Clustering: Basics & Hierarchical Clustering

Philip D. Waggoner

MACS 40500: Computational Methods for American Politics

October 22, 2019

Lecture Outline

Lecture Outline

Clustering Basics

- Clustering attempts to group (or cluster) objects (i and $i\prime$) based on some rule defining the similarity (or dissimilarity) between the objects

Clustering Basics

- Clustering attempts to group (or cluster) objects (i and $i\prime$) based on some rule defining the similarity (or dissimilarity) between the objects
- Note that there is a distinction between clustering and classification/discrimination:

Clustering Basics

- Clustering attempts to group (or cluster) objects (i and $i\prime$) based on some rule defining the similarity (or dissimilarity) between the objects
- Note that there is a distinction between clustering and classification/discrimination:
    - **Clustering**: the group labels are not known a priori

Clustering Basics

- Clustering attempts to group (or cluster) objects (i and $i\prime$) based on some rule defining the similarity (or dissimilarity) between the objects
- Note that there is a distinction between clustering and classification/discrimination:
  - **Clustering**: the group labels are not known a priori
  - **Classification**: the group labels are known for a trained sample (next week)

Clustering Basics

- Clustering attempts to group (or cluster) objects (i and $i\prime$) based on some rule defining the similarity (or dissimilarity) between the objects
- Note that there is a distinction between clustering and classification/discrimination:
  - **Clustering**: the group labels are not known a priori
  - **Classification**: the group labels are known for a trained sample (next week)
- Thus, the typical goal in clustering is to discover the "natural groupings" present in the data

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise)

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise) $\rightsquigarrow$ move from signletons to progressively larger clusters based on similarity

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise) $\rightsquigarrow$ move from signletons to progressively larger clusters based on similarity
  - **Partitioning** (assignment, both "soft" and "hard")

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise) $\rightsquigarrow$ move from signletons to progressively larger clusters based on similarity
  - **Partitioning** (assignment, both "soft" and "hard") $\rightsquigarrow$ maintain a set of clusters and assign points nearest to the cluster centroid

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise) $\rightsquigarrow$ move from signletons to progressively larger clusters based on similarity
  - **Partitioning** (assignment, both "soft" and "hard") $\rightsquigarrow$ maintain a set of clusters and assign points nearest to the cluster centroid
- Key difference between these classes:

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise) $\rightsquigarrow$ move from signletons to progressively larger clusters based on similarity
  - **Partitioning** (assignment, both "soft" and "hard") $\rightsquigarrow$ maintain a set of clusters and assign points nearest to the cluster centroid
- Key difference between these classes: **subdividing the data**

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise) $\rightsquigarrow$ move from signletons to progressively larger clusters based on similarity
  - **Partitioning** (assignment, both "soft" and "hard") $\rightsquigarrow$ maintain a set of clusters and assign points nearest to the cluster centroid
- Key difference between these classes: **subdividing the data**
  - Hierarchical $\rightsquigarrow$ No

Clustering Basics

- Broadly, then, we can categorize clustering techniques into two camps:
  - **Hierarchical** (pairwise) $\rightsquigarrow$ move from signletons to progressively larger clusters based on similarity
  - **Partitioning** (assignment, both "soft" and "hard") $\rightsquigarrow$ maintain a set of clusters and assign points nearest to the cluster centroid
- Key difference between these classes: **subdividing the data**
  - Hierarchical $\rightsquigarrow$ No
  - Partitioning $\rightsquigarrow$ Yes

Clustering Basics

- Regardless of the approach, the **grouping** of data is central to reducing the dimensionality of the space, and so making the data more accessible and understandable

Clustering Basics

- Regardless of the approach, the **grouping** of data is central to reducing the dimensionality of the space, and so making the data more accessible and understandable
- But note that precisely how we think about grouping is dependent on the distribution of the data, which has implications for algorithm selection and validation

Clustering Basics

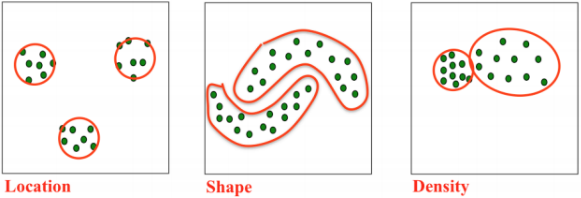
- Regardless of the approach, the **grouping** of data is central to reducing the dimensionality of the space, and so making the data more accessible and understandable
- But note that precisely how we think about grouping is dependent on the distribution of the data, which has implications for algorithm selection and validation
- In general, we can think of three main types of grouping:

Clustering Basics

- Regardless of the approach, the **grouping** of data is central to reducing the dimensionality of the space, and so making the data more accessible and understandable
- But note that precisely how we think about grouping is dependent on the distribution of the data, which has implications for algorithm selection and validation
- In general, we can think of three main types of grouping:
  - Location
  - Shape
  - Density

Clustering Basics

- Regardless of the approach, the **grouping** of data is central to reducing the dimensionality of the space, and so making the data more accessible and understandable
- But note that precisely how we think about grouping is dependent on the distribution of the data, which has implications for algorithm selection and validation
- In general, we can think of three main types of grouping:
  - Location
  - Shape
  - Density



Clustering Basics

- We will focus on three major types of clustering today and next class:

Clustering Basics

- We will focus on three major types of clustering today and next class:
  - Hierarchical agglomerative clustering (today)

Clustering Basics

- We will focus on three major types of clustering today and next class:
  - Hierarchical agglomerative clustering (today)
  - Hard partitioning (focus on k-means clustering *next class*)

Clustering Basics

- We will focus on three major types of clustering today and next class:
  - Hierarchical agglomerative clustering (today)
  - Hard partitioning (focus on k-means clustering *next class*)
  - Soft (probabilistic) partitioning (focus on the EM algorithm and Gaussian mixture models *next class*)

Clustering Basics

- We will focus on three major types of clustering today and next class:
  - Hierarchical agglomerative clustering (today)
  - Hard partitioning (focus on k-means clustering *next class*)
  - Soft (probabilistic) partitioning (focus on the EM algorithm and Gaussian mixture models *next class*)
- *Note*: Next class, also application on presidential vote shares by state in `R`

Lecture Outline

Diagnosing Clusterability

- Before we can cluster (or even decide whether clustering makes sense as a reasonable path forward), we should explore the feature space a bit

Diagnosing Clusterability

- Before we can cluster (or even decide whether clustering makes sense as a reasonable path forward), we should explore the feature space a bit

- There are several ways to do this:

Diagnosing Clusterability

- Before we can cluster (or even decide whether clustering makes sense as a reasonable path forward), we should explore the feature space a bit
- There are several ways to do this:
  - Informally (simple distribution plots)

Diagnosing Clusterability

- Before we can cluster (or even decide whether clustering makes sense as a reasonable path forward), we should explore the feature space a bit
- There are several ways to do this:
  - Informally (simple distribution plots)
  - Visually (VAT/ODI plots)

Diagnosing Clusterability

- Before we can cluster (or even decide whether clustering makes sense as a reasonable path forward), we should explore the feature space a bit
- There are several ways to do this:
  - Informally (simple distribution plots)
  - Visually (VAT/ODI plots)
  - Mathematically (sparse sampling)

A Note on "Informed Guessing"

A Note on “Informed Guessing”

- Importantly, note that with these approaches to assessing clusterability, as with virtually all of clustering, there is great ambiguity

A Note on "Informed Guessing"

- Importantly, note that with these approaches to assessing clusterability, as with virtually all of clustering, there is great ambiguity

- We can do our best to make sense of a complex feature space, but as our data are unlabeled, we are essentially always "guessing" about what these patterns are revealing

A Note on “Informed Guessing”

- Importantly, note that with these approaches to assessing clusterability, as with virtually all of clustering, there is great ambiguity
- We can do our best to make sense of a complex feature space, but as our data are unlabeled, we are essentially always “guessing” about what these patterns are revealing
- This is a limitation of UML acorss the board; yet simultaneously the reason it is so important to combine UML with other data reduction and modeling processes

Diagnosing Clusterability: Informally

- Let's start with informal diagnosis using the famous Fisher Iris data set:

Diagnosing Clusterability: Informally

- Let's start with informal diagnosis using the famous Fisher Iris data set:
  - Petal Length

Diagnosing Clusterability: Informally

- Let's start with informal diagnosis using the famous Fisher Iris data set:
  - Petal Length
  - Petal Width

Diagnosing Clusterability: Informally

- Let's start with informal diagnosis using the famous Fisher Iris data set:
  - Petal Length
  - Petal Width
  - Sepal Length

Diagnosing Clusterability: Informally

- Let's start with informal diagnosis using the famous Fisher Iris data set:
  - Petal Length
  - Petal Width
  - Sepal Length
  - Sepal Width

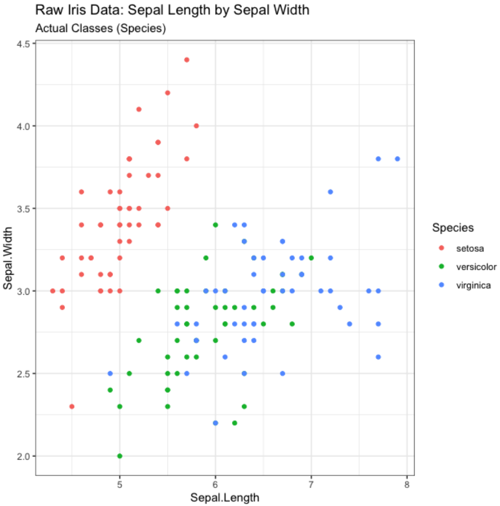
Diagnosing Clusterability: Informally

- Let's start with informal diagnosis using the famous Fisher Iris data set:
  - Petal Length
  - Petal Width
  - Sepal Length
  - Sepal Width
  - 3 Species: setosa, versicolor, and virginica

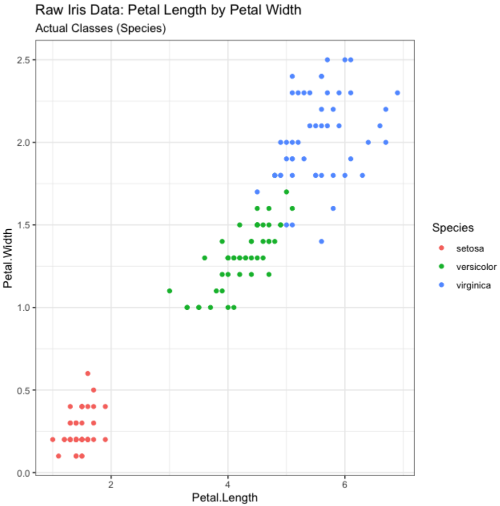
Diagnosing Clusterability: Informally

- Let's start with informal diagnosis using the famous Fisher Iris data set:
  - Petal Length
  - Petal Width
  - Sepal Length
  - Sepal Width
  - 3 Species: setosa, versicolor, and virginica
  - 150 observations (50 of each)

Diagnosing Clusterability: Informally (Sepal Length by Sepal Width)



Diagnosing Clusterability: Informally (Petal Length by Petal Width)



Diagnosing Clusterability: VAT (ODI) Plots

- Next, let's get a little more precise on diagnosing clusterability using Visual Assessment of Tendency (VAT) plots, and still with the Fisher Iris data set

Diagnosing Clusterability: VAT (ODI) Plots

- Next, let's get a little more precise on diagnosing clusterability using Visual Assessment of Tendency (VAT) plots, and still with the Fisher Iris data set
- Originally dervied by Bezdek and Hathaway (2002), these VAT plots are also often called Ordered Dissimilarity Images (ODI)

Diagnosing Clusterability: VAT (ODI) Plots

- Next, let's get a little more precise on diagnosing clusterability using Visual Assessment of Tendency (VAT) plots, and still with the Fisher Iris data set

- Originally dervied by Bezdek and Hathaway (2002), these VAT plots are also often called Ordered Dissimilarity Images (ODI)

- **Dissimilarity**: first, visualize the dissimilarity matrix

Diagnosing Clusterability: VAT (ODI) Plots

- Next, let's get a little more precise on diagnosing clusterability using Visual Assessment of Tendency (VAT) plots, and still with the Fisher Iris data set

- Originally dervied by Bezdek and Hathaway (2002), these VAT plots are also often called Ordered Dissimilarity Images (ODI)

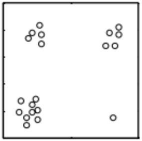
- **Dissimilarity**: first, visualize the dissimilarity matrix

- **Ordered**: objects, o, that are spatially proximate (measured as k) are displayed in consecutive order (if k_i is near k_j, then $o_1, o_2 \forall o \equiv o_{ki}, o_{kj}$)

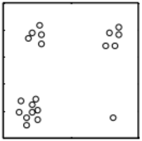
Diagnosing Clusterability: VAT (ODI) Plots

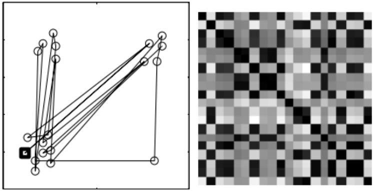
- Next, let's get a little more precise on diagnosing clusterability using Visual Assessment of Tendency (VAT) plots, and still with the Fisher Iris data set
- Originally dervied by Bezdek and Hathaway (2002), these VAT plots are also often called Ordered Dissimilarity Images (ODI)
- **Dissimilarity**: first, visualize the dissimilarity matrix
- **Ordered**: objects, o, that are spatially proximate (measured as k) are displayed in consecutive order (if k_i is near k_j, then $o_1, o_2 \forall o \equiv o_{ki}, o_{kj}$)
- The visual result becomes darker blocks along the diagonal reflect greater spatial similarity, compared to lighter shaded blocks, which inversely suggest greater dissimilarity

Order is Important

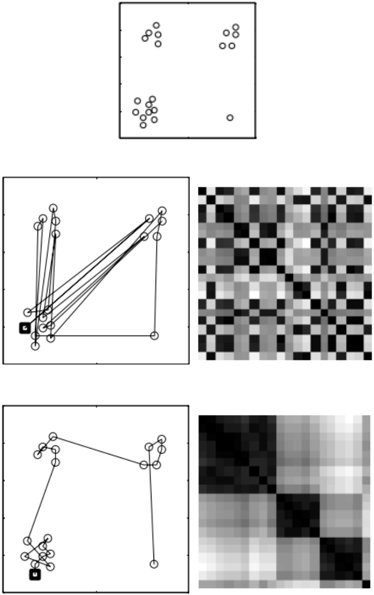


Order is Important





Order is Important



VAT (ODI): Iris Data

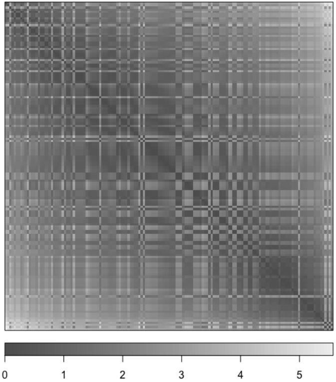


Figure: ODI: Sepal

VAT (ODI): Iris Data

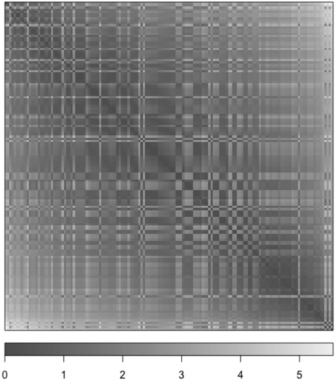


Figure: ODI: Sepal

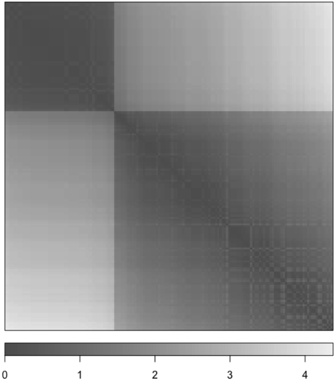


Figure: ODI: Petal

A Quick Comparison

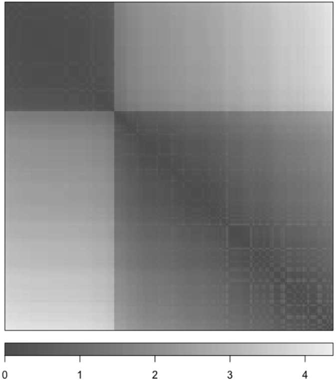


Figure: ODI: Petal

A Quick Comparison

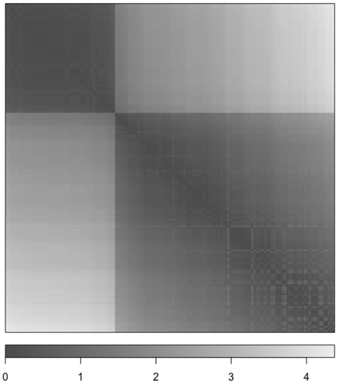


Figure: ODI: Petal

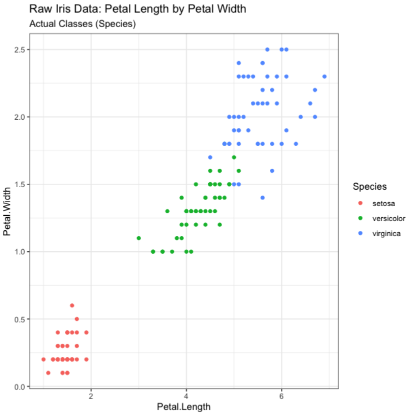


Figure: Raw Data: Petal

Diagnosing Clusterability: Hopkins Statistic

- We can also mathematically derive what the VAT plots are revealing using a simple, but powerful statistic called the Hopkins statistic

Diagnosing Clusterability: Hopkins Statistic

- We can also mathematically derive what the VAT plots are revealing using a simple, but powerful statistic called the Hopkins statistic
- The Hopkins (or "H") statistic tests the the null hypothesis of spatial randomness in the data using a sparse sampling test

Diagnosing Clusterability: Hopkins Statistic

- We can also mathematically derive what the VAT plots are revealing using a simple, but powerful statistic called the Hopkins statistic
- The Hopkins (or "H") statistic tests the the null hypothesis of spatial randomness in the data using a sparse sampling test
- It calculates the probability that a given dataset is generated by a uniform (random noise, with no clusters) distribution or not (non-random, with clustering likely)
- This general procedure is called **sparse sampling**

A Null Hypothesis Framework

- We specify a null hypothesis test,

A Null Hypothesis Framework

- We specify a null hypothesis test,

  H_0: the data is uniformly ("equally") distributed

A Null Hypothesis Framework

- We specify a null hypothesis test,

  H_0: the data is uniformly ("equally") distributed

  H_A: the data is not uniformly distributed

A Null Hypothesis Framework

- We specify a null hypothesis test,

  H_0: the data is uniformly ("equally") distributed

  H_A: the data is not uniformly distributed

- **Goal**: calculate the pairwise dissimilarity across all observations in the **actual** data is compared to a set of **simulated** dataset drawn from some random distribution (usually uniform) with the *same* standard deviation as the original data

A Null Hypothesis Framework

- We specify a null hypothesis test,

  H_0: the data is uniformly ("equally") distributed

  H_A: the data is not uniformly distributed

- **Goal**: calculate the pairwise dissimilarity across all observations in the **actual** data is compared to a set of **simulated** dataset drawn from some random distribution (usually uniform) with the *same* standard deviation as the original data

- In the form of a question: is the actual data random, compared to the synthetic data set, which we *know* is random?

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_i\prime$

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_{i'}$
- Calculate the distance between p_i and $p_{i'}$ and denote it as $u_j = dist(p_i, p_{i'})$

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_i\prime$
- Calculate the distance between p_i and $p_i\prime$ and denote it as $u_j = dist(p_i, p_i\prime)$
- Create a **simulated** dataset, $D\prime$, drawn from a random, uniform distribution with n observations (q), with the same standard deviation as D

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_i\prime$
- Calculate the distance between p_i and $p_i\prime$ and denote it as $u_j = dist(p_i, p_i\prime)$
- Create a **simulated** dataset, $D\prime$, drawn from a random, uniform distribution with n observations (q), with the same standard deviation as D
- For each observation $q_i \in D\prime$, find it's nearest neighbor $q_i\prime$

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_i\prime$
- Calculate the distance between p_i and $p_i\prime$ and denote it as $u_j = dist(p_i, p_i\prime)$
- Create a **simulated** dataset, $D\prime$, drawn from a random, uniform distribution with n observations (q), with the same standard deviation as D
- For each observation $q_i \in D\prime$, find it's nearest neighbor $q_i\prime$
- Calculate the distance between q_i and $q_i\prime$ and denote it $w_j = dist(q_i, q_i\prime)$

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_i\prime$
- Calculate the distance between p_i and $p_i\prime$ and denote it as $u_j = dist(p_i, p_i\prime)$
- Create a **simulated** dataset, $D\prime$, drawn from a random, uniform distribution with n observations (q), with the same standard deviation as D
- For each observation $q_i \in D\prime$, find it's nearest neighbor $q_i\prime$
- Calculate the distance between q_i and $q_i\prime$ and denote it $w_j = dist(q_i, q_i\prime)$
- H is calculated as the sum of the mean distance in the actual data divided by the sum of the mean distances in the actual and simulated data

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_i\prime$
- Calculate the distance between p_i and $p_i\prime$ and denote it as $u_j = dist(p_i, p_i\prime)$
- Create a **simulated** dataset, $D\prime$, drawn from a random, uniform distribution with n observations (q), with the same standard deviation as D
- For each observation $q_i \in D\prime$, find it's nearest neighbor $q_i\prime$
- Calculate the distance between q_i and $q_i\prime$ and denote it $w_j = dist(q_i, q_i\prime)$
- H is calculated as the sum of the mean distance in the actual data divided by the sum of the mean distances in the actual and simulated data

$$H = \frac{\sum_{j=1}^{m} u_j}{\sum_{j=1}^{m} u_j + \sum_{j=1}^{m} w_j} \tag{1}$$

Calculating the Hopkins Statistic

- Sample uniformly n observations, p, from our **actual** data, D
- For each observation $p_i \in D$, find the nearest neighbor $p_i\prime$
- Calculate the distance between p_i and $p_i\prime$ and denote it as $u_j = dist(p_i, p_i\prime)$
- Create a **simulated** dataset, $D\prime$, drawn from a random, uniform distribution with n observations (q), with the same standard deviation as D
- For each observation $q_i \in D\prime$, find it's nearest neighbor $q_i\prime$
- Calculate the distance between q_i and $q_i\prime$ and denote it $w_j = dist(q_i, q_i\prime)$
- H is calculated as the sum of the mean distance in the actual data divided by the sum of the mean distances in the actual and simulated data

$$H = \frac{\sum_{j=1}^{m} u_j}{\sum_{j=1}^{m} u_j + \sum_{j=1}^{m} w_j} \tag{1}$$

- In general, $H > 0.5$ leads to rejection of H_0, suggesting the data are non-random, and are "clusterable"

Lecture Outline

Similarity and Dissimilarity

- Distance data are relational, where greater values reflect can either reflect greater similarity or dissimilarity across observations or features

Similarity and Dissimilarity

- Distance data are relational, where greater values reflect can either reflect greater similarity or dissimilarity across observations or features
- Important considerations include the nature of the variables, scales of measurement, and domain expertise

Similarity and Dissimilarity

- Distance data are relational, where greater values reflect can either reflect greater similarity or dissimilarity across observations or features
- Important considerations include the nature of the variables, scales of measurement, and domain expertise
- When *items* are clustered, proximity is usually indicated by some sort of distance

Similarity and Dissimilarity

- Distance data are relational, where greater values reflect can either reflect greater similarity or dissimilarity across observations or features
- Important considerations include the nature of the variables, scales of measurement, and domain expertise
- When *items* are clustered, proximity is usually indicated by some sort of distance
- By contrast, *features* are usually grouped on the basis of correlation coefficients (but so can observations)

Standardization

- Before calculating distance metrics, it is important to first, and *always*, standardize data in clustering applications

Standardization

- Before calculating distance metrics, it is important to first, and *always*, standardize data in clustering applications
- Notably, distance calculations are strongly influenced by unit measurement and magnitude

Standardization

- Before calculating distance metrics, it is important to first, and *always*, standardize data in clustering applications

- Notably, distance calculations are strongly influenced by unit measurement and magnitude

- Suppose you had two features on which you wanted to cluster units: weight (lbs.) and household income (dollars)

Standardization

- Before calculating distance metrics, it is important to first, and *always*, standardize data in clustering applications
- Notably, distance calculations are strongly influenced by unit measurement and magnitude
- Suppose you had two features on which you wanted to cluster units: weight (lbs.) and household income (dollars)
- In such a case, different units of measurement *and* distributions (skewed) will always return biased results

Standardization

- Before calculating distance metrics, it is important to first, and *always*, standardize data in clustering applications
- Notably, distance calculations are strongly influenced by unit measurement and magnitude
- Suppose you had two features on which you wanted to cluster units: weight (lbs.) and household income (dollars)
- In such a case, different units of measurement *and* distributions (skewed) will always return biased results
- Thus first, always standardize input features to ensure they are "unitless" (commonly setting $\mu = 0$ and $\sigma = 1$)

Similarity and Dissimilarity

- We begin with some configuration of actual data, $\delta(p, q)$, and we want to calculate a measurement of distance, $d(p, q)$, that accurately reflects this, such that $d(p, q) \rightarrow \delta(p, q)$

Similarity and Dissimilarity

- We begin with some configuration of actual data, $\delta(p, q)$, and we want to calculate a measurement of distance, $d(p, q)$, that accurately reflects this, such that $d(p, q) \rightarrow \delta(p, q)$
- Generally a distance measure $d(p, q)$ between two points p and q satisfies the following properties, where g is any other intermediate point:

Similarity and Dissimilarity

- We begin with some configuration of actual data, $\delta(p,q)$, and we want to calculate a measurement of distance, $d(p,q)$, that accurately reflects this, such that $d(p,q) \rightarrow \delta(p,q)$

- Generally a distance measure $d(p,q)$ between two points p and q satisfies the following properties, where g is any other intermediate point:

    - $d(p,q) = d(q,p)$

Similarity and Dissimilarity

- We begin with some configuration of actual data, $\delta(p, q)$, and we want to calculate a measurement of distance, $d(p, q)$, that accurately reflects this, such that $d(p, q) \rightarrow \delta(p, q)$

- Generally a distance measure $d(p, q)$ between two points p and q satisfies the following properties, where g is any other intermediate point:

  - $d(p, q) = d(q, p)$

  - $d(p, q) > 0$, if $p \neq q$

Similarity and Dissimilarity

- We begin with some configuration of actual data, $\delta(p, q)$, and we want to calculate a measurement of distance, $d(p, q)$, that accurately reflects this, such that $d(p, q) \rightarrow \delta(p, q)$

- Generally a distance measure $d(p, q)$ between two points p and q satisfies the following properties, where g is any other intermediate point:

  - $d(p, q) = d(q, p)$

  - $d(p, q) > 0$, if $p \neq q$

  - $d(p, q) = 0$, if $p = q$

Similarity and Dissimilarity

- We begin with some configuration of actual data, $\delta(p,q)$, and we want to calculate a measurement of distance, $d(p,q)$, that accurately reflects this, such that $d(p,q) \rightarrow \delta(p,q)$

- Generally a distance measure $d(p,q)$ between two points p and q satisfies the following properties, where g is any other intermediate point:

  - $d(p,q) = d(q,p)$
  - $d(p,q) > 0$, if $p \neq q$
  - $d(p,q) = 0$, if $p = q$
  - $d(p,q) \leq d(p,g) + d(g,q)$

Spatial Measures Starting with Minkowski

Spatial Measures Starting with Minkowski

$$d_m(p, q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{2}$$

where, setting $m \geq 1$ defines some true distance

Spatial Measures Starting with Minkowski

$$d_m(p, q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{2}$$

where, setting $m \geq 1$ defines some true distance

- $m = 1$: Manhattan ("city block") Distance:

Spatial Measures Starting with Minkowski

$$d_m(p, q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{2}$$

where, setting $m \geq 1$ defines some true distance

- $m = 1$: Manhattan ("city block") Distance:

$$d_{manhattan}(p, q) = \sum_{i=1}^{n} |p_i - q_i|$$

Spatial Measures Starting with Minkowski

$$d_m(p, q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{2}$$

where, setting $m \geq 1$ defines some true distance

- $m = 1$: Manhattan ("city block") Distance:

$$d_{manhattan}(p, q) = \sum_{i=1}^{n} |p_i - q_i|$$

- Canberra Distance (weighted version of Manhattan):

Spatial Measures Starting with Minkowski

$$d_m(p, q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{2}$$

where, setting $m \geq 1$ defines some true distance

- $m = 1$: Manhattan ("city block") Distance:

$$d_{manhattan}(p, q) = \sum_{i=1}^{n} |p_i - q_i|$$

- Canberra Distance (weighted version of Manhattan):

$$d_{canberra}(p, q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|}$$

Spatial Measures

$$d_m(p, q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{3}$$

where, setting $m \geq 1$ defines some true distance

- $m = 2$: Euclidean Distance:

Spatial Measures

$$d_m(p,q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{3}$$

where, setting $m \geq 1$ defines some true distance

- $m = 2$: Euclidean Distance:

$$d_{euclidean}(p,q) = (\sum_{i=1}^{n} (p_i - q_i)^2)^{\frac{1}{2}}$$

or, more compactly (and commonly),

Spatial Measures

$$d_m(p,q) = (\sum_{i=1}^{n} |p_i - q_i|^m)^{\frac{1}{m}} \tag{3}$$

where, setting $m \geq 1$ defines some true distance

- $m = 2$: Euclidean Distance:

$$d_{euclidean}(p,q) = (\sum_{i=1}^{n} (p_i - q_i)^2)^{\frac{1}{2}}$$

or, more compactly (and commonly),

$$d_{euclidean}(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Correlation Measures

- Correlation measures calculate similarity between observations based on...

Correlation Measures

- Correlation measures calculate similarity between observations based on...**correlation**

Correlation Measures

- Correlation measures calculate similarity between observations based on...**correlation**

- Meaning, two observations could be correlated across features, but far apart in Euclidean space, suggesting that we're interested in *attribute* similarity

Correlation Measures

- Correlation measures calculate similarity between observations based on...**correlation**
- Meaning, two observations could be correlated across features, but far apart in Euclidean space, suggesting that we're interested in *attribute* similarity
- Pearson Distance:

Correlation Measures

- Correlation measures calculate similarity between observations based on...**correlation**
- Meaning, two observations could be correlated across features, but far apart in Euclidean space, suggesting that we're interested in *attribute* similarity
- Pearson Distance:

$$d_{pearson}(p, q) = 1 - \frac{\sum_{i=1}^{n}(p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2 \sum_{i=1}^{n}(q_i - \bar{q})^2}}$$

Correlation Measures

- Correlation measures calculate similarity between observations based on...**correlation**
- Meaning, two observations could be correlated across features, but far apart in Euclidean space, suggesting that we're interested in *attribute* similarity
- Pearson Distance:

$$d_{pearson}(p, q) = 1 - \frac{\sum_{i=1}^{n}(p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2 \sum_{i=1}^{n}(q_i - \bar{q})^2}}$$

- Eisen Cosine Distance:

Correlation Measures

- Correlation measures calculate similarity between observations based on...**correlation**

- Meaning, two observations could be correlated across features, but far apart in Euclidean space, suggesting that we're interested in *attribute* similarity

- Pearson Distance:

$$d_{pearson}(p, q) = 1 - \frac{\sum_{i=1}^{n}(p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2 \sum_{i=1}^{n}(q_i - \bar{q})^2}}$$

- Eisen Cosine Distance:

$$d_{eisen}(p, q) = 1 - \frac{|\sum_{i=1}^{n} p_i q_i|}{\sqrt{\sum_{i=1}^{n} p_i^2 \sum_{i=1}^{n} q_i^2}}$$

What can I cluster?

- At this point you may be thinking, what can I cluster thats not a flower?

What can I cluster?

- At this point you may be thinking, what can I cluster thats not a flower? $\rightsquigarrow$ a lot of things...

What can I cluster?

- At this point you may be thinking, what can I cluster thats not a flower? $\rightsquigarrow$ a lot of things...
  - Respondents on large surveys
  - Geopolitical studies
  - Market preferences
  - Economies
  - Social media users
  - Perceptual studies
  - Geographic trends
  - And, on Friday, partisan voting by state
  - And, on your HW, state legislative professionalism

Lecture Outline

Agglomerative Hierarchical Clustering

- The most common form of hierarchical clustering is agglomerative, or "bottom-up" clustering

Agglomerative Hierarchical Clustering

- The most common form of hierarchical clustering is agglomerative, or "bottom-up" clustering
- Opposite partitioning methods, hierarchical clustering is concerned with creating clusters one observation at a time (or recursively for divisive)

Agglomerative Hierarchical Clustering

- The most common form of hierarchical clustering is agglomerative, or "bottom-up" clustering
- Opposite partitioning methods, hierarchical clustering is concerned with creating clusters one observation at a time (or recursively for divisive)
- Each observation is treated as a cluster, and is fused with the closest observation in space

Agglomerative Hierarchical Clustering

- The most common form of hierarchical clustering is agglomerative, or "bottom-up" clustering
- Opposite partitioning methods, hierarchical clustering is concerned with creating clusters one observation at a time (or recursively for divisive)
- Each observation is treated as a cluster, and is fused with the closest observation in space
- This cluster is joined with other similar (close) observations in a pairwise fashion until all observations belong to a cluster, converging when all observations belong to a single cluster, k

Agglomerative Hierarchical Clustering

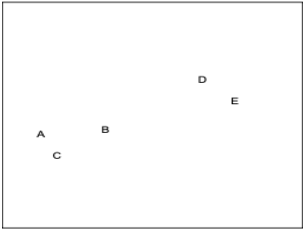
- The most common form of hierarchical clustering is agglomerative, or "bottom-up" clustering
- Opposite partitioning methods, hierarchical clustering is concerned with creating clusters one observation at a time (or recursively for divisive)
- Each observation is treated as a cluster, and is fused with the closest observation in space
- This cluster is joined with other similar (close) observations in a pairwise fashion until all observations belong to a cluster, converging when all observations belong to a single cluster, k
- Key terms: distance measure, linkage methods, dendrogram, tree-cutting

Hierarchical Clustering: The Idea

- Let's consider simple case

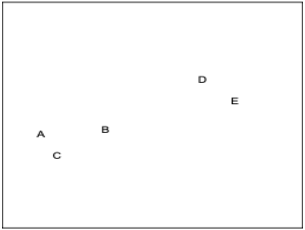
Hierarchical Clustering: The Idea

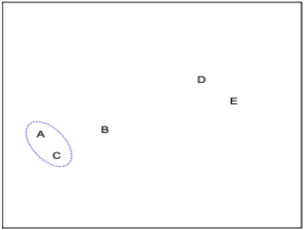
- Let's consider simple case



Hierarchical Clustering: The Idea

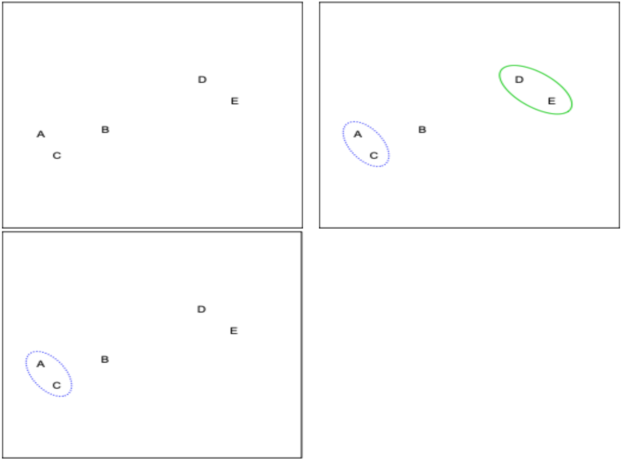
- Let's consider simple case





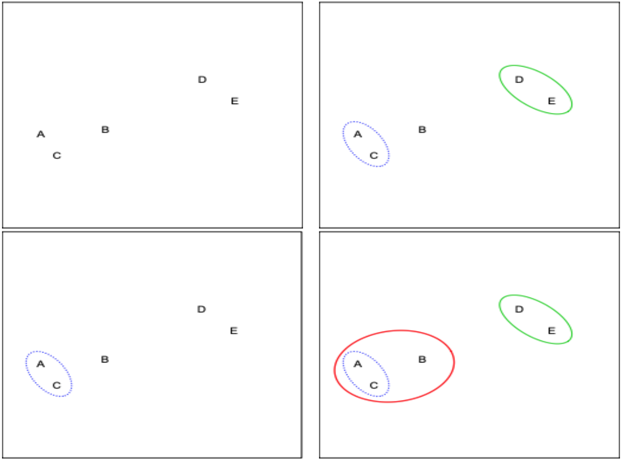
Hierarchical Clustering: The Idea

- Let's consider simple case



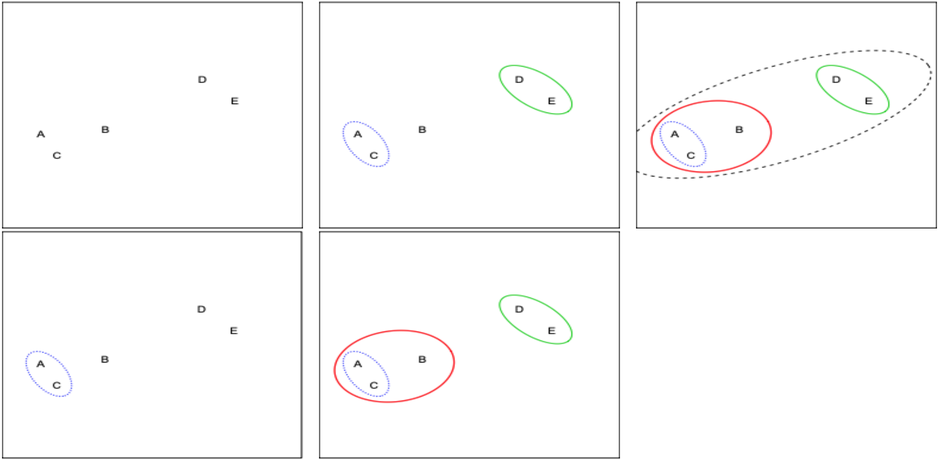
Hierarchical Clustering: The Idea

- Let's consider simple case



Hierarchical Clustering: The Idea

- Let's consider simple case



A Simple Agglomerative Hierarchical Clustering Algorithm

- Treat each observation as a singleton

A Simple Agglomerative Hierarchical Clustering Algorithm

- Treat each observation as a singleton
- Begin with n observations and some [distance] measure of all the $\binom{n}{2} = \frac{n(n-1)}{2}$ pairwise dissimilarities

A Simple Agglomerative Hierarchical Clustering Algorithm

- Treat each observation as a singleton
- Begin with n observations and some [distance] measure of all the $\binom{n}{2} = \frac{n(n-1)}{2}$ pairwise dissimilarities
- Examine all pairwise inter-cluster dissimilarities among the clusters, and identify the pair of clusters that are *least* dissimilar

A Simple Agglomerative Hierarchical Clustering Algorithm

- Treat each observation as a singleton
- Begin with n observations and some [distance] measure of all the $\binom{n}{2} = \frac{n(n-1)}{2}$ pairwise dissimilarities
- Examine all pairwise inter-cluster dissimilarities among the clusters, and identify the pair of clusters that are *least* dissimilar
- Fuse these clusters together based on the linkage method

A Simple Agglomerative Hierarchical Clustering Algorithm

- Treat each observation as a singleton
- Begin with n observations and some [distance] measure of all the $\binom{n}{2} = \frac{n(n-1)}{2}$ pairwise dissimilarities
- Examine all pairwise inter-cluster dissimilarities among the clusters, and identify the pair of clusters that are *least* dissimilar
- Fuse these clusters together based on the linkage method
- Compute the new pairwise inter-cluster dissimilarities among the remaining clusters

A Simple Agglomerative Hierarchical Clustering Algorithm

- Treat each observation as a singleton
- Begin with n observations and some [distance] measure of all the $\binom{n}{2} = \frac{n(n-1)}{2}$ pairwise dissimilarities
- Examine all pairwise inter-cluster dissimilarities among the clusters, and identify the pair of clusters that are *least* dissimilar
- Fuse these clusters together based on the linkage method
- Compute the new pairwise inter-cluster dissimilarities among the remaining clusters
- Stop when we reach k clusters ($k = 1$ in agglomerative; $k = n$ in divisive)

Lecture Outline

Linkage Methods

- Linkage defines the (dis)similarity between two groups of observations

Linkage Methods

- Linkage defines the (dis)similarity between two groups of observations
- There are five common types of linkage: complete, single, Ward's method, average, and centroid

Linkage Methods

- **Complete** linkage uses the *maximal* inter-cluster dissimilarity,

$$d_{complete}(C_x, C_y) = max\{C_x, C_y\}$$

Linkage Methods

- **Complete** linkage uses the *maximal* inter-cluster dissimilarity,

$$d_{complete}(C_x, C_y) = max\{C_x, C_y\}$$

- **Single** linkage uses the *minimal* inter-cluster dissimilarity,

$$d_{single}(C_x, C_y) = min\{C_x, C_y\}$$

Linkage Methods

- **Complete** linkage uses the *maximal* inter-cluster dissimilarity,

$$d_{complete}(C_x, C_y) = max\{C_x, C_y\}$$

- **Single** linkage uses the *minimal* inter-cluster dissimilarity,

$$d_{single}(C_x, C_y) = min\{C_x, C_y\}$$

- **Ward's** linkage method joins the two clusters whose fusion is constrained by the smallest increase in SSE calculated per cluster, C,

$$\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Linkage Methods

- **Average** linkage uses the *mean* inter-cluster dissimilarity,

$$d_{average}(C_x, C_y) = \frac{\sum_i \sum_j d_{ij}}{N_{C_x} N_{C_y}}$$

where, d_{ij} is the pairwise distance between observations i and j, and N_* is the total number of observations in the computed cluster, C

Linkage Methods

- **Average** linkage uses the *mean* inter-cluster dissimilarity,

$$d_{average}(C_x, C_y) = \frac{\sum_i \sum_j d_{ij}}{N_{C_x} N_{C_y}}$$

where, d_{ij} is the pairwise distance between observations i and j, and N_* is the total number of observations in the computed cluster, C

- **Centroid** linkage computes the dissimilarity between the *centroid* for cluster, $\bar{x}_{C_*}$,

$$d_{centroid}(C_x, C_y) = d(\bar{x}_{C_x}, \bar{x}_{C_y})$$

Linkage Methods

- **Average** linkage uses the *mean* inter-cluster dissimilarity,

$$d_{average}(C_x, C_y) = \frac{\sum_i \sum_j d_{ij}}{N_{C_x} N_{C_y}}$$

where, d_{ij} is the pairwise distance between observations i and j, and N_* is the total number of observations in the computed cluster, C

- **Centroid** linkage computes the dissimilarity between the *centroid* for cluster, $\bar{x}_{C_*}$,

$$d_{centroid}(C_x, C_y) = d(\bar{x}_{C_x}, \bar{x}_{C_y})$$

- Difference between average and centroid:

Linkage Methods

- **Average** linkage uses the *mean* inter-cluster dissimilarity,

$$d_{average}(C_x, C_y) = \frac{\sum_i \sum_j d_{ij}}{N_{C_x} N_{C_y}}$$

where, d_{ij} is the pairwise distance between observations i and j, and N_* is the total number of observations in the computed cluster, C

- **Centroid** linkage computes the dissimilarity between the *centroid* for cluster, $\bar{x}_{C_*}$,

$$d_{centroid}(C_x, C_y) = d(\bar{x}_{C_x}, \bar{x}_{C_y})$$

- Difference between average and centroid: average $\rightsquigarrow$ average of all *pairwise* calculations;

Linkage Methods

- **Average** linkage uses the *mean* inter-cluster dissimilarity,

$$d_{average}(C_x, C_y) = \frac{\sum_i \sum_j d_{ij}}{N_{C_x} N_{C_y}}$$

where, d_{ij} is the pairwise distance between observations i and j, and N_* is the total number of observations in the computed cluster, C

- **Centroid** linkage computes the dissimilarity between the *centroid* for cluster, $\bar{x}_{C_*}$,

$$d_{centroid}(C_x, C_y) = d(\bar{x}_{C_x}, \bar{x}_{C_y})$$

- Difference between average and centroid: average $\rightsquigarrow$ average of all *pairwise* calculations; centroid $\rightsquigarrow$ fuses across centroids, which are intra-cluster averages

Which linkage method should we select?

Which linkage method should we select?

- There is no rule on determining which method is "best", as best is subjective, and typically project (or domain) dependent

Which linkage method should we select?

- There is no rule on determining which method is "best", as best is subjective, and typically project (or domain) dependent
  - **Average**, **complete** and **single** linkage are most popular among statisticians, while **centroid** is often used in genomics and **Ward's** is most common in text mining

Which linkage method should we select?

- There is no rule on determining which method is “best”, as best is subjective, and typically project (or domain) dependent
  - **Average**, **complete** and **single** linkage are most popular among statisticians, while **centroid** is often used in genomics and **Ward’s** is most common in text mining
  - **Average** and **complete** linkage tend to be preferred because they tend to yield more balanced dendrograms (more in a moment)

Which linkage method should we select?

- There is no rule on determining which method is "best", as best is subjective, and typically project (or domain) dependent
  - **Average**, **complete** and **single** linkage are most popular among statisticians, while **centroid** is often used in genomics and **Ward's** is most common in text mining
  - **Average** and **complete** linkage tend to be preferred because they tend to yield more balanced dendrograms (more in a moment)
  - **Single** linkage can result in elongated, stringy-type clusters where each observation fuses one-at-a-time (i.e., its not pretty...)

Which linkage method should we select?

- There is no rule on determining which method is "best", as best is subjective, and typically project (or domain) dependent
  - **Average**, **complete** and **single** linkage are most popular among statisticians, while **centroid** is often used in genomics and **Ward's** is most common in text mining
  - **Average** and **complete** linkage tend to be preferred because they tend to yield more balanced dendrograms (more in a moment)
  - **Single** linkage can result in elongated, stringy-type clusters where each observation fuses one-at-a-time (i.e., its not pretty...)
  - **Ward's** method is based on minimizing the "loss of information" from joining two groups

Reiterating Differences between Distance Measures

- Two types of distance in hierarchical clustering might be confusing: "distance measure" and "linkage method"

- Worth reiterating the difference here to avoid confusion

Reiterating Differences between Distance Measures

- Two types of distance in hierarchical clustering might be confusing: "distance measure" and "linkage method"
- Worth reiterating the difference here to avoid confusion
- **Distance**: the measure of (dis)similarity between *observations* (e.g., $d_{euclidean}(p, q)$)

Reiterating Differences between Distance Measures

- Two types of distance in hierarchical clustering might be confusing: "distance measure" and "linkage method"
- Worth reiterating the difference here to avoid confusion
- **Distance**: the measure of (dis)similarity between *observations* (e.g., $d_{euclidean}(p, q)$)
- **Linkage**: also distance, but the measure of (dis)similarity between *clusters* (e.g., $d_{single}(C_x, C_y)$)

Reiterating Differences between Distance Measures

- Two types of distance in hierarchical clustering might be confusing: "distance measure" and "linkage method"
- Worth reiterating the difference here to avoid confusion
- **Distance**: the measure of (dis)similarity between *observations* (e.g., $d_{euclidean}(p, q)$)
- **Linkage**: also distance, but the measure of (dis)similarity between *clusters* (e.g., $d_{single}(C_x, C_y)$)
- Therefore, the input for a hierarchical clustering algorithm is an $N \times N$ distance matrix, from which **inter-cluster distances** are calculated via the selected linkage method

Lecture Outline

Evaluating Hierarchical Clustering Output: Dendrograms

- So we fit a hierarchical clustering algorithm... now what?

Evaluating Hierarchical Clustering Output: Dendrograms

- So we fit a hierarchical clustering algorithm... now what?
- The most common, clear way to diagnose and explore output is a dendrogram

Evaluating Hierarchical Clustering Output: Dendrograms

- So we fit a hierarchical clustering algorithm... now what?
- The most common, clear way to diagnose and explore output is a dendrogram
- A tree-like structure with "leaves" and "branches"

Evaluating Hierarchical Clustering Output: Dendrograms

- So we fit a hierarchical clustering algorithm... now what?
- The most common, clear way to diagnose and explore output is a dendrogram
- A tree-like structure with "leaves" and "branches"
  - **Leaves**: observations

Evaluating Hierarchical Clustering Output: Dendrograms

- So we fit a hierarchical clustering algorithm... now what?
- The most common, clear way to diagnose and explore output is a dendrogram
- A tree-like structure with "leaves" and "branches"
  - **Leaves**: observations
  - **Branches**: inter-cluster connectors

Evaluating Hierarchical Clustering Output: Dendrograms

- So we fit a hierarchical clustering algorithm... now what?
- The most common, clear way to diagnose and explore output is a dendrogram
- A tree-like structure with "leaves" and "branches"
  - **Leaves**: observations
  - **Branches**: inter-cluster connectors
- As each **leaf** is fused with another, progressing from the bottom-up until all singletons are merged into a single **tree**

Evaluating Hierarchical Clustering Output: Dendrograms

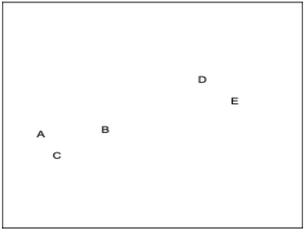
- So we fit a hierarchical clustering algorithm... now what?
- The most common, clear way to diagnose and explore output is a dendrogram
- A tree-like structure with "leaves" and "branches"
  - **Leaves**: observations
  - **Branches**: inter-cluster connectors
- As each **leaf** is fused with another, progressing from the bottom-up until all singletons are merged into a single **tree**
- There is only one value-axis: the Y axis is the measured distance

Evaluating Hierarchical Clustering Output: Dendrograms

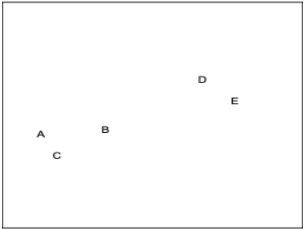
- So we fit a hierarchical clustering algorithm... now what?
- The most common, clear way to diagnose and explore output is a dendrogram
- A tree-like structure with "leaves" and "branches"
  - **Leaves**: observations
  - **Branches**: inter-cluster connectors
- As each **leaf** is fused with another, progressing from the bottom-up until all singletons are merged into a single **tree**
- There is only one value-axis: the Y axis is the measured distance
- So we can get clear clustering when branches along the Y axis are long (suggesting greater distance from other clusters), and less obvious clustering when the branches are shorter

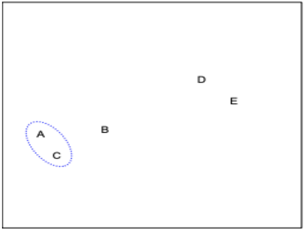
Returning to a Simple Case

Returning to a Simple Case

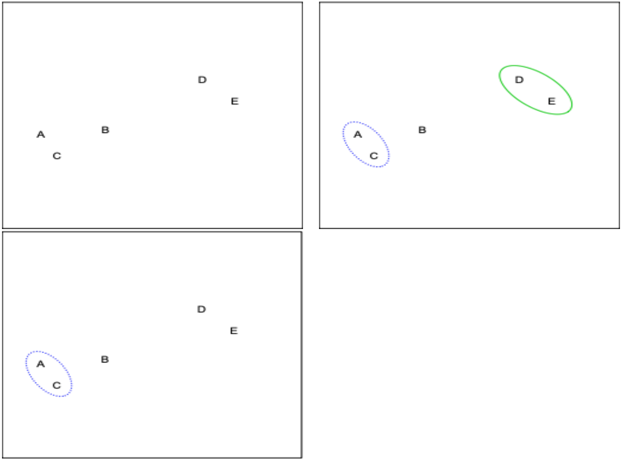


Returning to a Simple Case

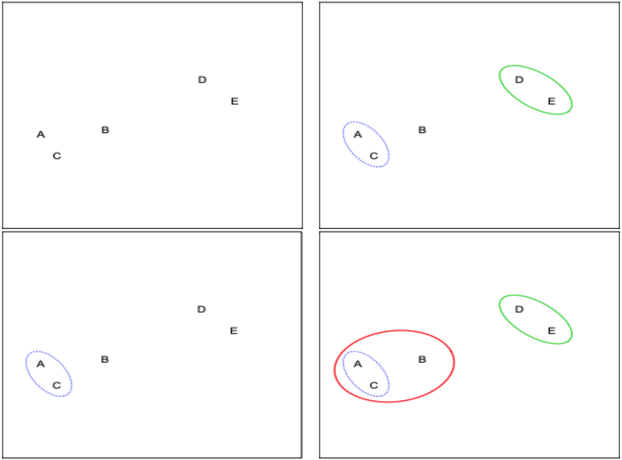




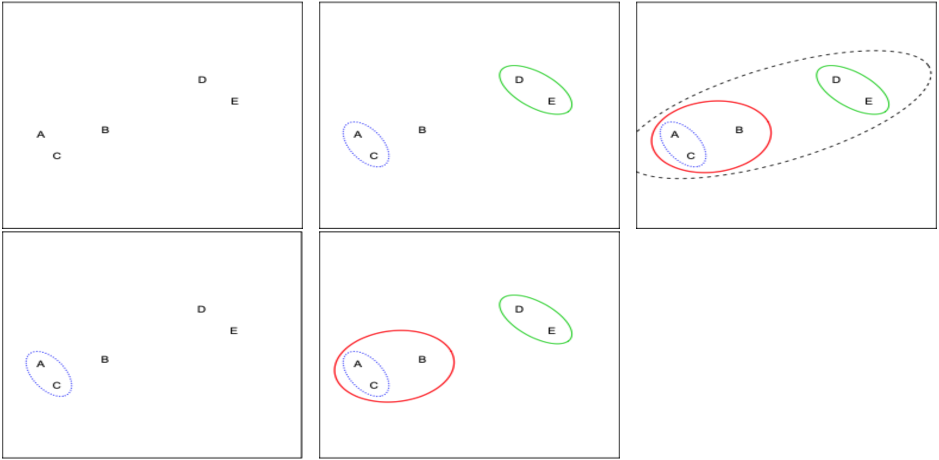
Returning to a Simple Case



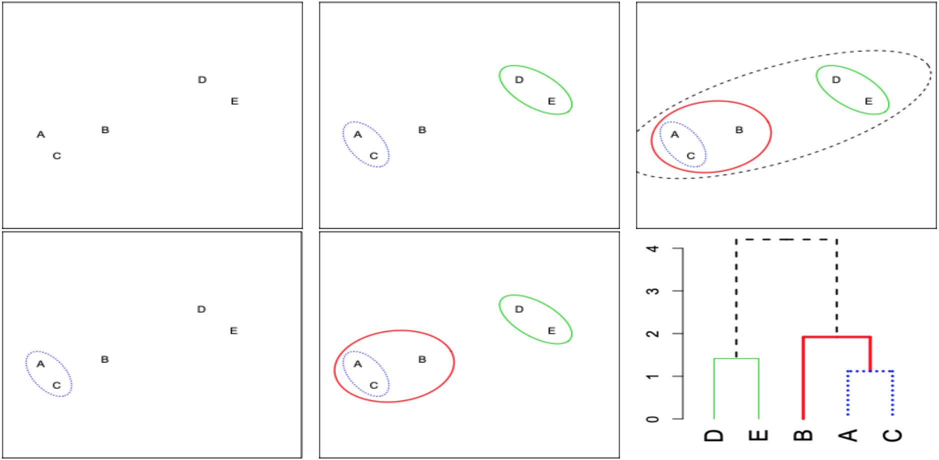
Returning to a Simple Case



Returning to a Simple Case



Returning to a Simple Case



Final Considerations

- Though hierarchical clustering is lauded for being assumption free in that we don't select k *a priori*, we actually do need to select k, but *post hoc*

Final Considerations

- Though hierarchical clustering is lauded for being assumption free in that we don't select k *a priori*, we actually do need to select k, but *post hoc*
- In other words, we fit our algorithm, we plotted our dendrogram, and we now have to do something useful with the output

Final Considerations

- Though hierarchical clustering is lauded for being assumption free in that we don't select k *a priori*, we actually do need to select k, but *post hoc*
- In other words, we fit our algorithm, we plotted our dendrogram, and we now have to do something useful with the output
- The "useful" thing is to cut the tree where clusters exist, which is typically where branches are longest on the Y axis

Final Considerations

- Though hierarchical clustering is lauded for being assumption free in that we don't select k *a priori*, we actually do need to select k, but *post hoc*
- In other words, we fit our algorithm, we plotted our dendrogram, and we now have to do something useful with the output
- The "useful" thing is to cut the tree where clusters exist, which is typically where branches are longest on the Y axis
- So how many clusters *should* accurately characterize the data...?

Final Considerations

- Though hierarchical clustering is lauded for being assumption free in that we don't select k *a priori*, we actually do need to select k, but *post hoc*
- In other words, we fit our algorithm, we plotted our dendrogram, and we now have to do something useful with the output
- The "useful" thing is to cut the tree where clusters exist, which is typically where branches are longest on the Y axis
- So how many clusters *should* accurately characterize the data...?
- That's inherently subjective and largely up to the researcher

Final Considerations

- Though hierarchical clustering is lauded for being assumption free in that we don't select k *a priori*, we actually do need to select k, but *post hoc*
- In other words, we fit our algorithm, we plotted our dendrogram, and we now have to do something useful with the output
- The "useful" thing is to cut the tree where clusters exist, which is typically where branches are longest on the Y axis
- So how many clusters *should* accurately characterize the data...?
- That's inherently subjective and largely up to the researcher
- HC is usually most useful in comparison to other approaches allowing for comparison and validation (more next class)

Final Considerations

- Though hierarchical clustering is lauded for being assumption free in that we don't select k *a priori*, we actually do need to select k, but *post hoc*
- In other words, we fit our algorithm, we plotted our dendrogram, and we now have to do something useful with the output
- The "useful" thing is to cut the tree where clusters exist, which is typically where branches are longest on the Y axis
- So how many clusters *should* accurately characterize the data...?
- That's inherently subjective and largely up to the researcher
- HC is usually most useful in comparison to other approaches allowing for comparison and validation (more next class)
- Note: HC is computationally inefficient and expensive, especially on large datasets

Lecture Outline

Divisive Hierarchical Clustering

- Divisive clustering is often referred to “top down” clustering

Divisive Hierarchical Clustering

- Divisive clustering is often referred to “top down” clustering
- We begin by assigning all of observations to a single cluster

Divisive Hierarchical Clustering

- Divisive clustering is often referred to “top down” clustering
- We begin by assigning all of observations to a single cluster
- We then partition the cluster into two *least* similar clusters, of all possible split values

Divisive Hierarchical Clustering

- Divisive clustering is often referred to "top down" clustering
- We begin by assigning all of observations to a single cluster
- We then partition the cluster into two *least* similar clusters, of all possible split values
- We proceed recursively on each cluster until each cluster is a singleton

Divisive Hierarchical Clustering

- Divisive clustering is often referred to “top down” clustering
- We begin by assigning all of observations to a single cluster
- We then partition the cluster into two *least* similar clusters, of all possible split values
- We proceed recursively on each cluster until each cluster is a singleton
- This is significantly more expensive even than agglomerative, given the many split calculations required at each split (hence its less popular)

Lecture Outline

Coming Up

- k-means clustering
- Gaussian mixture models
- Demonstration in `R`: 2012 state Democratic vote shares

Coming Up

- k-means clustering
- Gaussian mixture models
- Demonstration in `R`: 2012 state Democratic vote shares
- First problem set due **Friday at 5 pm** to our GH repo (quick word on the data and concept)