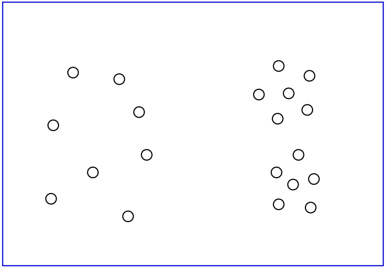
Cluster trees, neighborhood graphs, and continuum percolation

Sanjoy Dasgupta

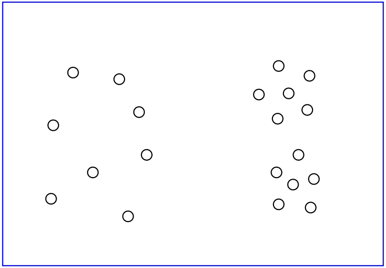
University of California, San Diego

Part I: Cluster trees

Clustering in $\mathbb{R}^d$



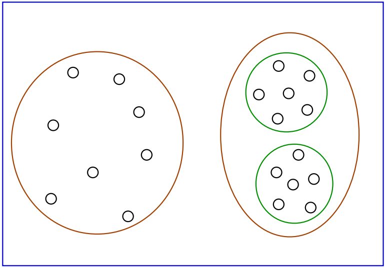
Clustering in $\mathbb{R}^d$



Two common uses of clustering:

- Vector quantization
- Finding meaningful structure in data

Clustering in $\mathbb{R}^d$



Two common uses of clustering:

- Vector quantization
- Finding meaningful structure in data

A hierarchical clustering algorithm

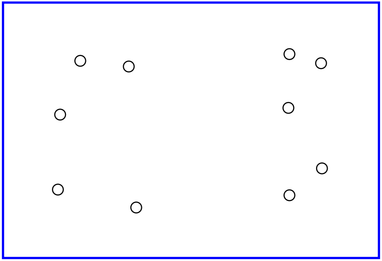
Joseph Kruskal, 1928-2010

The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

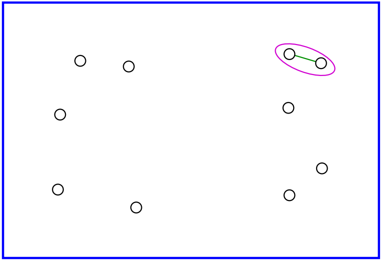


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

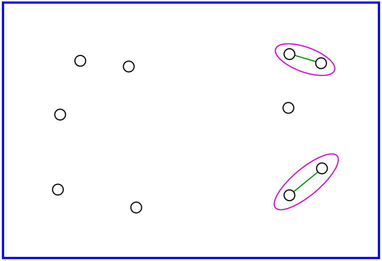


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

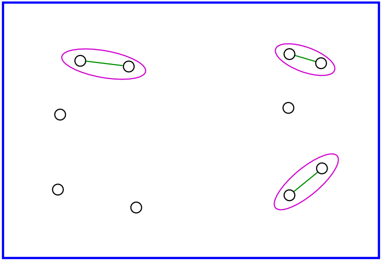


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

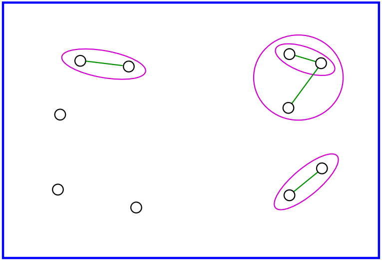


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

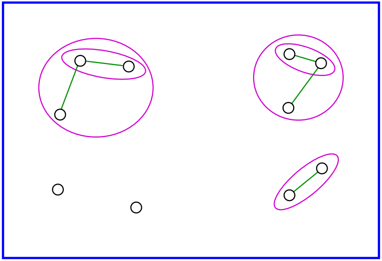


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
    - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

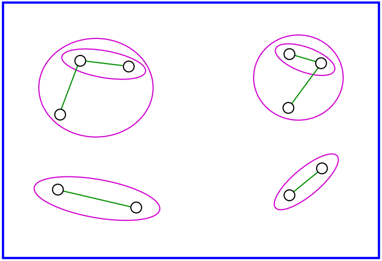


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

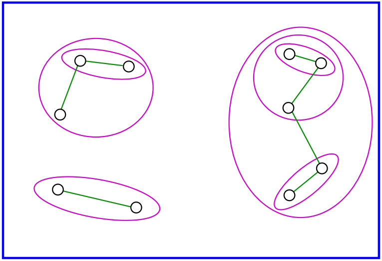


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

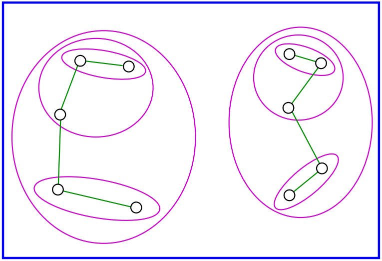


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010

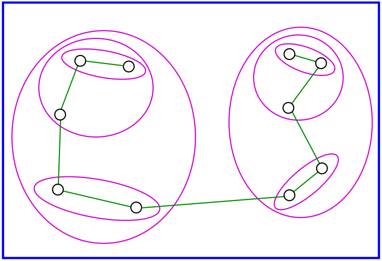


The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

A hierarchical clustering algorithm

Joseph Kruskal, 1928-2010



The single linkage algorithm:

- Start with each point in its own, singleton, cluster
- Repeat until there is just one cluster:
  - Merge the two clusters with the closest pair of points
- Disregard singleton clusters

Statistical theory in clustering

Data points $X_1, \ldots, X_n$ are independent random draws from an unknown density f on $\mathbb{R}^d$

Statistical theory in clustering

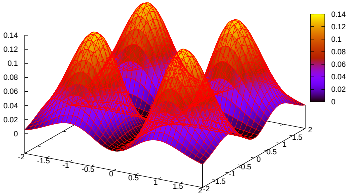
Data points $X_1, \ldots, X_n$ are independent random draws from an unknown density f on $\mathbb{R}^d$

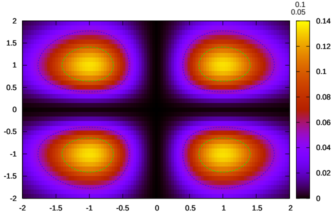
- Different random sample $\Rightarrow$ similar clustering (if n is large)
- As $n \to \infty$: approach "natural clusters" of f

Statistical theory in clustering

Data points $X_1, \ldots, X_n$ are independent random draws from an unknown density f on $\mathbb{R}^d$

- Different random sample $\Rightarrow$ similar clustering (if n is large)
- As $n \to \infty$: approach "natural clusters" of f

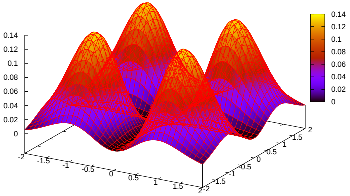


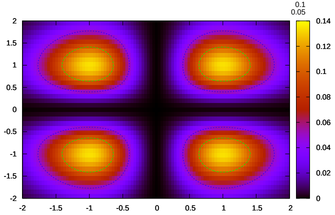


Statistical theory in clustering

Data points $X_1, \ldots, X_n$ are independent random draws from an unknown density f on $\mathbb{R}^d$

- Different random sample $\Rightarrow$ similar clustering (if n is large)
- As $n \to \infty$: approach "natural clusters" of f

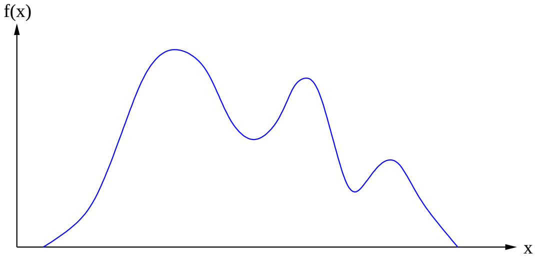




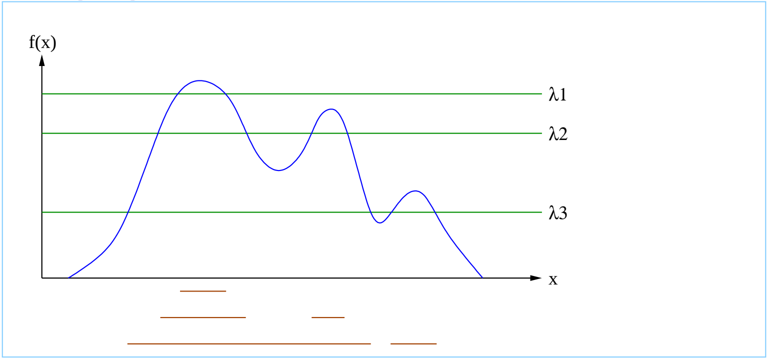
cluster $\equiv$ connected component of $\{x : f(x) \geq \lambda\}$, any $\lambda > 0$

These clusters form an infinite hierarchy, the *cluster tree*.

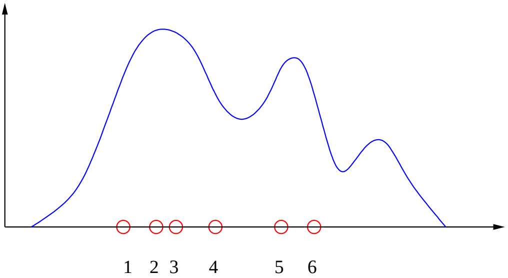
Converging to the cluster tree



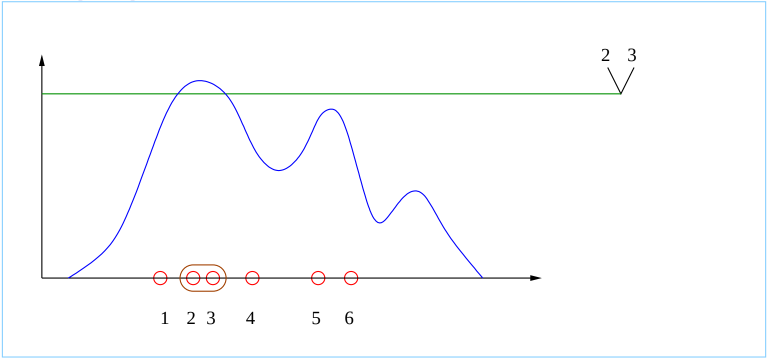
Converging to the cluster tree



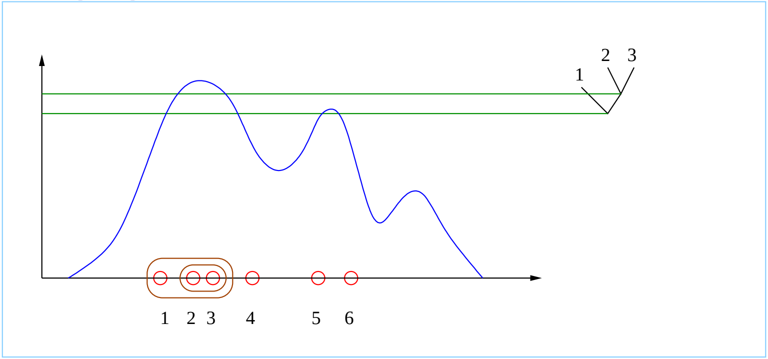
Converging to the cluster tree



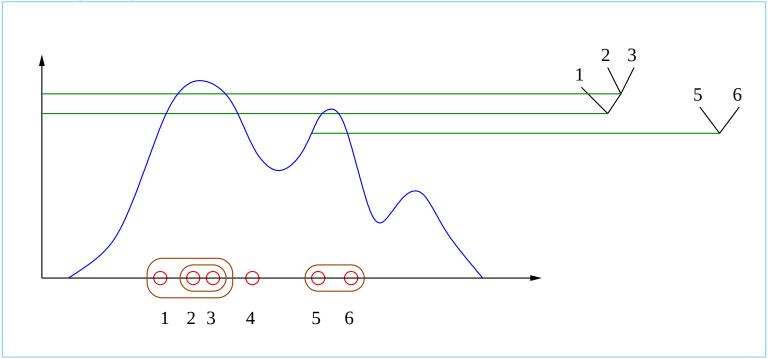
Converging to the cluster tree



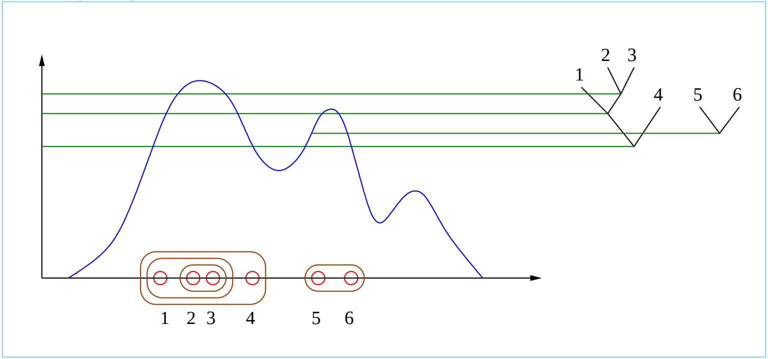
Converging to the cluster tree



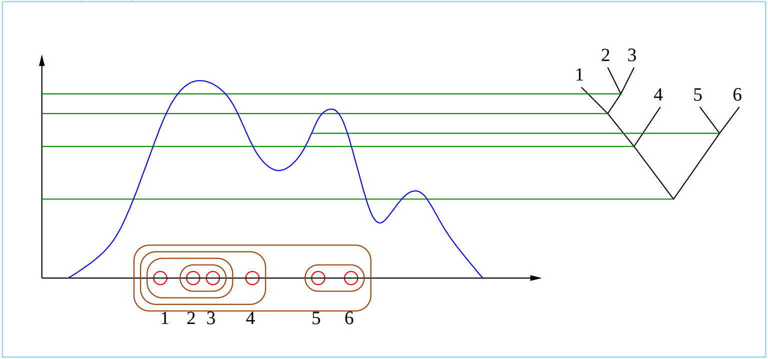
Converging to the cluster tree



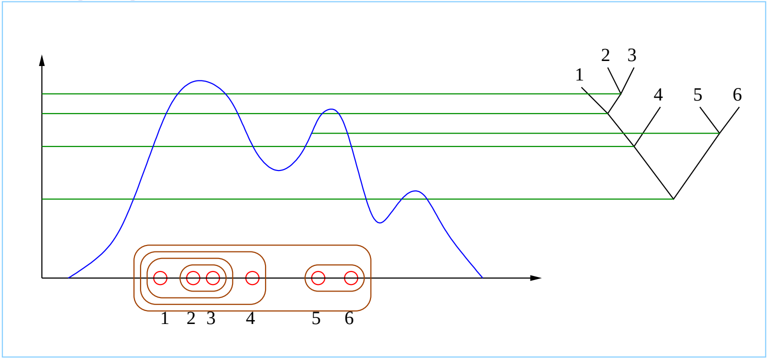
Converging to the cluster tree



Converging to the cluster tree



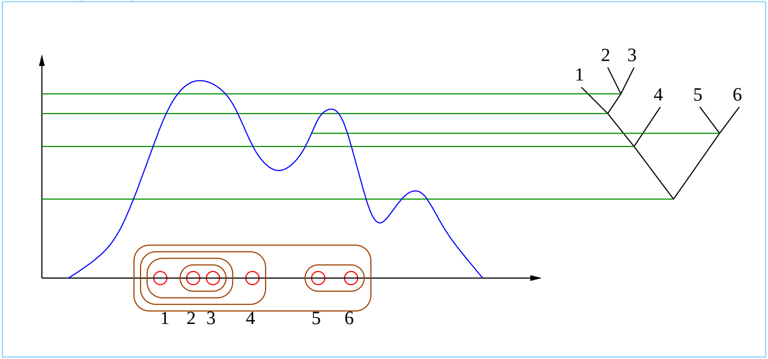
Converging to the cluster tree



Consistency: Let A, A' be connected components of $\{f \geq \lambda\}$, for any λ. In the tree constructed from n data points X_n, let A_n be the smallest cluster containing $A \cap X_n$; likewise A'_n. Then:

$$\lim_{n \to \infty} \text{Prob}[A_n \text{ is disjoint from } A'_n] = 1$$

Converging to the cluster tree



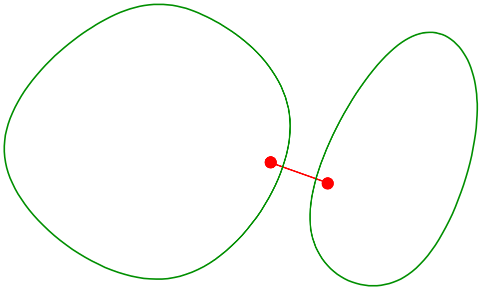
Consistency: Let A, A' be connected components of $\{f \geq \lambda\}$, for any λ. In the tree constructed from n data points X_n, let A_n be the smallest cluster containing $A \cap X_n$; likewise A'_n. Then:

$$\lim_{n \to \infty} \text{Prob}[A_n \text{ is disjoint from } A'_n] = 1$$

Hartigan 1975: Single linkage is consistent for $d = 1$.

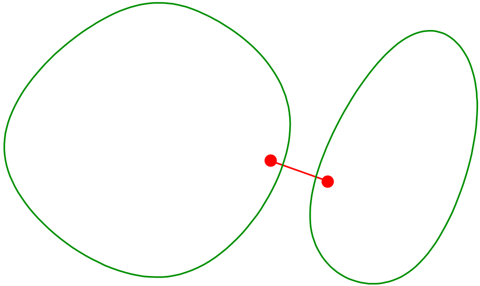
Higher dimension

Hartigan 1982: Single linkage is not consistent for $d > 1$.



Higher dimension

Hartigan 1982: Single linkage is not consistent for $d > 1$.



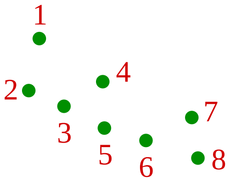
Chaudhuri-D '10: a simple variant of single linkage is consistent in any dimension, with a good finite sample convergence rate.

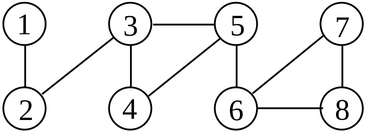
Related work

- Single linkage satisfies a partial consistency property
  Penrose 1995
- Algorithms to capture a user-specified level set $\{x : f(x) \geq \lambda\}$
  Maier-Hein-von Luxburg 2009, Rinaldo-Wasserman 2009,
  Singh-Scott-Nowak 2009
- Other estimators for the cluster tree
  Wishart 1969 (very similar to ours), Wong and Lane 1983,
  Stuetzle and Nugent 2010

Part II: Near neighbor graphs

Capturing a data set's local structure





An undirected graph with

- A node for each data point
- Edges between "neighboring" points

Uses: clustering, semisupervised learning, embeddings, regularization, ...

Two types of neighborhood graph

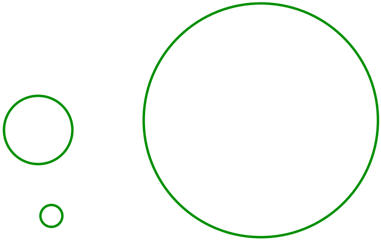
Connect points at distance $\leq r$

Connect each point to its k nearest neighbors

Two types of neighborhood graph

Connect points at distance $\leq r$

Problem: clusters at different scales

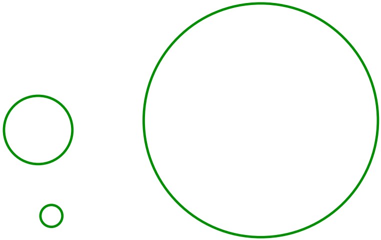


Connect each point to its k nearest neighbors

Two types of neighborhood graph

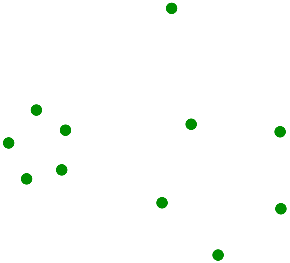
Connect points at distance $\leq r$

Problem: clusters at different scales

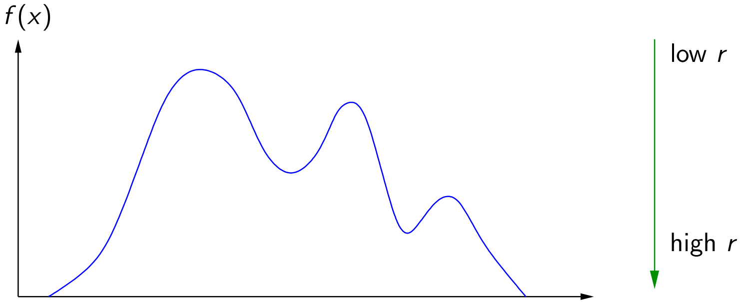


Connect each point to its k nearest neighbors

Problem: spurious connections

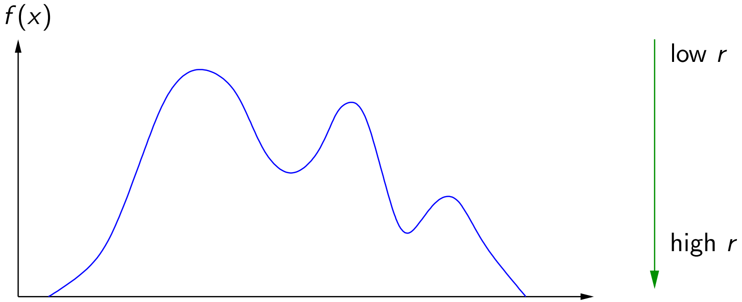


Single linkage, amended



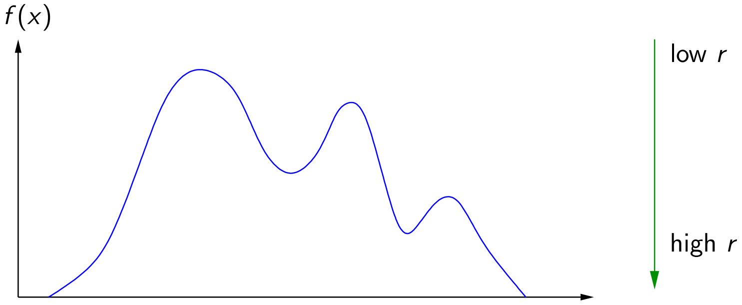
- For each x_i: set $r(x_i)$ = distance to nearest neighbor
- As r increases from 0 to ∞:
  - Construct graph G_r:
    Nodes $\{x_i : r(x_i) \leq r\}$
    Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq r$
  - Output the connected components of G_r

Single linkage, amended



- For each x_i: set $r(x_i)$ = distance to kth nearest neighbor
- As r increases from 0 to ∞:
  - Construct graph G_r:
    Nodes $\{x_i : r(x_i) \leq r\}$
    Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
  - Output the connected components of G_r

Single linkage, amended

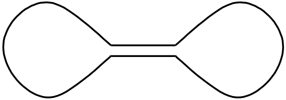


- For each x_i: set $r(x_i)$ = distance to kth nearest neighbor
- As r increases from 0 to ∞:
  - Construct graph G_r:
    Nodes $\{x_i : r(x_i) \leq r\}$
    Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
  - Output the connected components of G_r

With $\sqrt{2} \leq \alpha \leq 2$ and $k \sim d \log n$, this is consistent for any d!

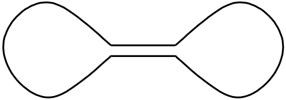
Which clusters are most salient?

Effect 1: thin bridges



Which clusters are most salient?

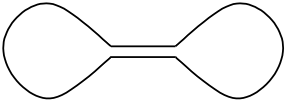
Effect 1: thin bridges



For any set Z, let Z_σ be all points within distance σ of it.

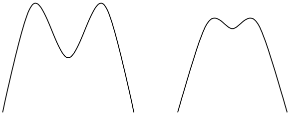
Which clusters are most salient?

Effect 1: thin bridges



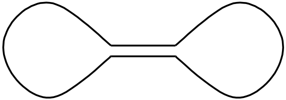
For any set Z, let Z_σ be all points within distance σ of it.

Effect 2: density dip



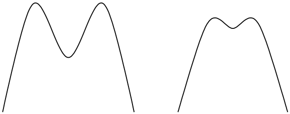
Which clusters are most salient?

Effect 1: thin bridges

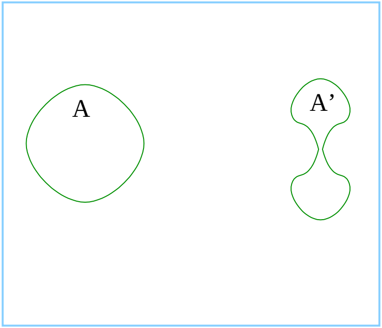


For any set Z, let Z_σ be all points within distance σ of it.

Effect 2: density dip

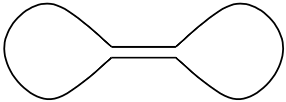


A and A' are (σ, ϵ)-separated if:
- separated by some set S
- max density in $S_\sigma \leq (1-\epsilon)$(min density in A_σ, A'_σ)



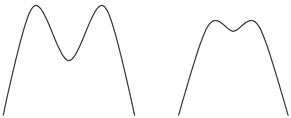
Which clusters are most salient?

Effect 1: thin bridges

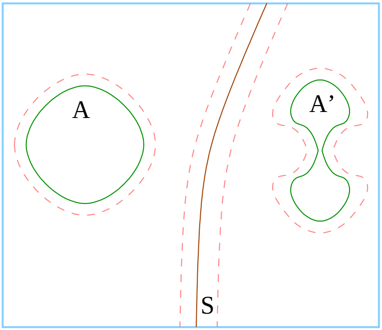


For any set Z, let Z_σ be all points within distance σ of it.

Effect 2: density dip

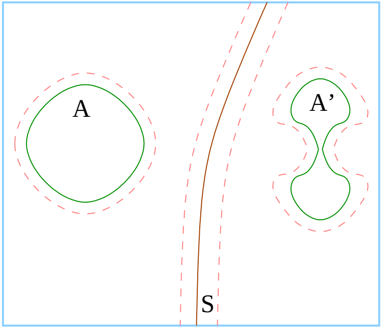


A and A' are (σ, ϵ)-separated if:
- separated by some set S
- max density in $S_\sigma \leq (1-\epsilon)$(min density in A_σ, A'_σ)



Rate of convergence

A and A' are (σ, ϵ)-separated if:
- separated by some set S
- max density in $S_\sigma \leq$ $(1-\epsilon)$(min density in A_σ, A'_σ)



With high probability, for all connected sets A, A':
if A, A' are (σ, ϵ)-separated, and have minimum density λ, then for

$$n \geq \frac{d}{\lambda \epsilon^2 \sigma^d}$$

there will be some intermediate graph G_r such that:

- There is no path between A and A' in G_r
- A and A' are individually connected in G_r

Part III: Continuum percolation

Connectivity in random graphs

Erdos-Renyi random graphs

- n nodes
- Edges placed at random: between each pair of nodes, independently, an edge with probability p

Connectivity in random graphs

Erdos-Renyi random graphs

- n nodes
- Edges placed at random: between each pair of nodes, independently, an edge with probability p

Random geometric graphs

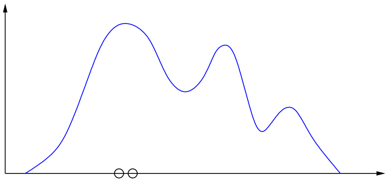
- n points randomly chosen from an unknown density
- One node per point
- Edges between nodes that are nearby in some sense

Identifying high-density regions

Algorithm:
For each i: $r(x_i)$ = dist to kth nearest neighbor
As r increases from 0 to ∞:

- Construct graph G_r:
  Nodes $\{x_i : r(x_i) \leq r\}$
  Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
- Output the connected components of G_r

Single linkage has $k = 1$,
hoping: low $r \Leftrightarrow$ high density

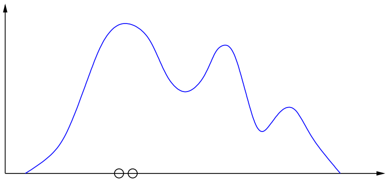


Identifying high-density regions

Algorithm:
For each i: $r(x_i)$ = dist to kth nearest neighbor
As r increases from 0 to ∞:

- Construct graph G_r:
  Nodes $\{x_i : r(x_i) \leq r\}$
  Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
- Output the connected components of G_r

Single linkage has $k = 1$, hoping: low $r \Leftrightarrow$ high density



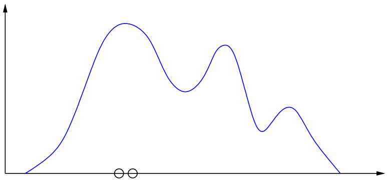
Vapnik-Chervonenkis bounds:
for *every* ball B in $\mathbb{R}^d$,
$\#$ pts in $B =$
$f(B) \cdot n \pm \sqrt{f(B) \cdot n \cdot d \log n}$.

Identifying high-density regions

Algorithm:
For each i: $r(x_i)$ = dist to kth nearest neighbor
As r increases from 0 to ∞:

- Construct graph G_r:
  Nodes $\{x_i : r(x_i) \leq r\}$
  Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
- Output the connected components of G_r

Single linkage has $k = 1$, hoping: low $r \Leftrightarrow$ high density

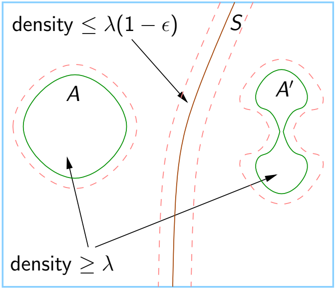


Vapnik-Chervonenkis bounds: for *every* ball B in $\mathbb{R}^d$,
$\#$ pts in $B =$
$f(B) \cdot n \pm \sqrt{f(B) \cdot n \cdot d \log n}$.

Moral: choose $k \geq d \log n$.

Separation

A, A' are (σ, ϵ)-separated.



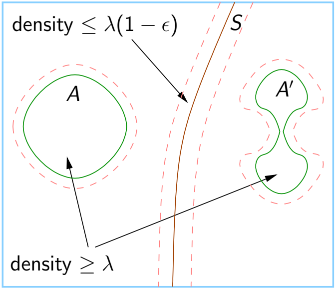
(Buffer zone has width σ.)

There is some value r at which:

1. Every point in A, A' has $\geq k$ points within distance r, and is thus a node in G_r
2. Any point in S_σ has $< k$ points within distance r, and thus isn't a node in G_r
3. $r \leq \sigma/2$

Separation

A, A' are (σ, ϵ)-separated.



(Buffer zone has width σ.)

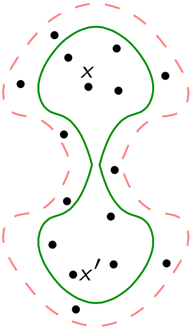
There is some value r at which:

1. Every point in A, A' has $\geq k$ points within distance r, and is thus a node in G_r
2. Any point in S_σ has $< k$ points within distance r, and thus isn't a node in G_r
3. $r \leq \sigma/2$

A is disconnected from A' in G_r

Connectedness

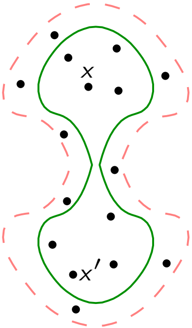
At this particular scale r, every point in A and A' (or within distance r of A, A') is active.



But, are these points connected in G_r?

Connectedness

At this particular scale r, every point in A and A' (or within distance r of A, A') is active.

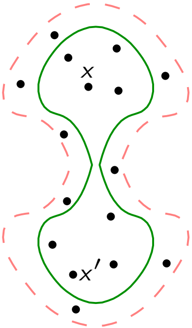


But, are these points connected in G_r?

The worst case:

Connectedness

At this particular scale r, every point in A and A' (or within distance r of A, A') is active.



But, are these points connected in G_r?

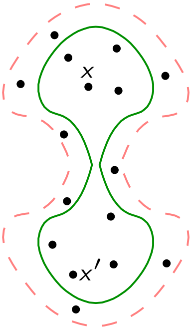
The worst case:

This is where α comes in:

Graph G_r:
Nodes $\{x_i : r(x_i) \leq r\}$
Edges (x_i, x_j) for $\|x_i - x_j\| \leq \alpha r$

Connectedness

At this particular scale r, every point in A and A' (or within distance r of A, A') is active.



But, are these points connected in G_r?

The worst case:

This is where α comes in:

Graph G_r: *Nodes* $\{x_i : r(x_i) \leq r\}$ *Edges* (x_i, x_j) for $\|x_i - x_j\| \leq \alpha r$

- $\alpha = 2$: easy to show connectivity
- $\alpha = \sqrt{2}$: our result

Connectedness (cont'd)

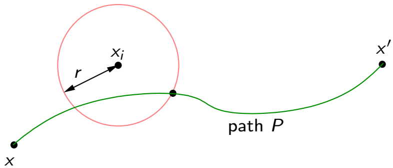
Proof sketch

x, x' are in cluster A, so there is a path P between them.

We'll exhibit data points $x_0 = x, x_1, \ldots, x_\ell = x'$ such that:

- The x_i are within distance r of P (and thus of A, and thus are active in G_r)
- $\|x_i - x_{i+1}\| \leq \alpha r$

So x is connected to x' in G_r.



Connectedness (cont'd)

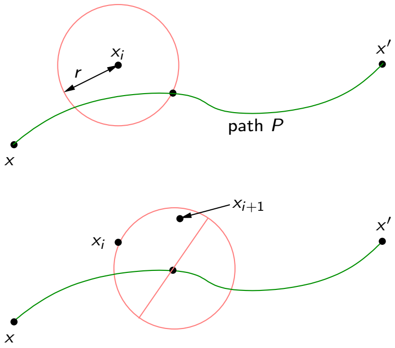
Proof sketch

x, x' are in cluster A, so there is a path P between them.

We'll exhibit data points $x_0 = x, x_1, \ldots, x_\ell = x'$ such that:

- The x_i are within distance r of P (and thus of A, and thus are active in G_r)
- $\|x_i - x_{i+1}\| \leq \alpha r$

So x is connected to x' in G_r.



Connectedness (cont'd)

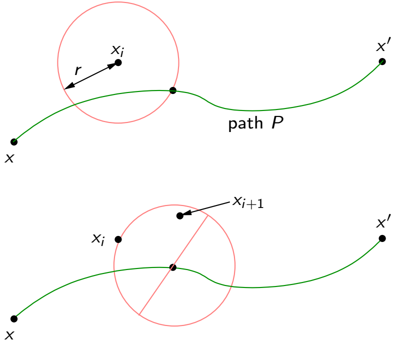
Proof sketch

x, x' are in cluster A, so there is a path P between them.

We'll exhibit data points $x_0 = x, x_1, \ldots, x_\ell = x'$ such that:

- The x_i are within distance r of P (and thus of A, and thus are active in G_r)
- $\|x_i - x_{i+1}\| \leq \alpha r$

So x is connected to x' in G_r.



Therefore $\|x_i - x_{i+1}\| \leq r\sqrt{2}$.

Connectedness (cont'd)

Proof sketch

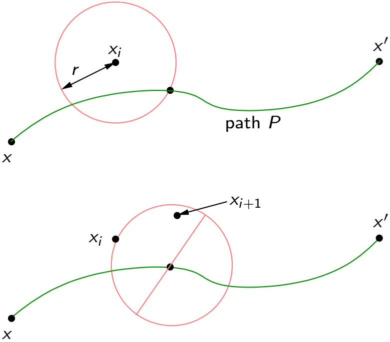
x, x' are in cluster A, so there is a path P between them.

We'll exhibit data points $x_0 = x, x_1, \ldots, x_\ell = x'$ such that:

- The x_i are within distance r of P (and thus of A, and thus are active in G_r)
- $\|x_i - x_{i+1}\| \leq \alpha r$

So x is connected to x' in G_r.

Open problem: will $\alpha = 1$ work?



Therefore $\|x_i - x_{i+1}\| \leq r\sqrt{2}$.

Lower bound via Fano's inequality

A game played with a predefined class of distributions $\{\theta_1, \ldots, \theta_\ell\}$.

- Nature picks $I \in \{1, 2, \ldots, \ell\}$
- Player is given n iid samples from from θ_I
- Player then guesses the identity of I

Lower bound via Fano's inequality

A game played with a predefined class of distributions $\{\theta_1, \ldots, \theta_\ell\}$.

- Nature picks $I \in \{1, 2, \ldots, \ell\}$
- Player is given n iid samples from from θ_I
- Player then guesses the identity of I

Theorem: If Nature chooses I uniformly at random, then the Player must draw at least

$$n \geq \frac{\log \ell}{2\beta}$$

samples in order to guess correctly with probability $\geq 1/2$, where

$$\beta = \frac{1}{\ell^2} \sum_{i,j=1}^{\ell} K(\theta_i, \theta_j).$$

An alternative cluster tree estimator

Original scheme constructs a hierarchy of neighborhood r-graphs:

- For each x_i: set $r_k(x_i)$ = distance to kth nearest neighbor
- As r increases from 0 to ∞:
  - Construct graph G_r:
    Nodes $\{x_i : r_k(x_i) \leq r\}$
    Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
  - Output the connected components of G_r

An alternative cluster tree estimator

Original scheme constructs a hierarchy of neighborhood r-graphs:

- For each x_i: set $r_k(x_i)$ = distance to kth nearest neighbor
- As r increases from 0 to ∞:
  - Construct graph G_r:
    Nodes $\{x_i : r_k(x_i) \leq r\}$
    Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
  - Output the connected components of G_r

[Kpotufe-von Luxburg 2011] Instead of G_r, use graph G_r^{NN}:

- Same nodes, $\{x_i : r(x_i) \leq r\}$
- Edges (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha \min(r_k(x_i), r_k(x_j))$

Similar rates of convergence for these potentially sparser graphs.

An alternative cluster tree estimator

Original scheme constructs a hierarchy of neighborhood r-graphs:

- For each x_i: set $r_k(x_i)$ = distance to kth nearest neighbor
- As r increases from 0 to ∞:
  - Construct graph G_r:
    Nodes $\{x_i : r_k(x_i) \leq r\}$
    Edges between any (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha r$
  - Output the connected components of G_r

[Kpotufe-von Luxburg 2011] Instead of G_r, use graph G_r^{NN}:

- Same nodes, $\{x_i : r(x_i) \leq r\}$
- Edges (x_i, x_j) for which $\|x_i - x_j\| \leq \alpha \min(r_k(x_i), r_k(x_j))$

Similar rates of convergence for these potentially sparser graphs.

Open problem: other simple estimators?

Revisiting Hartigan-consistency

Recall Hartigan's notion of consistency:

> *Let A, A' be connected components of $\{f \geq \lambda\}$, for any λ. In the tree constructed from n data points X_n, let A_n be the smallest cluster containing $A \cap X_n$; likewise A'_n. Then:*
>
> $$\lim_{n \to \infty} \textit{Prob}[A_n \textit{ is disjoint from } A'_n] = 1$$

In other words, distinct clusters should (for large enough n) be disjoint in the estimated tree.

Revisiting Hartigan-consistency

Recall Hartigan's notion of consistency:

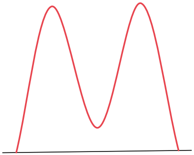
> *Let A, A' be connected components of $\{f \geq \lambda\}$, for any λ. In the tree constructed from n data points X_n, let A_n be the smallest cluster containing $A \cap X_n$; likewise A'_n. Then:*
>
> $$\lim_{n \to \infty} \textit{Prob}[A_n \textit{ is disjoint from } A'_n] = 1$$

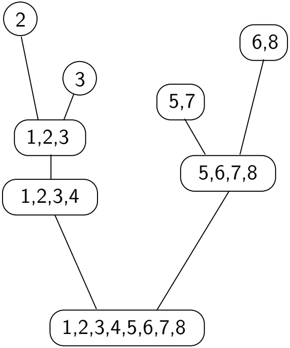
In other words, distinct clusters should (for large enough n) be disjoint in the estimated tree.

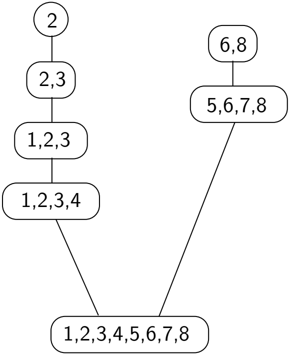
But this doesn't guard against excessive fragmentation within the estimated tree.

Excessive fragmentation: example

Density:







Pruning the cluster tree

- Build the cluster tree as before: at each scale r, there is a neighborhood graph G_r
- For each r: merge components of G_r that are connected in $G_{r+\delta(r)}$

Pruning the cluster tree

- Build the cluster tree as before: at each scale r, there is a neighborhood graph G_r
- For each r: merge components of G_r that are connected in $G_{r+\delta(r)}$

Kpotufe and von-Luxburg 2011: roughly the same consistency guarantees and rate of convergence hold, and in addition, under extra conditions, there is no spurious fragmentation.

Pruning the cluster tree

- Build the cluster tree as before: at each scale r, there is a neighborhood graph G_r
- For each r: merge components of G_r that are connected in $G_{r+\delta(r)}$

Kpotufe and von-Luxburg 2011: roughly the same consistency guarantees and rate of convergence hold, and in addition, under extra conditions, there is no spurious fragmentation.

Belkin-Eldridge-Wang 2015: A stronger notion of consistency that accounts for fragmentation.

More open problems

1. Other natural notions of cluster for a density f? Are there situations in which a hierarchy is not enough?
2. This notion of cluster is for densities. What about discrete distributions?
3. An $O(n \log n)$ algorithm?

Thanks

Many thanks to my co-authors Kamalika Chaudhuri, Samory Kpotufe, and Ulrike von Luxburg.