density cluster tree?
   
   - Hartigan consistency is a *limit property*: doesn’t quantify distance to true tree.

3. Do algorithms **exist** which are *Hartigan consistent*?
1. What properties ensure that an algorithm captures the density cluster tree?
   ▶ Hartigan consistency
   ▶ We’ll see shortly that Hartigan consistency is insufficient
   ▶ But it is still a desirable property of an algorithm...

2. How close is a clustering to the ideal density cluster tree?
   ▶ Hartigan consistency is a limit property: doesn’t quantify distance to true tree.

3. Do algorithms exist which are Hartigan consistent?
   ▶ Hartigan analyzed single linkage clustering, showed that it is not consistent in $d > 1$. 
1. What properties ensure that an algorithm captures the density cluster tree?
   - Hartigan consistency
   - We’ll see shortly that Hartigan consistency is insufficient
   - But it is still a desirable property of an algorithm...

2. How close is a clustering to the ideal density cluster tree?
   - Hartigan consistency is a limit property: doesn’t quantify distance to true tree.

3. Do algorithms exist which are Hartigan consistent?
   - Hartigan analyzed single linkage clustering, showed that it is not consistent in $d > 1$.
   - 30 years pass...
1. What properties ensure that an algorithm captures the density cluster tree?
   - Hartigan consistency
   - We’ll see shortly that Hartigan consistency is insufficient
   - But it is still a desirable property of an algorithm...

2. How close is a clustering to the ideal density cluster tree?
   - Hartigan consistency is a limit property: doesn’t quantify distance to true tree.

3. Do algorithms exist which are Hartigan consistent?
   - Hartigan analyzed single linkage clustering, showed that it is not consistent in $d > 1$.
   - 30 years pass...
   - Several algorithms shown to be consistent, including robust single linkage (Chaudhuri and Dasgupta, 2010) and tree pruning (Kpotufe and von Luxburg, 2011)
Hartigan consistency is insufficient

Hartigan lacks a strong notion of *connectedness.*
Hartigan consistency is insufficient

Hartigan lacks a strong notion of connectedness.
Hartigan consistency is insufficient

Hartigan lacks a strong notion of \textit{connectedness}.
Hartigan consistency is insufficient

Hartigan lacks a strong notion of *connectedness*. 
Hartigan consistency is insufficient

Hartigan lacks a strong notion of connectedness.
Hartigan consistency is insufficient

This tree does not violate Hartigan consistency!
Hartigan consistency is insufficient

This tree does not violate Hartigan consistency!
Hartigan consistency is insufficient

This tree does not violate Hartigan consistency!
Hartigan consistency is insufficient

What about this tree?
Hartigan consistency is insufficient

What about this tree? Also consistent!
**Hartigan consistency** is insufficient

A tree can be *Hartigan consistent* yet very different from the true tree.

\[ C_f \]

\[ \hat{C}_{f,n} \]
Beyond *Hartigan consistency*

- *Hartigan consistency* lacks *connectedness*
Beyond *Hartigan consistency*

- *Hartigan consistency* lacks connectedness
- We need a different, *stronger* notion of consistency
Beyond *Hartigan consistency*

- *Hartigan consistency* lacks *connectedness*
- We need a different, *stronger* notion of consistency
- We introduce *minimality* to address connectedness
Beyond *Hartigan consistency*

- *Hartigan consistency* lacks connectedness
- We need a different, stronger notion of consistency
- We introduce *minimality* to address connectedness
- We introduce *separation* as a weaker form of *Hartigan’s* notion
Beyond *Hartigan consistency*

- *Hartigan consistency* lacks connectedness
- We need a different, stronger notion of consistency
- We introduce *minimality* to address connectedness
- We introduce *separation* as a weaker form of *Hartigan*’s notion
- Together they’ll imply *Hartigan consistency*
Minimality

$C \cap X_n$ should be connected at $\lambda - \delta$, with $\delta \to 0$ as $n \to \infty$
Minimality

$C \cap X_n$ should be connected at $\lambda - \delta$, with $\delta \to 0$ as $n \to \infty$
Minimality

$C \cap X_n$ should be connected at $\lambda - \delta$, with $\delta \to 0$ as $n \to \infty$
Minimality

$\mathcal{C} \cap X_n$ should be connected at $\lambda - \delta$, with $\delta \to 0$ as $n \to \infty$
Minimality

\[ \hat{C}_{f,n} \] ensures \textit{minimality} if given any cluster \( C \) of \( \{f \geq \lambda\} \), \( C \cap X_n \) is connected at level \( \lambda - \delta \) for any \( \delta > 0 \) as \( n \to \infty \).
Separation

$A \cap X_n$ and $B \cap X_n$ should be separated at $\mu + \delta$, with $\delta \to 0$ as $n \to \infty$
**Separation**

$A \cap X_n$ and $B \cap X_n$ should be separated at $\mu + \delta$, with $\delta \to 0$ as $n \to \infty$. 

\[ \mathcal{C}_f \]

\[ \hat{\mathcal{C}}_{f,n} \]
Separation

$A \cap X_n$ and $B \cap X_n$ should be separated at $\mu + \delta$, with $\delta \to 0$ as $n \to \infty$
**Separation**

$A \cap X_n$ and $B \cap X_n$ should be separated at $\mu + \delta$, with $\delta \to 0$ as $n \to \infty$. 

\[ a_1 \quad a_2 \quad a_3 \]

\[ b_1 \quad b_2 \]

$C_f$  

$\hat{C}_{f,n}$
Separation

$\hat{C}_{f,n}$ ensures separation if given any disjoint clusters $A$ and $B$ of $\{f \geq \lambda\}$ merging at $\mu$, $A \cap X_n$ and $B \cap X_n$ are separated at level $\mu + \delta$ for any $\delta > 0$ as $n \to \infty$. 
Theorem

If a clustering method ensures minimality and separation, then it is Hartigan consistent.

Minimality and Separation $\implies$ Hartigan Consistency

Hartigan Consistency $\iff$ Minimality and Separation
1. What properties ensure that an algorithm captures the *density cluster tree*?
1. What properties ensure that an algorithm captures the *density cluster tree*?
   ▶ We introduce *Minimality* and *Separation*.
1. What properties ensure that an algorithm captures the density cluster tree?
   - We introduce *Minimality* and *Separation*
   - *Minimality* addresses shortcomings of Hartigan consistency
1. What properties ensure that an algorithm captures the density cluster tree?
   - We introduce *Minimality* and *Separation*
   - *Minimality* addresses shortcomings of *Hartigan consistency*
   - *Minimality* + *Separation* $\implies$ *Hartigan Consistency*
1. What properties ensure that an algorithm captures the *density cluster tree*?
   ▶ We introduce *Minimality* and *Separation*
   ▶ *Minimality* addresses shortcomings of *Hartigan consistency*
   ▶ *Minimality + Separation* → *Hartigan Consistency*

2. How close is a clustering to the ideal density cluster tree?
1. What properties ensure that an algorithm captures the *density cluster tree*?
   - We introduce *Minimality* and *Separation*
   - *Minimality* addresses shortcomings of *Hartigan consistency*
   - *Minimality* $+$ *Separation* $\implies$ Hartigan Consistency

2. How close is a clustering to the ideal density cluster tree?
   - We now introduce a *merge distortion metric* on cluster trees.
1. **What properties** ensure that an algorithm captures the *density cluster tree*?
   - We introduce *Minimality* and *Separation*
   - *Minimality* addresses shortcomings of *Hartigan consistency*
   - *Minimality + Separation* $\implies$ *Hartigan Consistency*

2. **How close** is a clustering to the ideal density cluster tree?
   - We now introduce a *merge distortion metric* on cluster trees.
   - Convergence will imply *minimality* and *separation*. 
Ideal and empirical merge height

The ideal merge height: \( m(a, b) \)

The empirical merge height: \( \hat{m}(a, b) \)

Minimality: \( \hat{m}(a, b) > m(a, b) \)

Separation: \( \hat{m}(a, b) < m(a, b) + \)
Ideal and empirical merge height

- The ideal merge height: $m(a, b)$
- The empirical merge height: $\hat{m}(a, b)$
Ideal and empirical merge height

- The **ideal merge height**: $m(a, b)$
- The **empirical merge height**: $\hat{m}(a, b)$
- **Minimality**: $\hat{m}(a, b) > m(a, b) - \delta$, with $\delta \to 0$
Ideal and empirical merge height

- The **ideal merge height**: $m(a, b)$
- The **empirical merge height**: $\hat{m}(a, b)$
- **Minimality**: $\hat{m}(a, b) > m(a, b) - \delta$, with $\delta \to 0$
- **Separation**: $\hat{m}(a, b) < m(a, b) + \delta$, with $\delta \to 0$
The ideal merge height: \( m(a, b) \)
The empirical merge height: \( \hat{m}(a, b) \)
Minimality: \( \hat{m}(a, b) > m(a, b) - \delta \), with \( \delta \to 0 \)
Separation: \( \hat{m}(a, b) < m(a, b) + \delta \), with \( \delta \to 0 \)
Together: \( \hat{m}(a, b) \to m(a, b) \) as \( n \to \infty \)
Ideal and empirical merge height

- The **ideal merge height**: \( m(a, b) \)
- The **empirical merge height**: \( \hat{m}(a, b) \)
- **Minimality**: \( \hat{m}(a, b) > m(a, b) - \delta \), with \( \delta \to 0 \)
- **Separation**: \( \hat{m}(a, b) < m(a, b) + \delta \), with \( \delta \to 0 \)
- Together: \( \hat{m}(a, b) \to m(a, b) \) as \( n \to \infty \)
Ideal and empirical merge height

- The **ideal merge height**: $m(a, b)$
- The **empirical merge height**: $\hat{m}(a, b)$
- **Minimality**: $\hat{m}(a, b) > m(a, b) - \delta$, with $\delta \to 0$
- **Separation**: $\hat{m}(a, b) < m(a, b) + \delta$, with $\delta \to 0$
- Together: $\hat{m}(a, b) \to m(a, b)$ as $n \to \infty$
Ideal and empirical merge height

- The *ideal merge height*: $m(a, b)$
- The *empirical merge height*: $\hat{m}(a, b)$
- **Minimality**: $\hat{m}(a, b) > m(a, b) - \delta$, with $\delta \to 0$
- **Separation**: $\hat{m}(a, b) < m(a, b) + \delta$, with $\delta \to 0$
- Together: $\hat{m}(a, b) \to m(a, b)$ as $n \to \infty$
Ideal and empirical merge height

- The ideal merge height: $m(a, b)$
- The empirical merge height: $\hat{m}(a, b)$
- **Minimality**: $\hat{m}(a, b) > m(a, b) - \delta$, with $\delta \to 0$
- **Separation**: $\hat{m}(a, b) < m(a, b) + \delta$, with $\delta \to 0$
- Together: $\hat{m}(a, b) \to m(a, b)$ as $n \to \infty$
Ideal and empirical merge height

We define the *merge distortion metric* between the density cluster tree and its estimate as:

\[ d(C_f, \hat{C}_{f,n}) = \max_{x, x' \in X_n} |m(x, x') - \hat{m}(x, x')| . \]
Theorem

Convergence of $\hat{C}_{f,n} \to C_f$

is equivalent to

\textit{uniform minimality} $+$ \textit{uniform separation}.
We have introduced *minimality, separation, and the merge distortion metric*...
We have introduced *minimality, separation,* and the *merge distortion metric*...

Do algorithms exist which have these properties/converge to the true density cluster tree?

- We analyze two:
  - Robust single linkage from (Chaudhuri and Dasgupta, 2010)
  - Split tree-based clustering from computational topology
Convergence of robust single linkage

- Robust single linkage (Chaudhuri and Dasgupta, 2010): elegant generalization of single linkage which incorporates density information
- Authors proved that it is Hartigan consistent
- Also showed that clusters not only separated, but connected at about the right level
Convergence of robust single linkage

- Robust single linkage (Chaudhuri and Dasgupta, 2010): elegant generalization of single linkage which incorporates density information
- Authors proved that it is Hartigan consistent
- Also showed that clusters not only separated, but connected at about the right level

Theorem
Suppose $f$ is $c$-Lipschitz, compactly supported, and for any $\lambda$, \{ $f \geq \lambda$ \} has finitely-many connected components. Then:
- Robust single linkage converges to the true cluster tree in the merge distortion metric.
Future work

- What other algorithms converge in the *merge distortion metric*?
- $\ell_2$ variant of the metric?
- Fast algorithms for approximating the distance.
- Hierarchical clustering without a density – how do we define distance?
Summary

1. What properties ensure that an algorithm captures the density cluster tree?
   ▶ We introduce Minimality and Separation
   ▶ Minimality addresses shortcomings of Hartigan consistency
   ▶ Minimality + Separation $\implies$ Hartigan Consistency
Summary

1. What properties ensure that an algorithm captures the *density cluster tree*?
   - We introduce *Minimality* and *Separation*
   - *Minimality* addresses shortcomings of *Hartigan consistency*
   - *Minimality + Separation → Hartigan Consistency*

2. How close is a clustering to the ideal density cluster tree?
   - We introduced a *merge distortion metric* on cluster trees.
   - Convergence implies *minimality* and *separation*. 
Summary

1. What properties ensure that an algorithm captures the *density cluster tree*?
   - We introduce *Minimality* and *Separation*
   - *Minimality* addresses shortcomings of *Hartigan consistency*
   - *Minimality* + *Separation* $\implies$ *Hartigan Consistency*

2. How close is a clustering to the ideal density cluster tree?
   - We introduced a *merge distortion metric* on cluster trees.
   - Convergence implies *minimality* and *separation*.

3. Do algorithms exist which have these properties/converge to the true density cluster tree?
   - Yes:
     - Robust single linkage (Chaudhuri and Dasgupta, 2010)
     - Split-tree-based algorithm.
Thank you!