Machine Learning

An overview of unsupervised methods

William La Cava
Postdoctoral Researcher
Computational Genetics Laboratory
lacava@upenn.edu

October 11, 2018

Outline

Outline

Outline

Outline

Outline

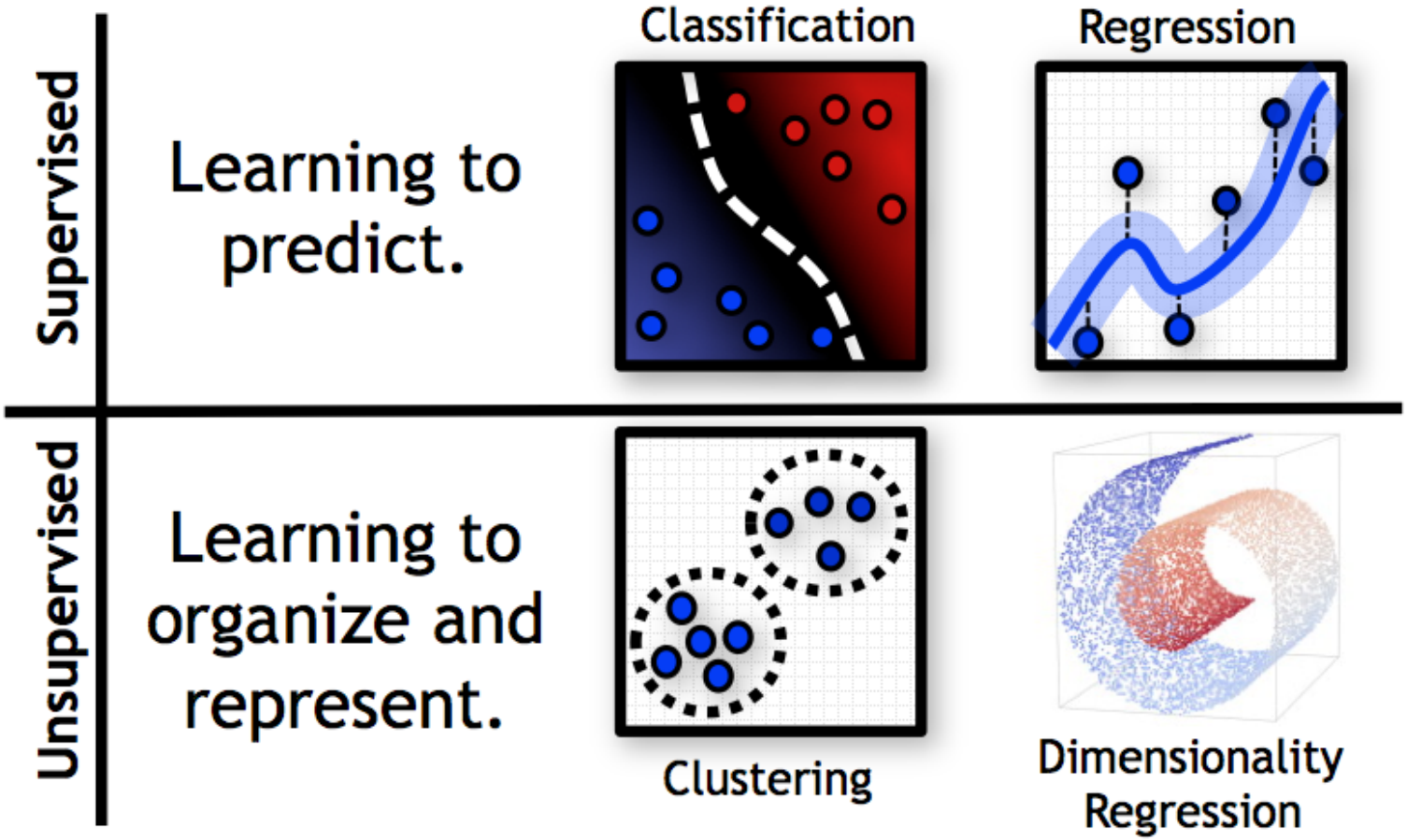
Outline

Outline

Outline

Machine Learning

Tasks



Unsupervised Learning

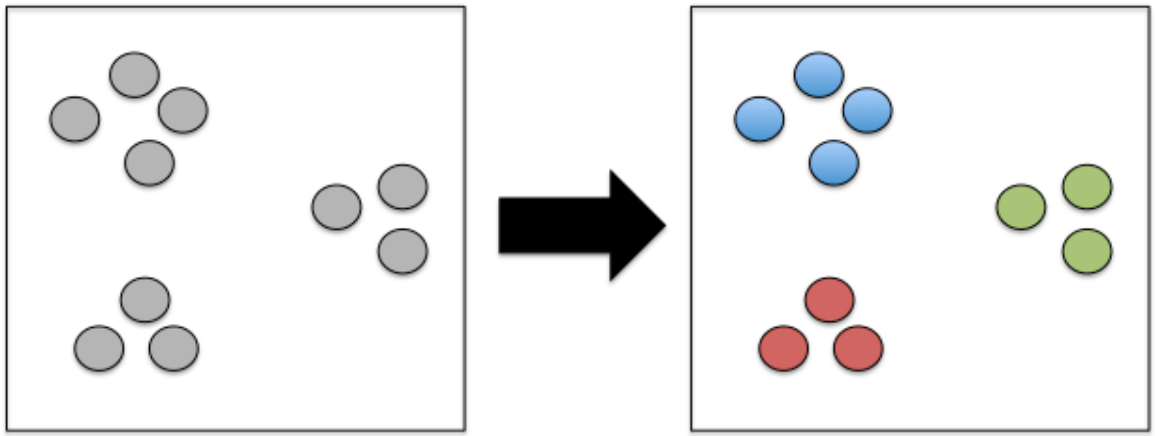
- Set of data: $\{\mathbf{x_i}, i = 1 \ \ldots \ N\}$ with d features

Unsupervised Learning

- Set of data: $\{\mathbf{x_i}, i = 1 \ \ldots \ N\}$ with d features

Definition (Clustering)

Given a set of data $\mathbf{x}$, partition the data cases into groups such that the data cases within each *partition* are more *similar* to each other than they are to data cases in other partitions.

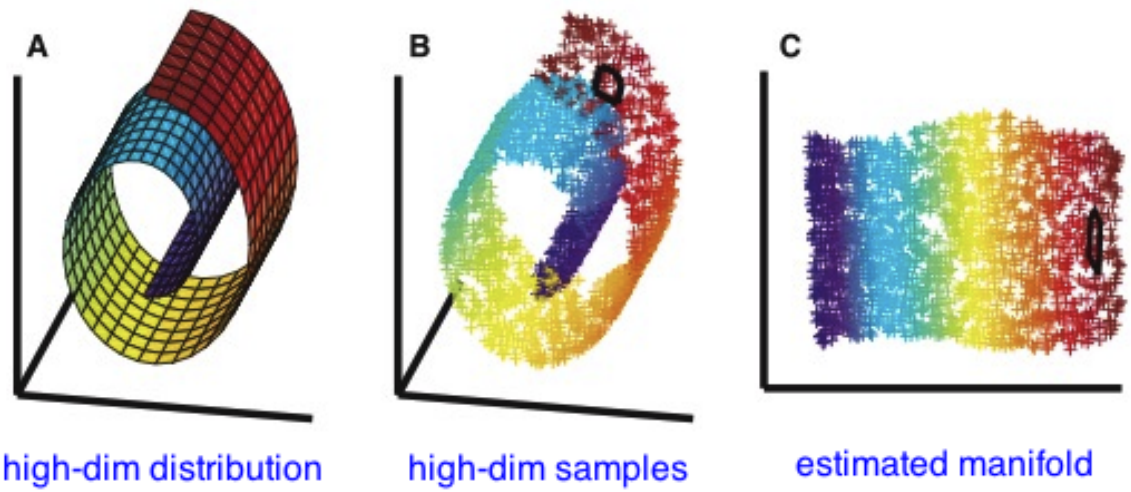


Unsupervised Learning

- Set of data: $\{\mathbf{x_i}, i = 1 \ \ldots \ N\}$ with d features

Definition (Dimensionality Reduction)

Given a set of data $\mathbf{x} \in \mathbb{R}^d$, map the feature vectors into a lower dimensional space $\mathbb{R}^k$ where $k < d$ while preserving certain properties of the data.



Examples

Supervised learning questions:

- *Clinical* What patient health characteristics are predictive of response to this treatment?
- *Genetics* For a cohort of patients, I have measured genotypes and the effective therapeutic dose of a drug. In new patients where I also measured genotypes, what dose should I use?

Examples

Unsupervised learning questions:

- *Clinical* Are there identifiable sub-groups of patients in my data (e.g., patients with similar demographics or that respond similarly to different treatments?)
- *Genetics* Are there patterns of gene expression in biopsies that I collected that suggest patients could be more precisely characterized in different molecular groups?

K-Means

- Attempts to group data into K clusters.
- Begins with randomly initialized centroids, $\mu_1 \ldots \mu_K$
- Two step process:
  1. Calculate distance from every data case to each centroid, and assign clusters accordingly.
  2. Update cluster centers μ to the mean of the data cases assigned to them.
- Keep going until cluster positions stop changing.

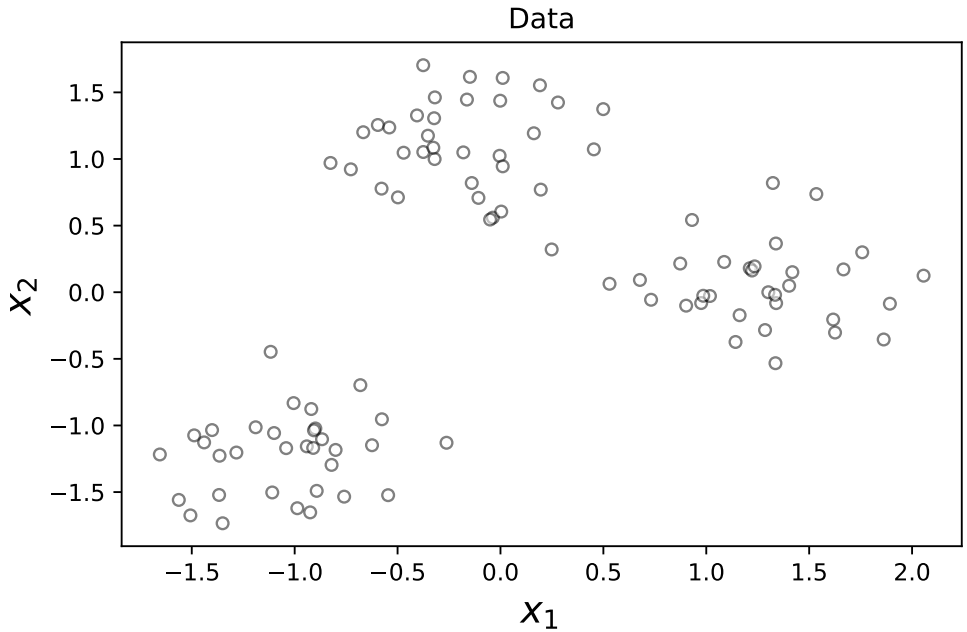
K-Means

- Minimizes the *within-cluster* variation:

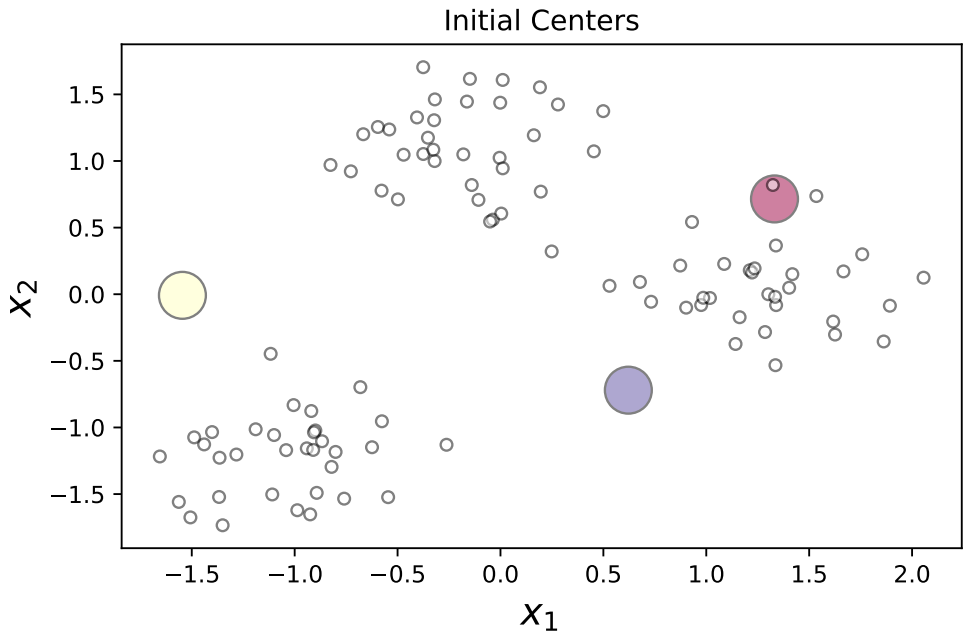
$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \sum_{k=1}^{K} \frac{1}{|\mathcal{C}_k|} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}_k} ||\mathbf{x}_i - \mathbf{x}_j||_2^2$$

- K-Means converges to the local optima of its initial centroid positions.

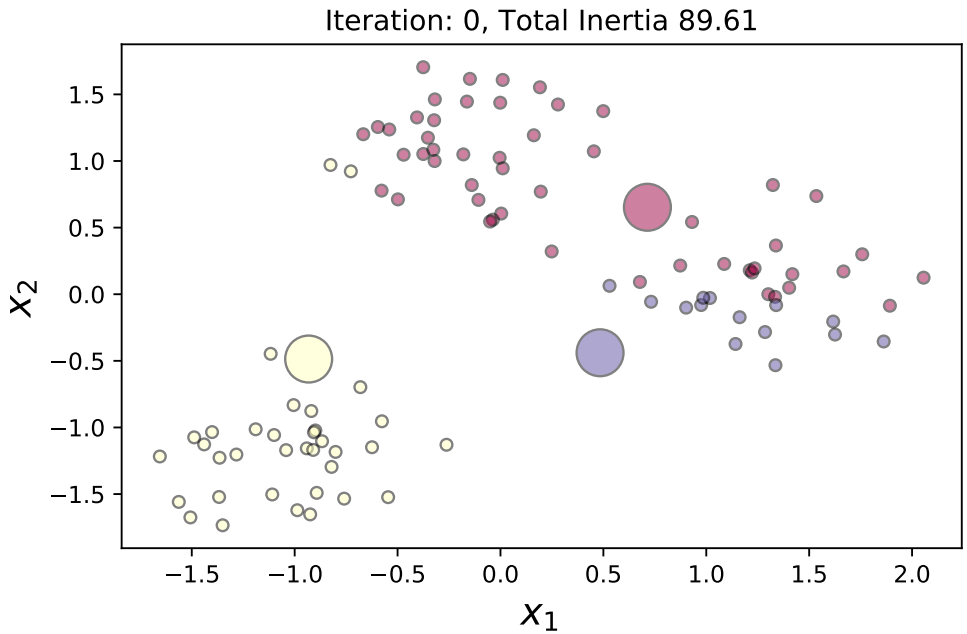
K-Means



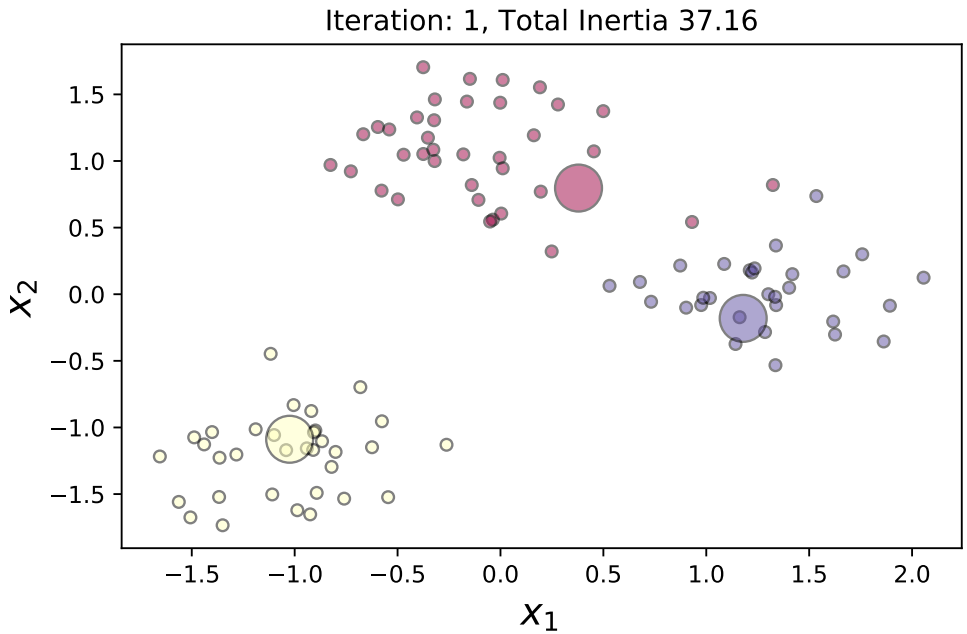
K-Means



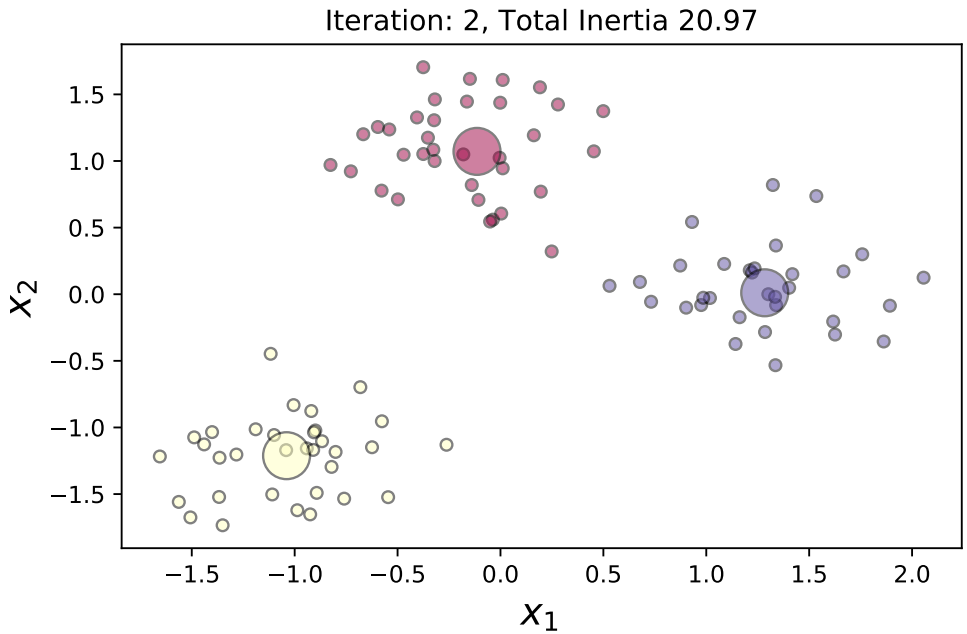
K-Means


Iteration: 0, Total Inertia 89.61
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

K-Means

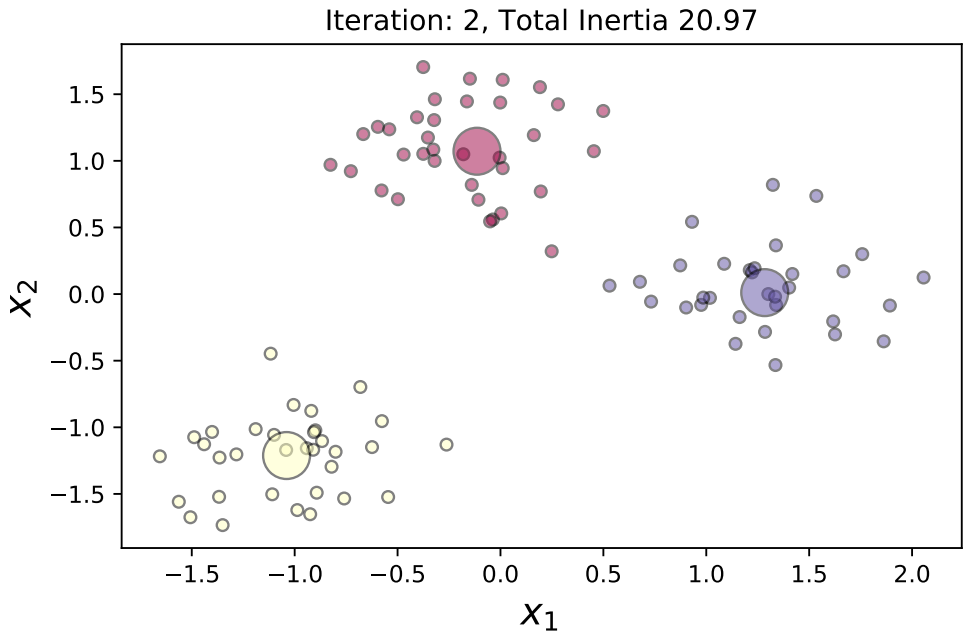

Iteration: 1, Total Inertia 37.16
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

K-Means


Iteration: 2, Total Inertia 20.97
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

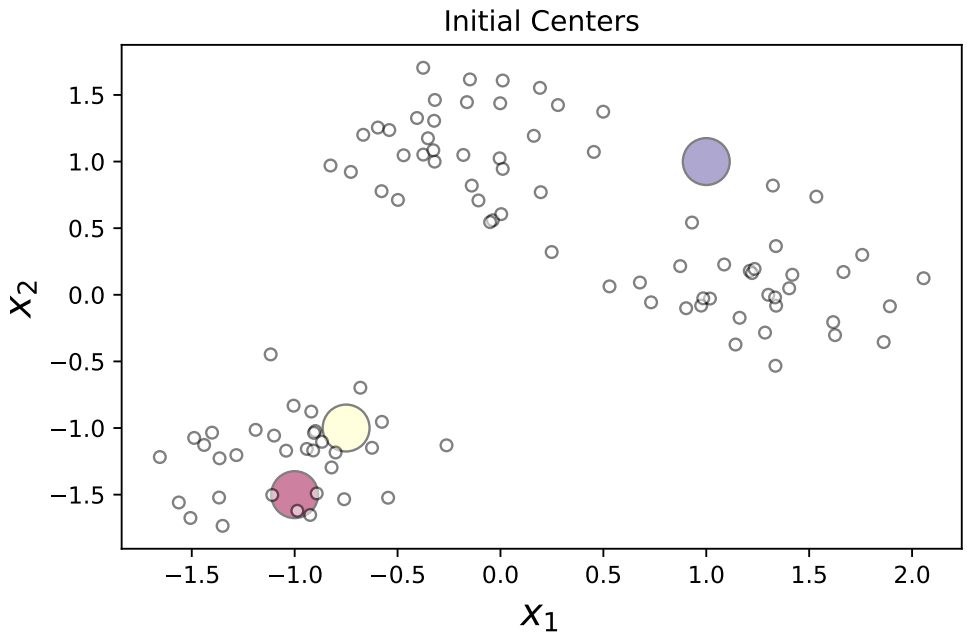
K-Means

Done!


Iteration: 2, Total Inertia 20.97
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

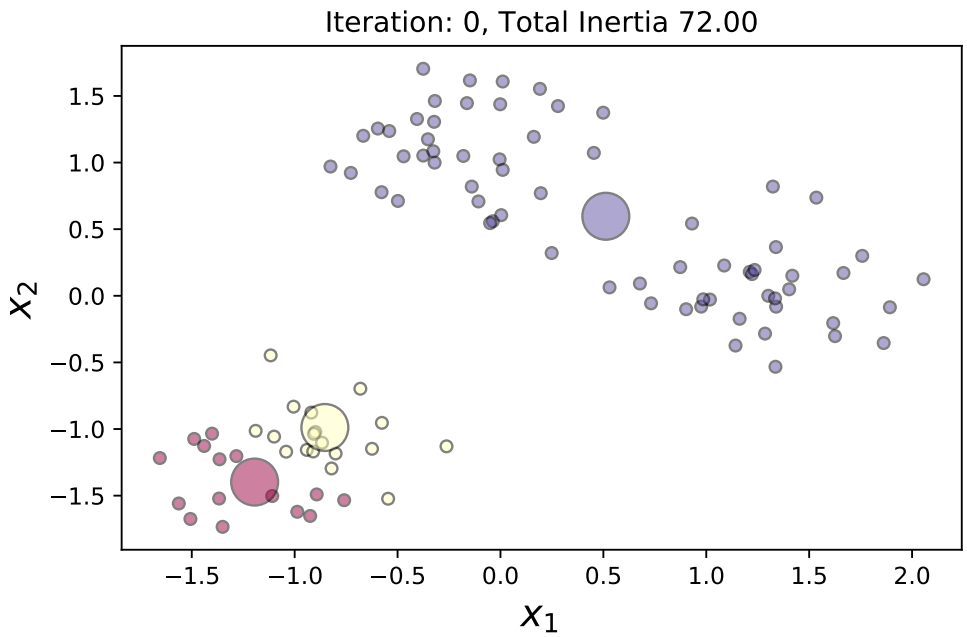
K-Means

Bad Initialization



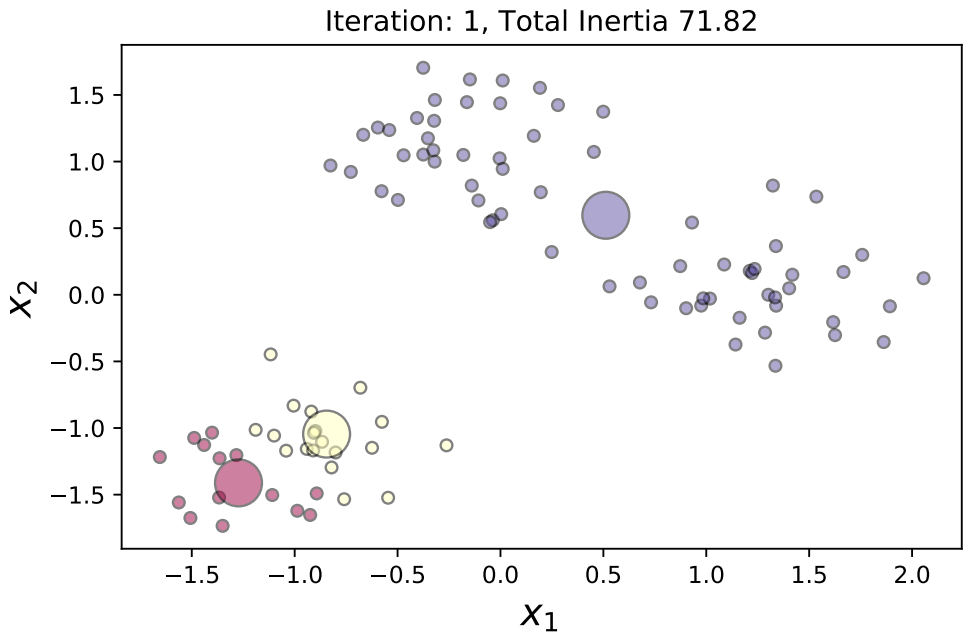
K-Means

Bad Initialization



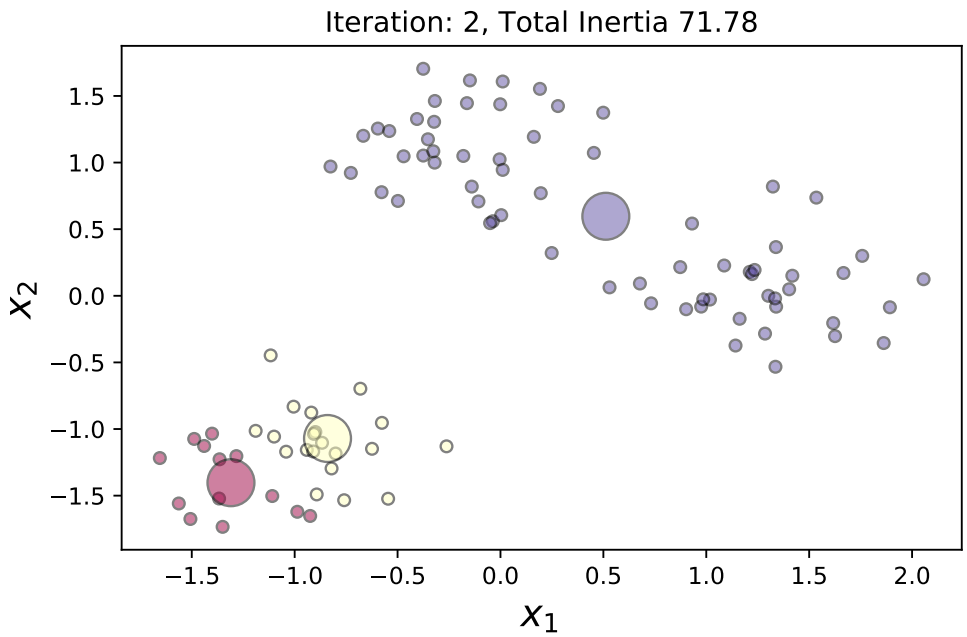
K-Means

Bad Initialization



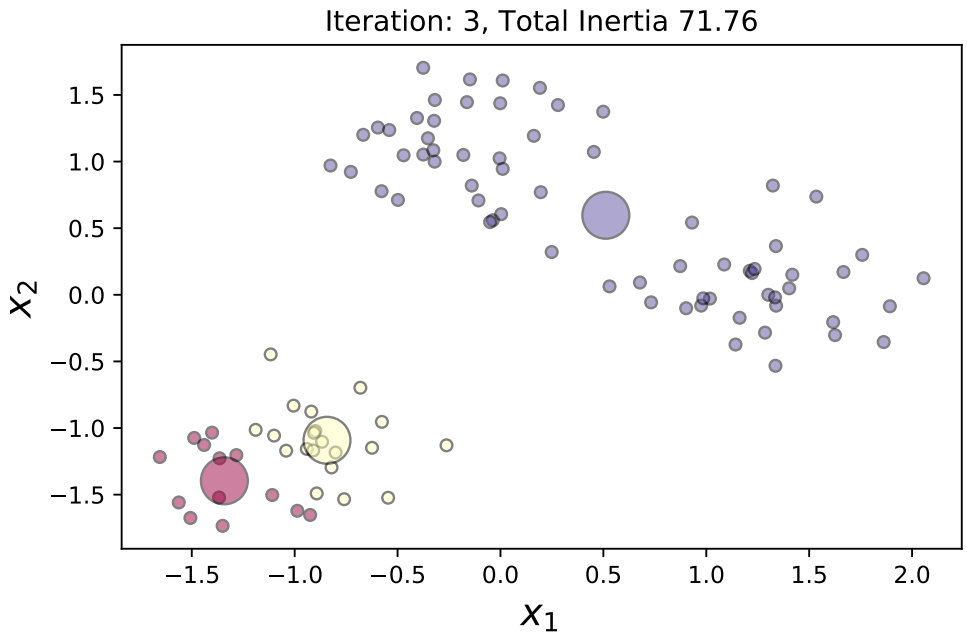
K-Means

Bad Initialization



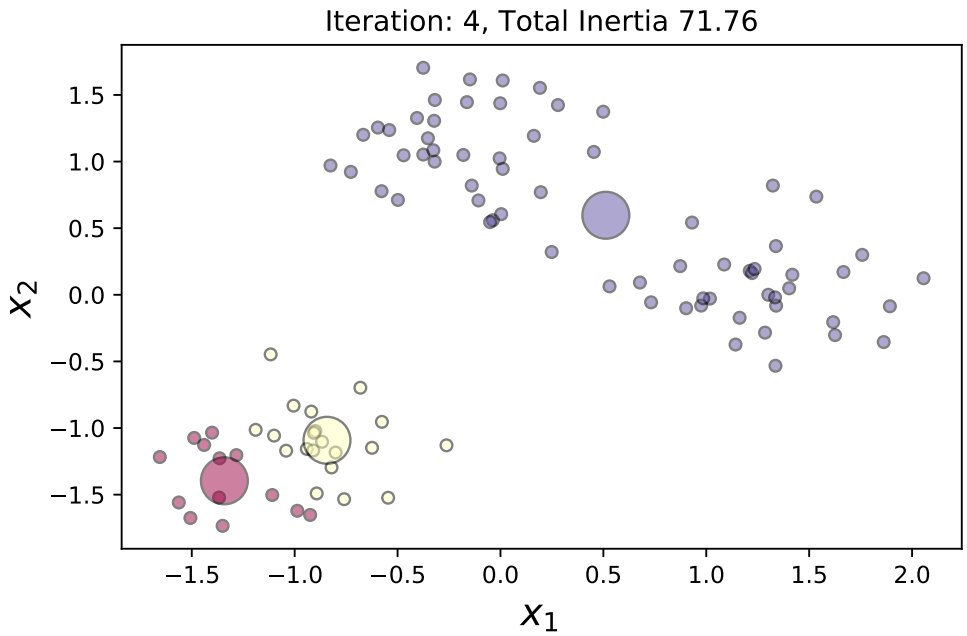
K-Means

Bad Initialization



K-Means

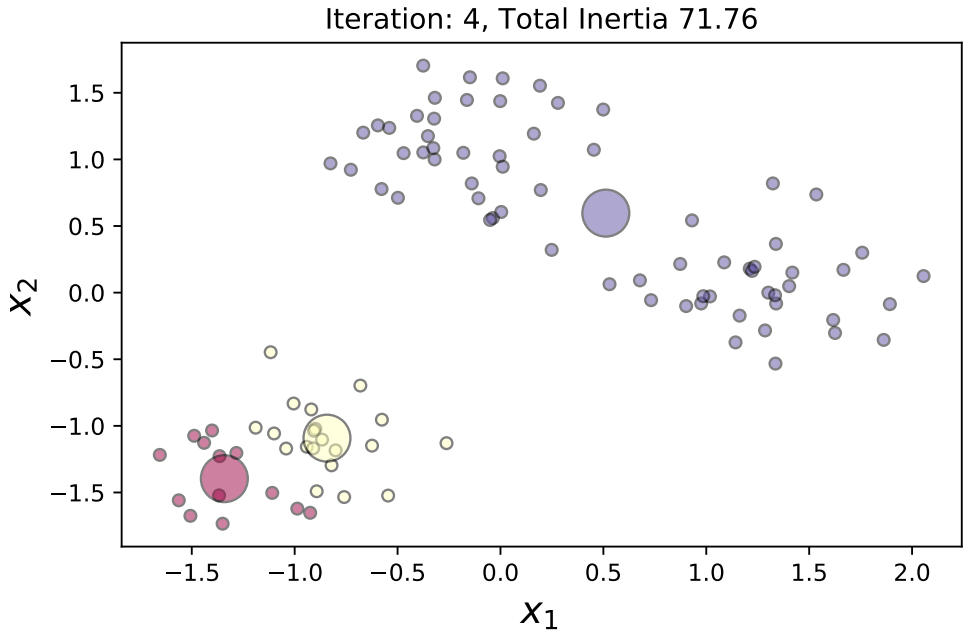
Bad Initialization



K-Means

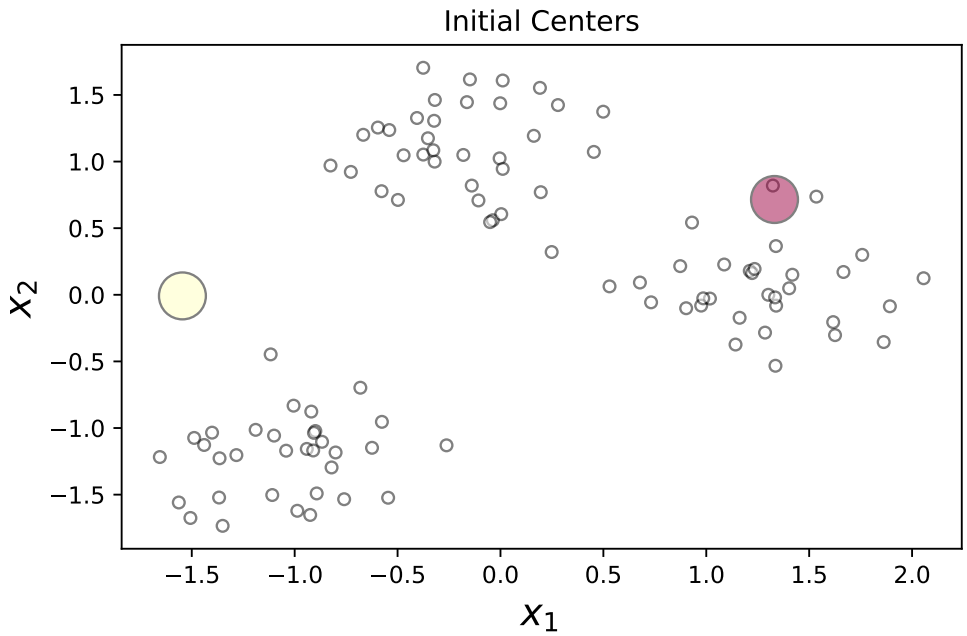
Bad Initialization

Done!



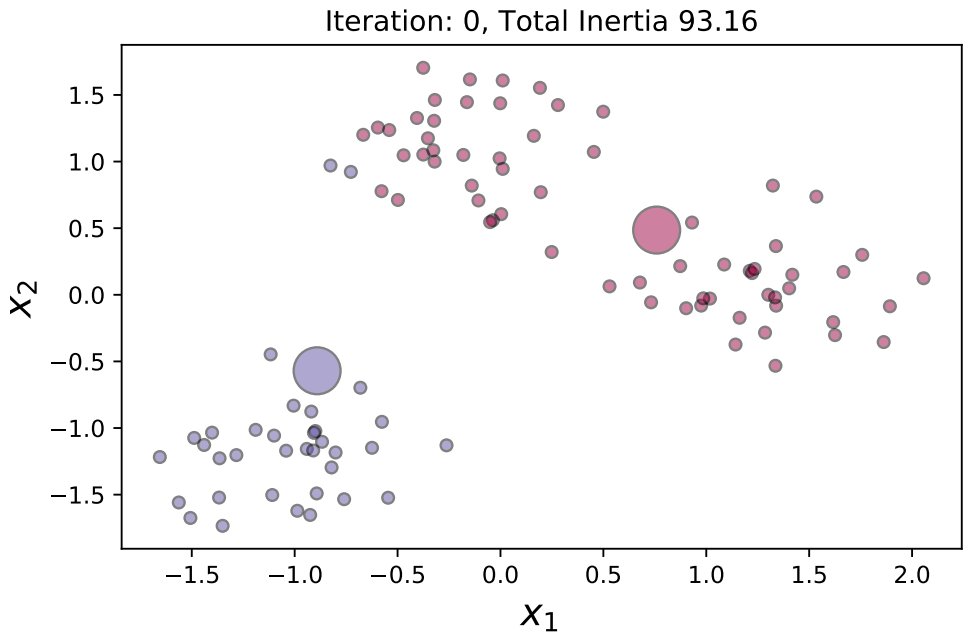
K-Means

Not Enough Clusters



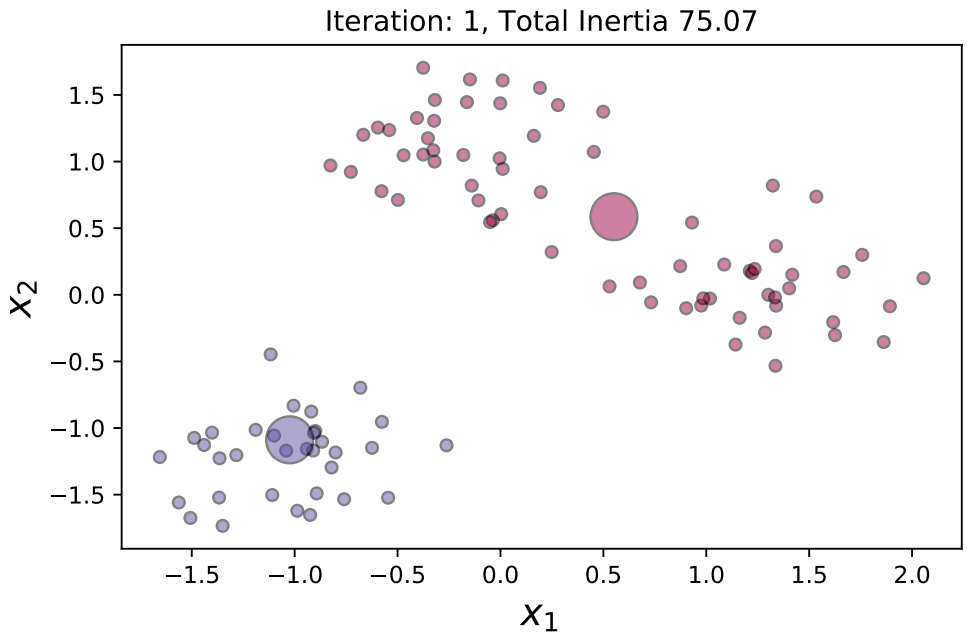
K-Means

Not Enough Clusters



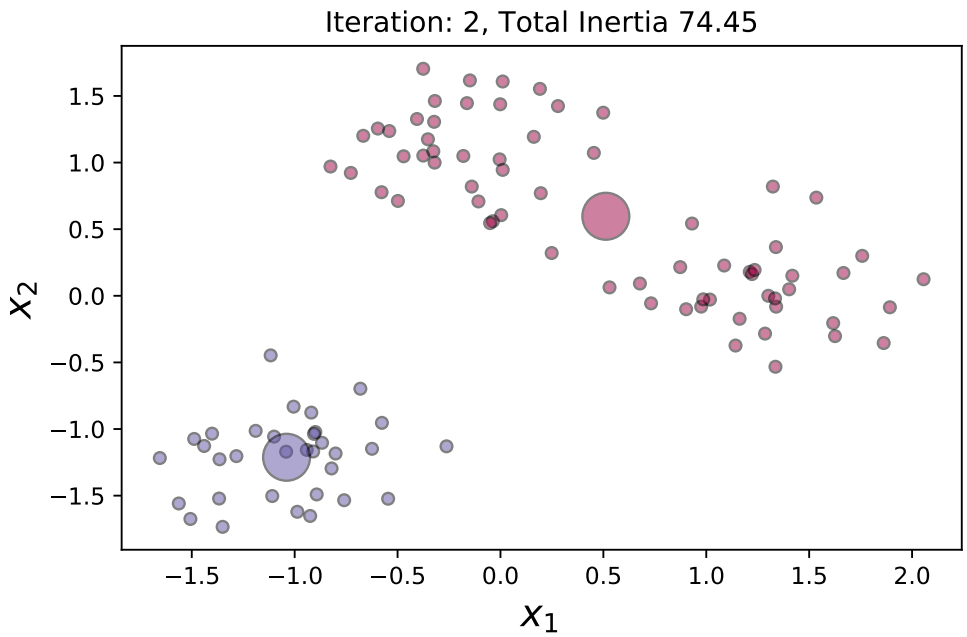
K-Means

Not Enough Clusters



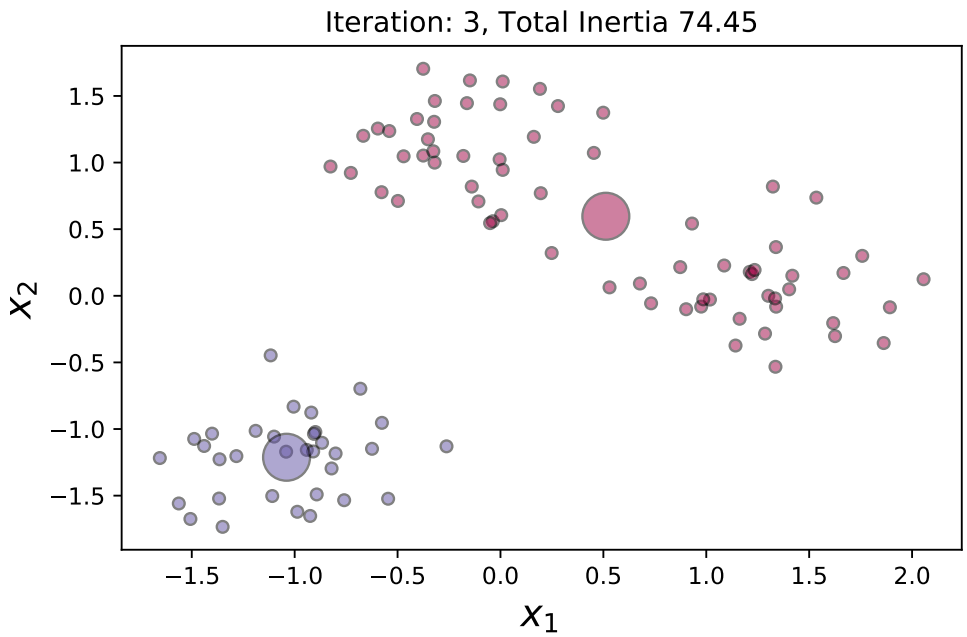
K-Means

Not Enough Clusters



K-Means

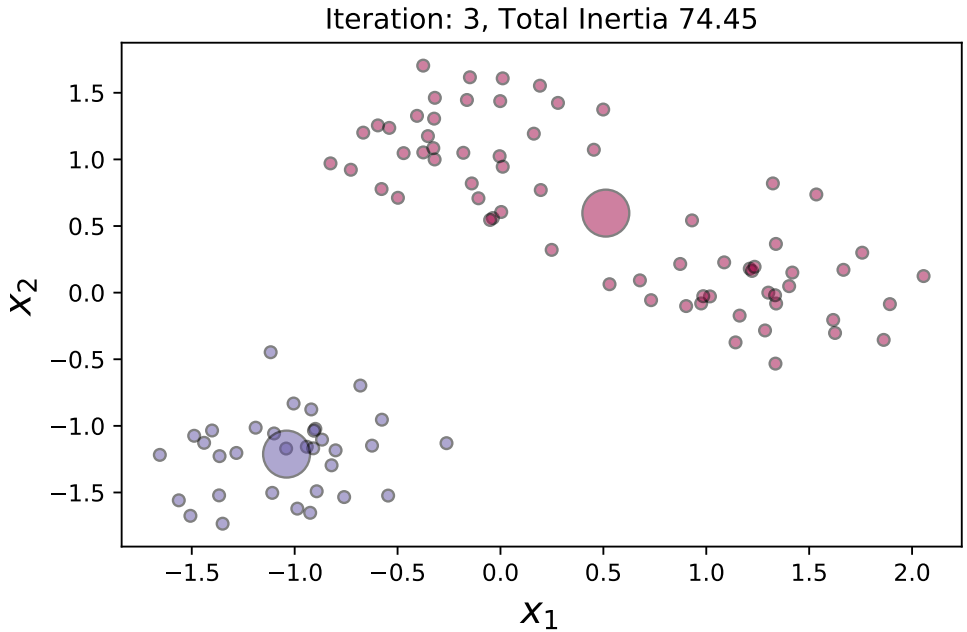
Not Enough Clusters



K-Means

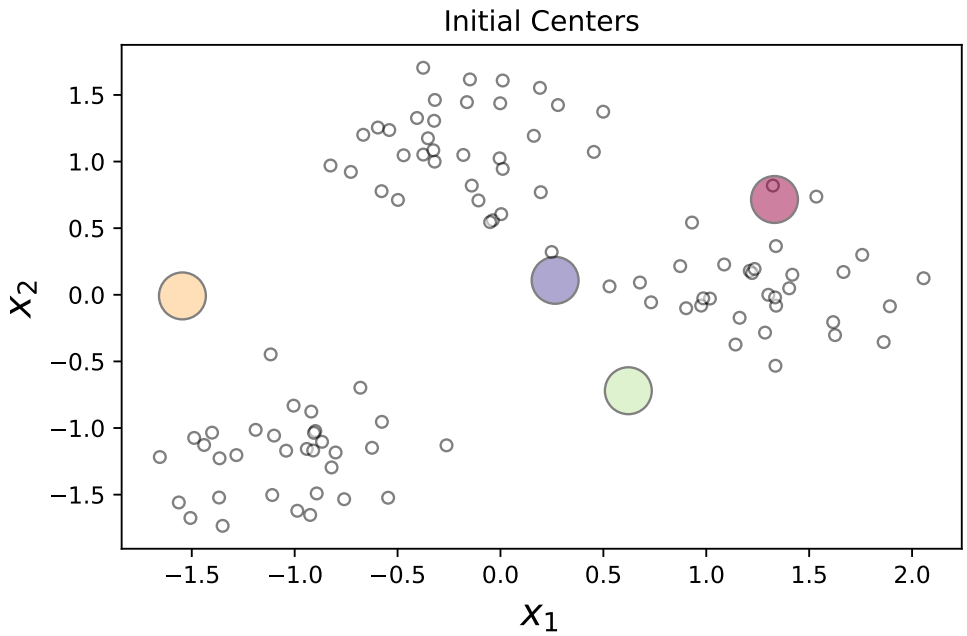
Not Enough Clusters

Done!



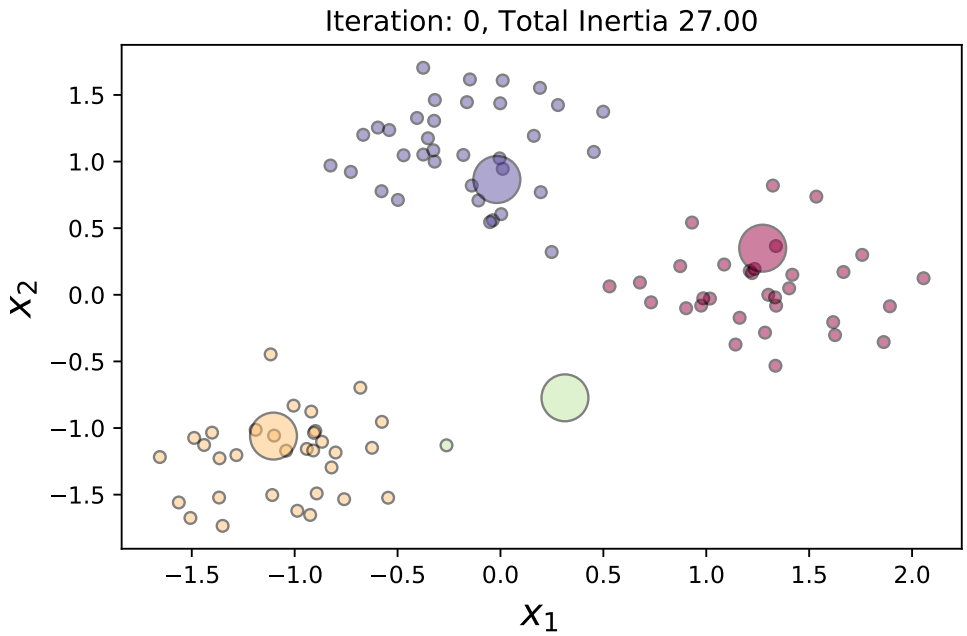
K-Means

Too Many Clusters



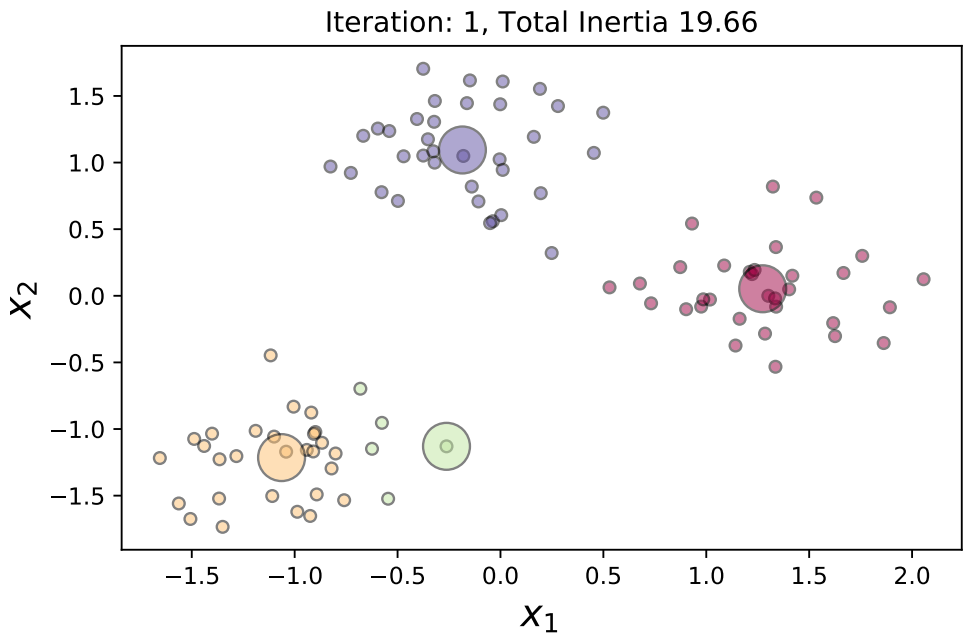
K-Means

Too Many Clusters



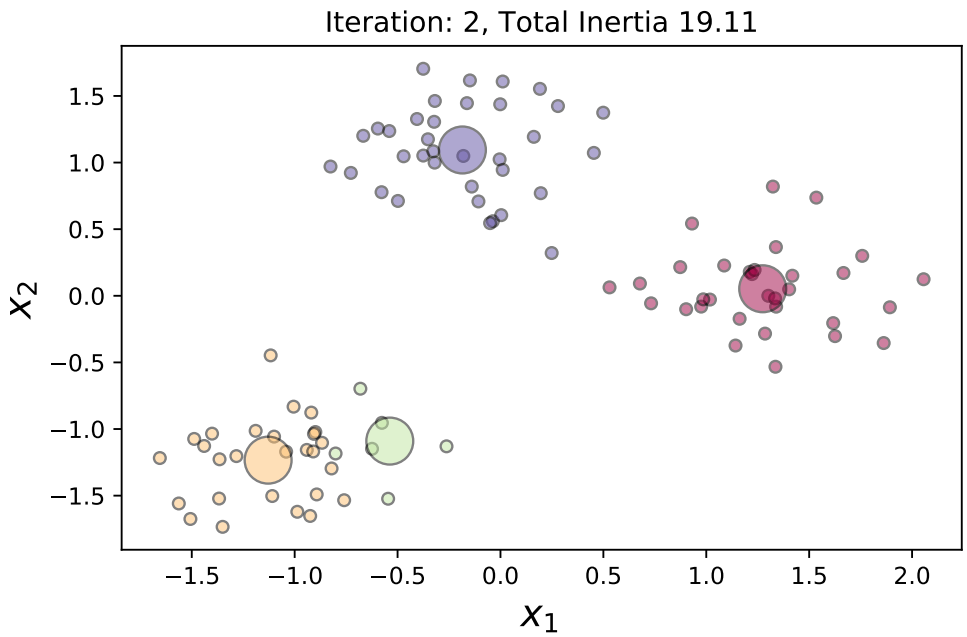
K-Means

Too Many Clusters



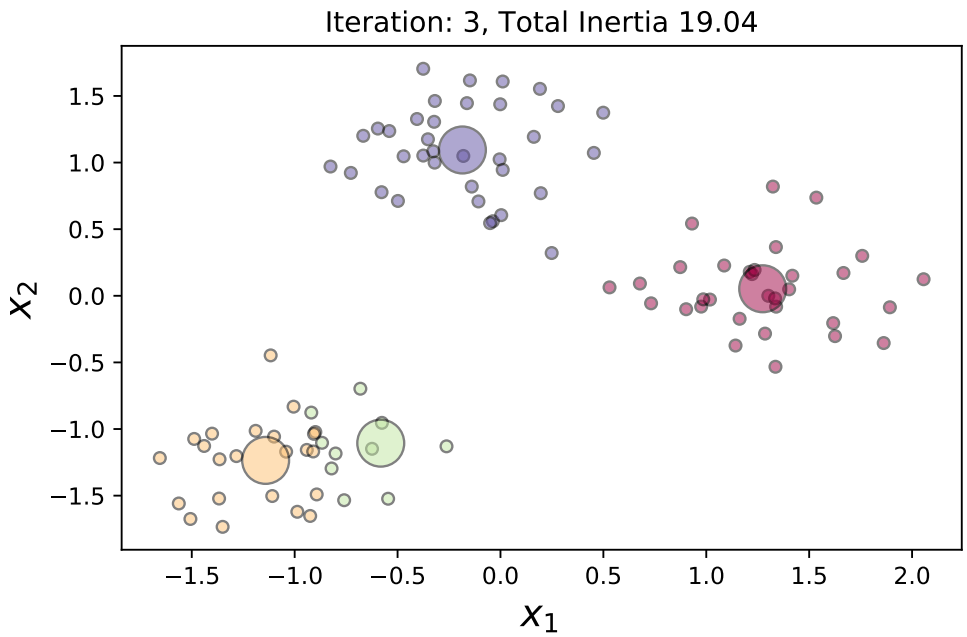
K-Means

Too Many Clusters



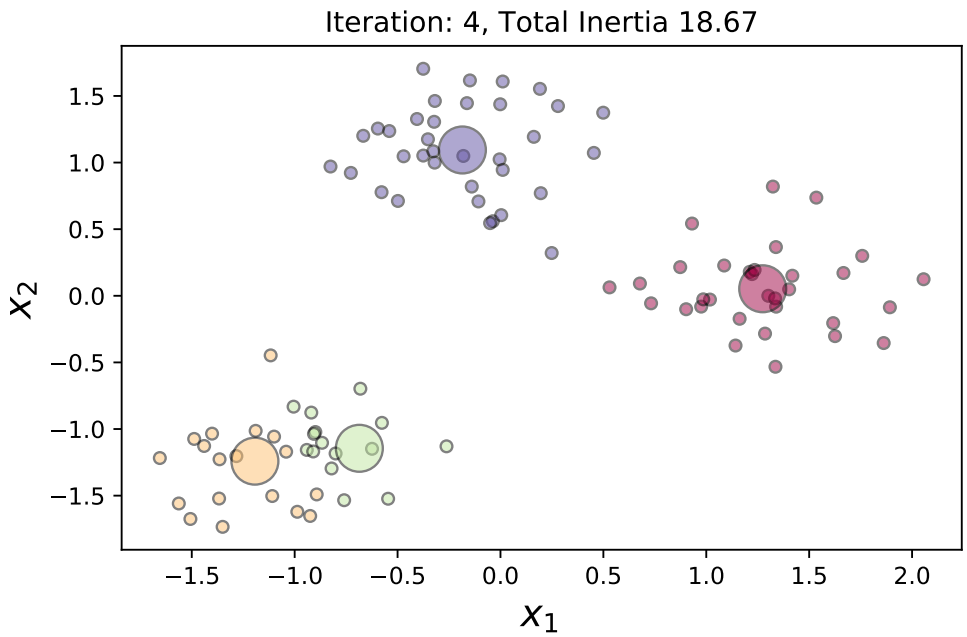
K-Means

Too Many Clusters



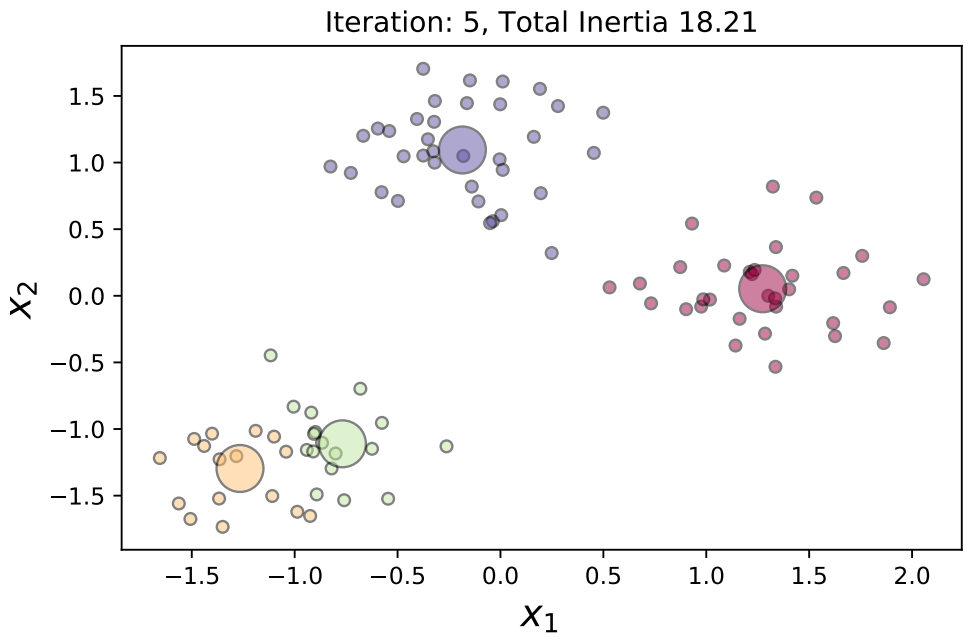
K-Means

Too Many Clusters



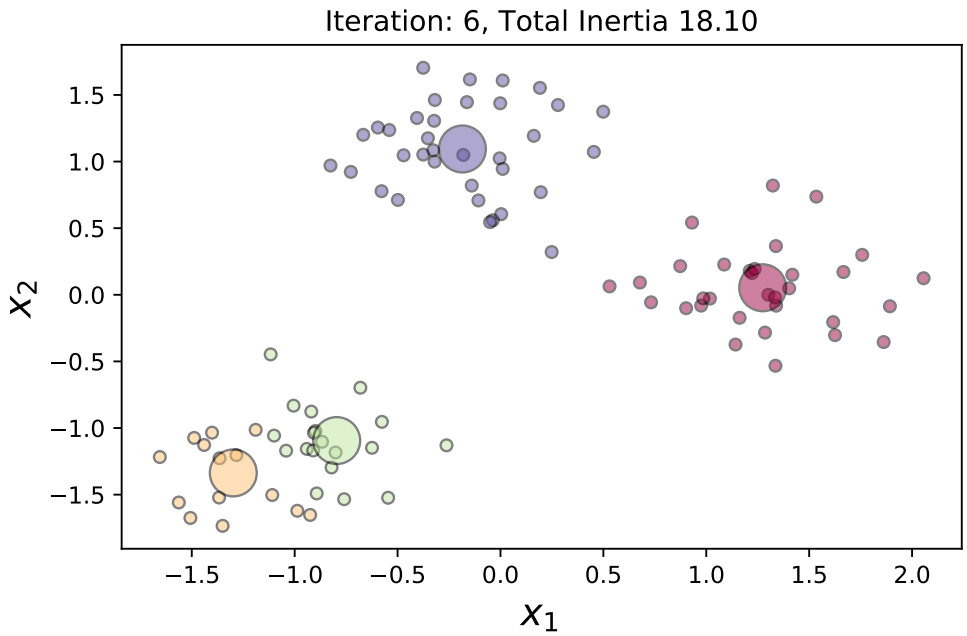
K-Means

Too Many Clusters



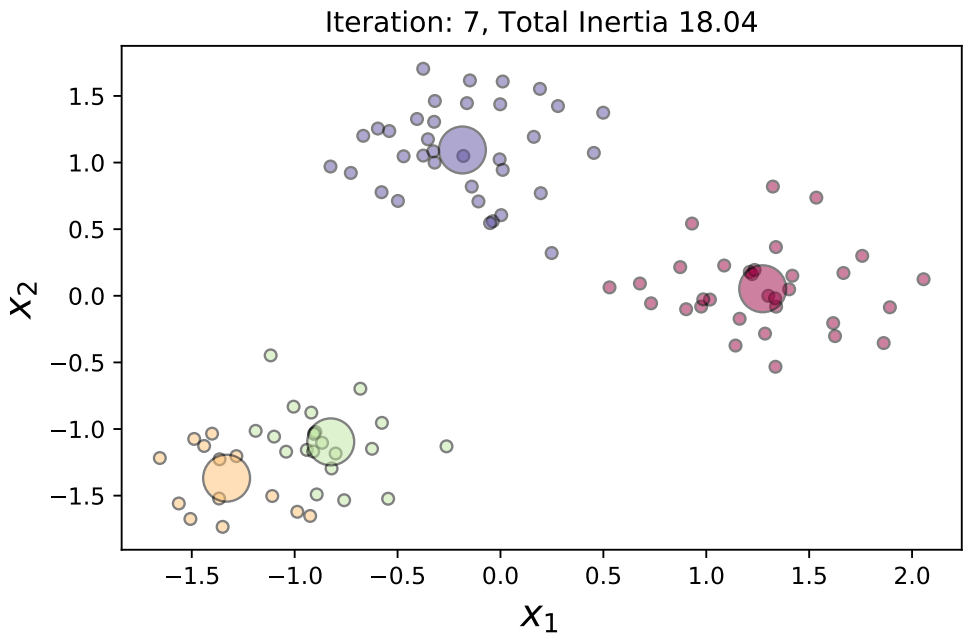
K-Means

Too Many Clusters



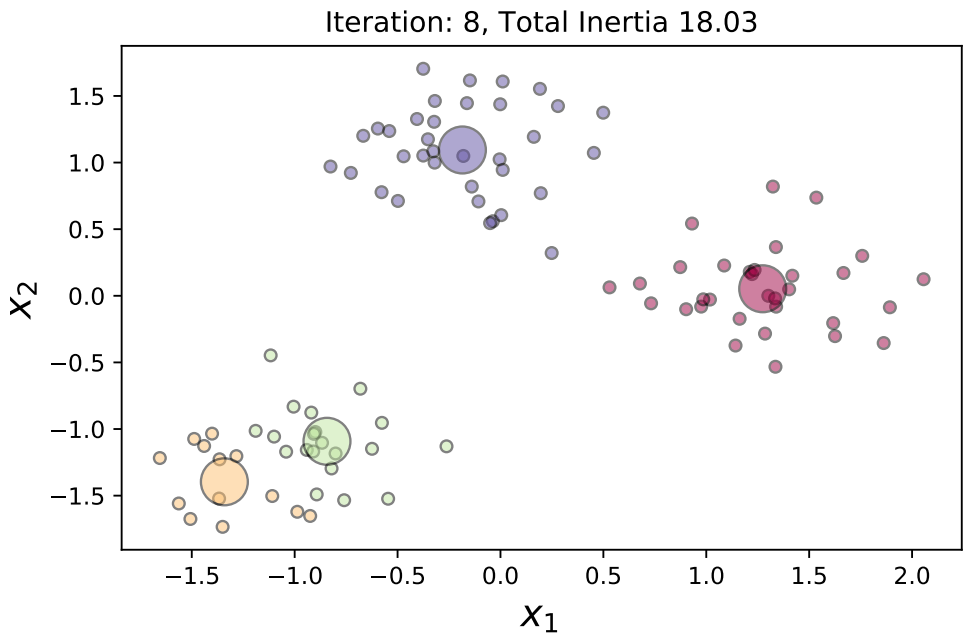
K-Means

Too Many Clusters



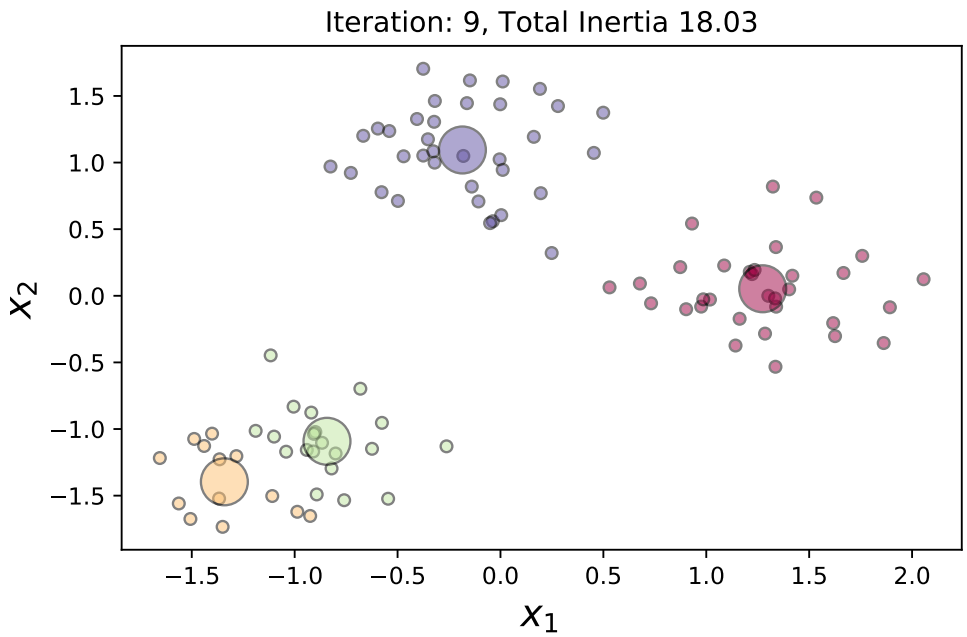
K-Means

Too Many Clusters



K-Means

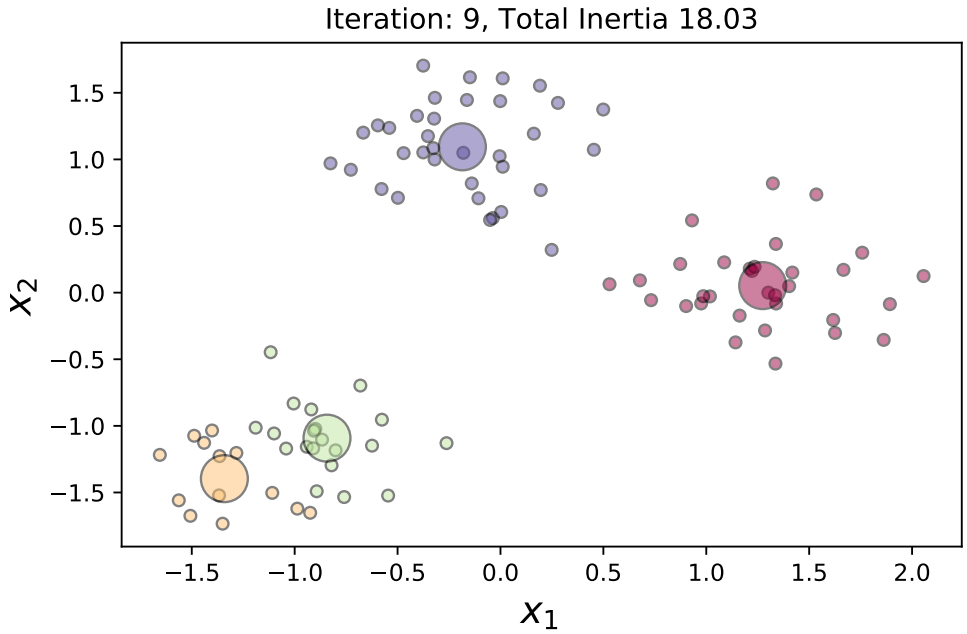
Too Many Clusters



K-Means

Too Many Clusters

Done!

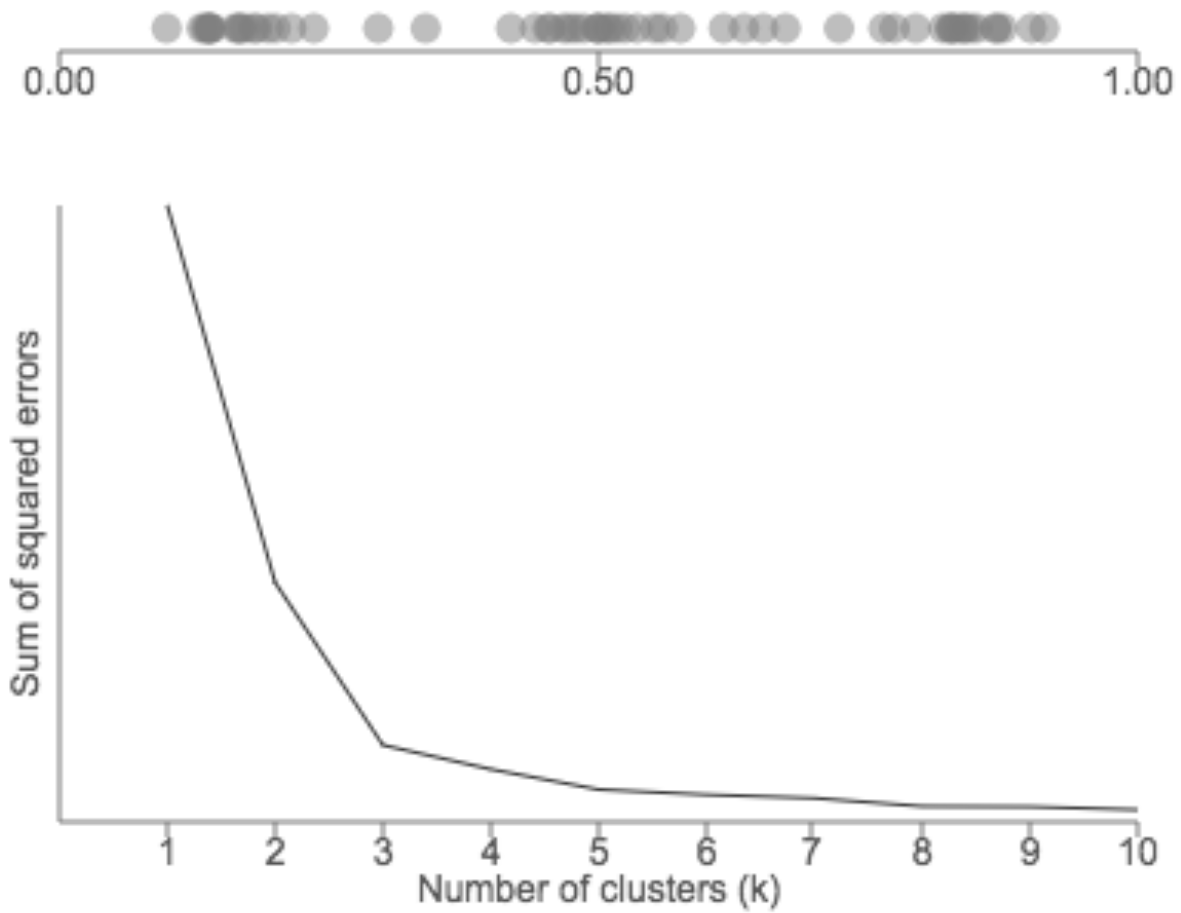


K-Means

- Sensitive to initial centroid positions.
- Sensitive to scaling of data dimensions.
- How to choose K?

K-Means

Elbow Method



Hierarchical Agglomerative Clustering

- *Hierarchical Clustering*: greedy tree-based clustering methods
- *Hierarchical Agglomerative Clustering* (HAC): the most popular type
  1. Start with all data cases assigned to own clusters.
  2. Greedily and recursively merge pairs of clusters.

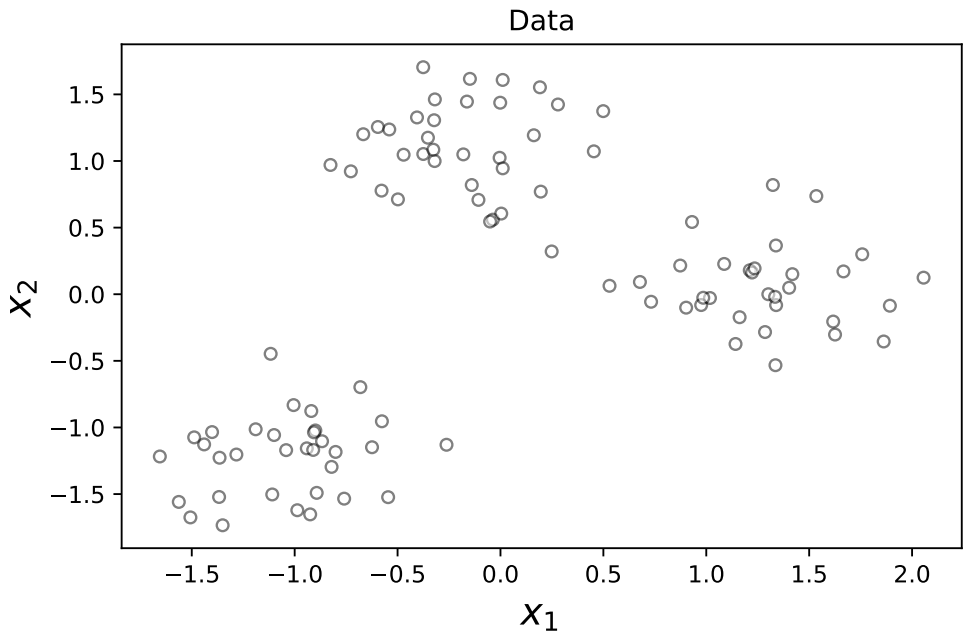
Hierarchical Agglomerative Clustering

Algorithm

HAC

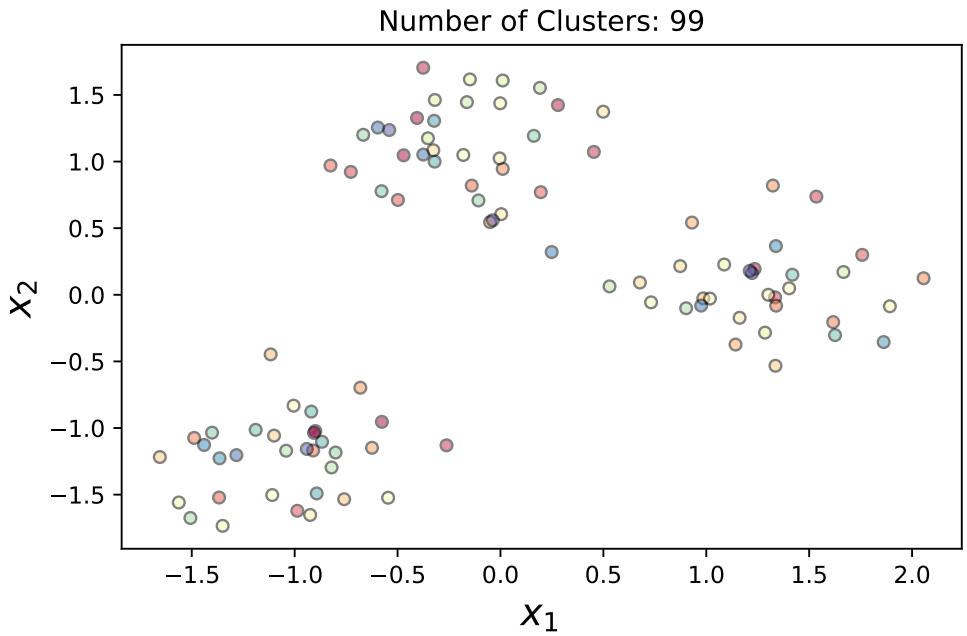
1. Start with all data cases assigned to own clusters.
2. Calculate all pairwise distances.
3. for $i = N, N-1, ...2$: $\leftarrow i =$ number of clusters
   1. Merge the two closest clusters among i clusters.
   2. Calculate the pairwise distances between all $i-1$ clusters.

HAC



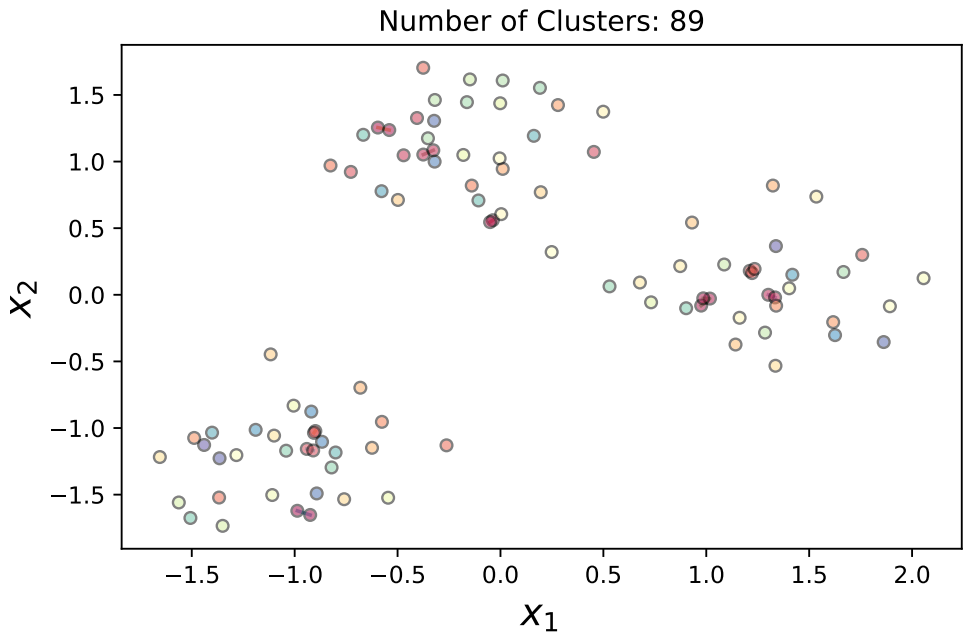
HAC

Example


Number of Clusters: 99
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

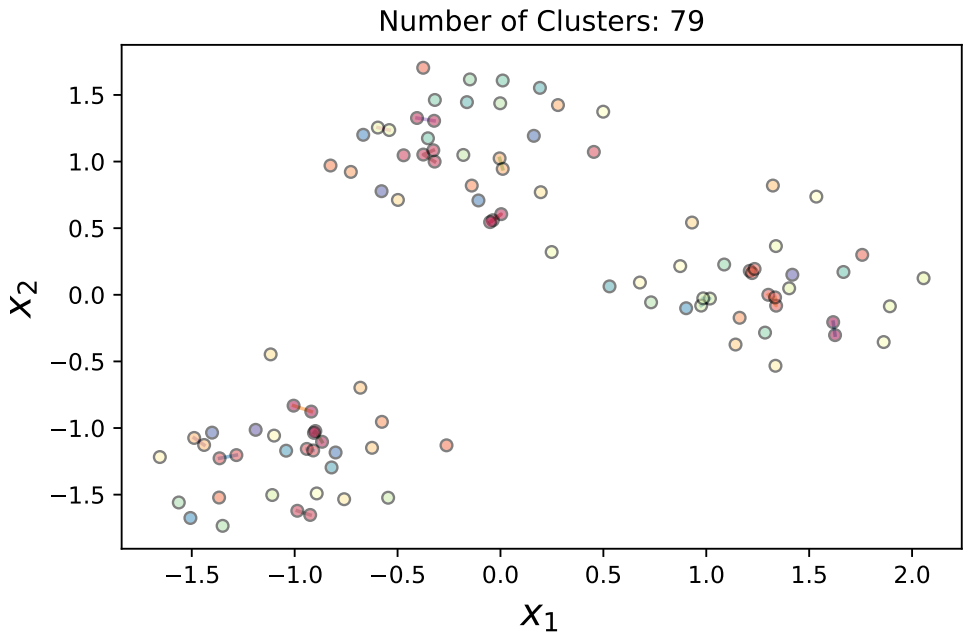
HAC

Example


Number of Clusters: 89
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

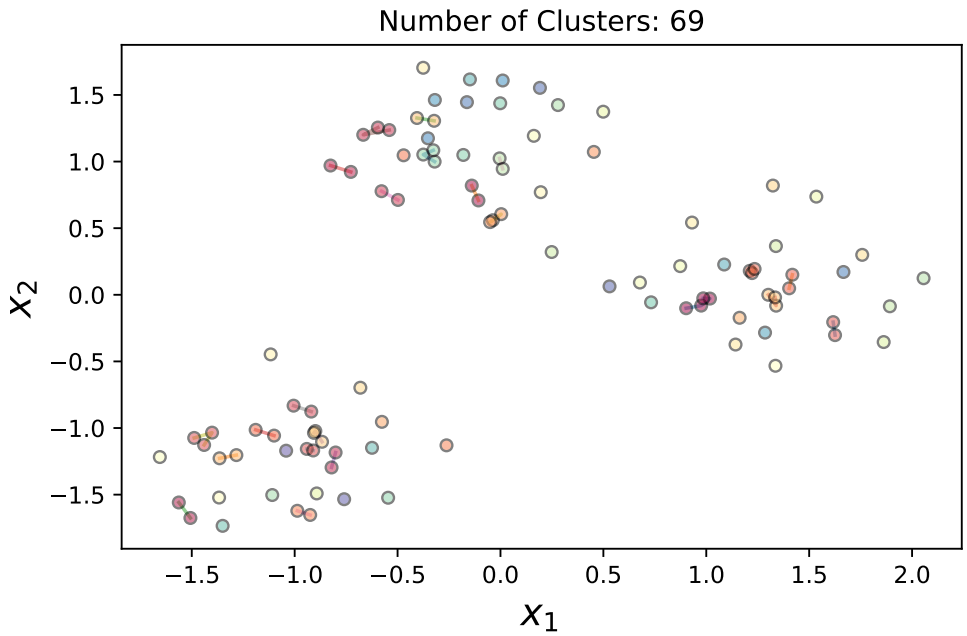
HAC

Example


Number of Clusters: 79
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

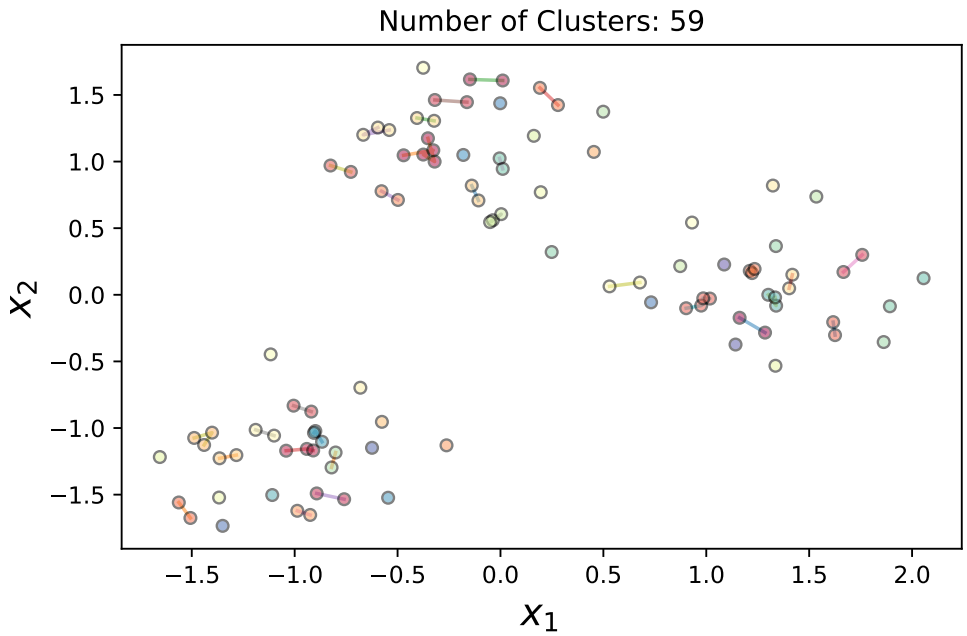
HAC

Example


Number of Clusters: 69
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

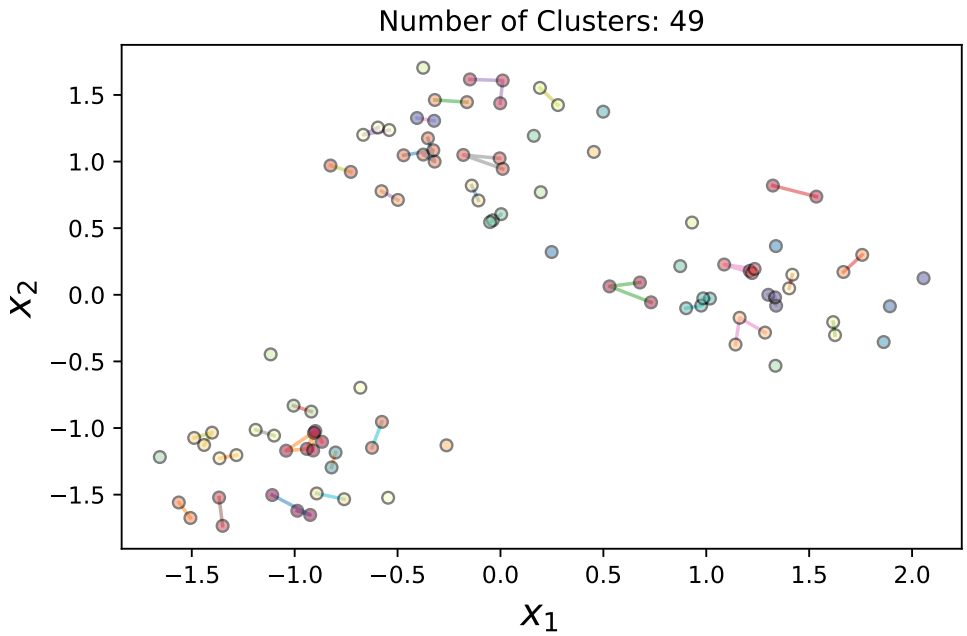
HAC

Example


Number of Clusters: 59
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

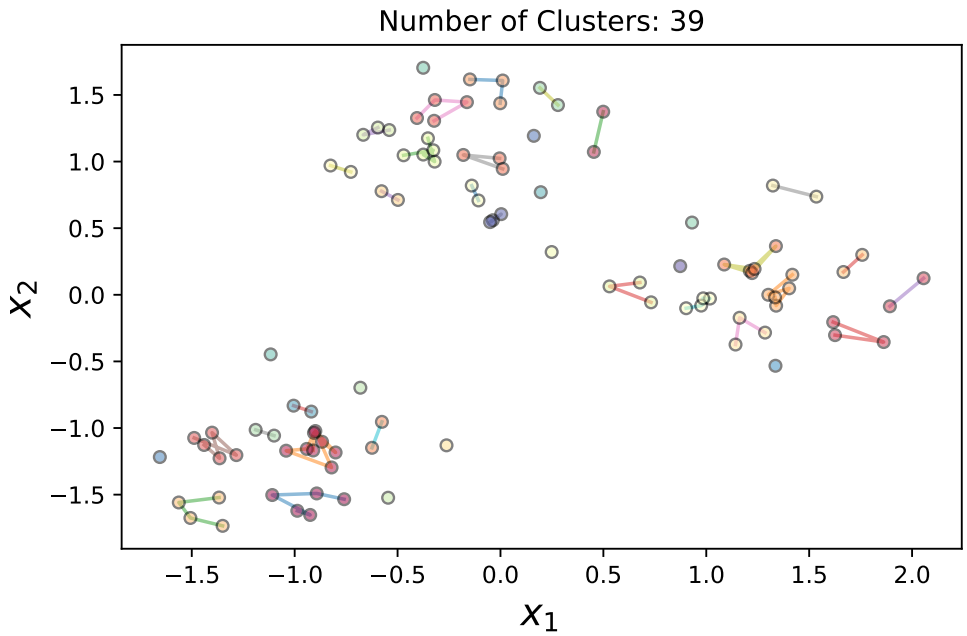
HAC

Example


Number of Clusters: 49
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

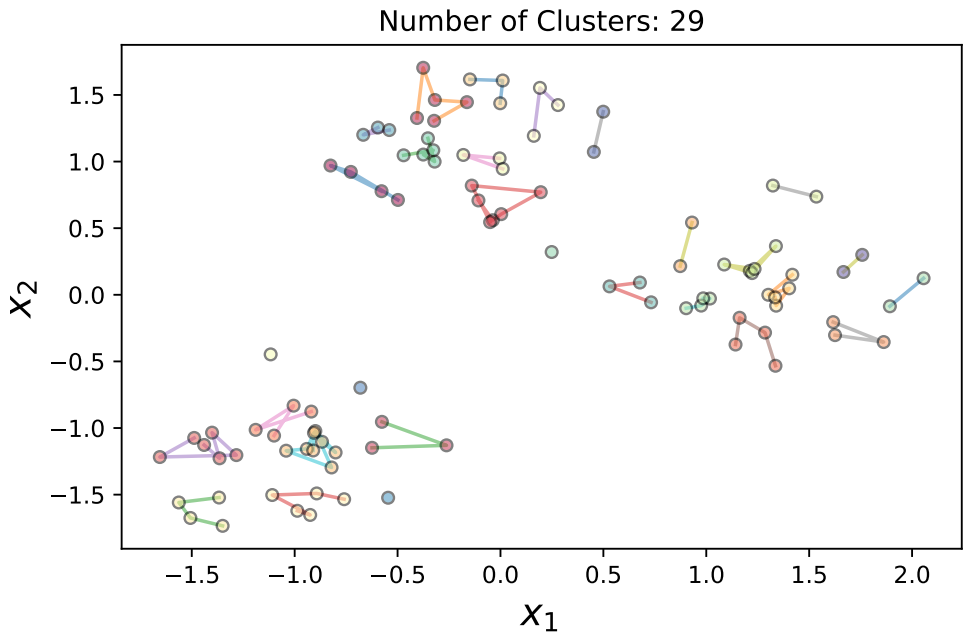
HAC

Example


Number of Clusters: 39
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

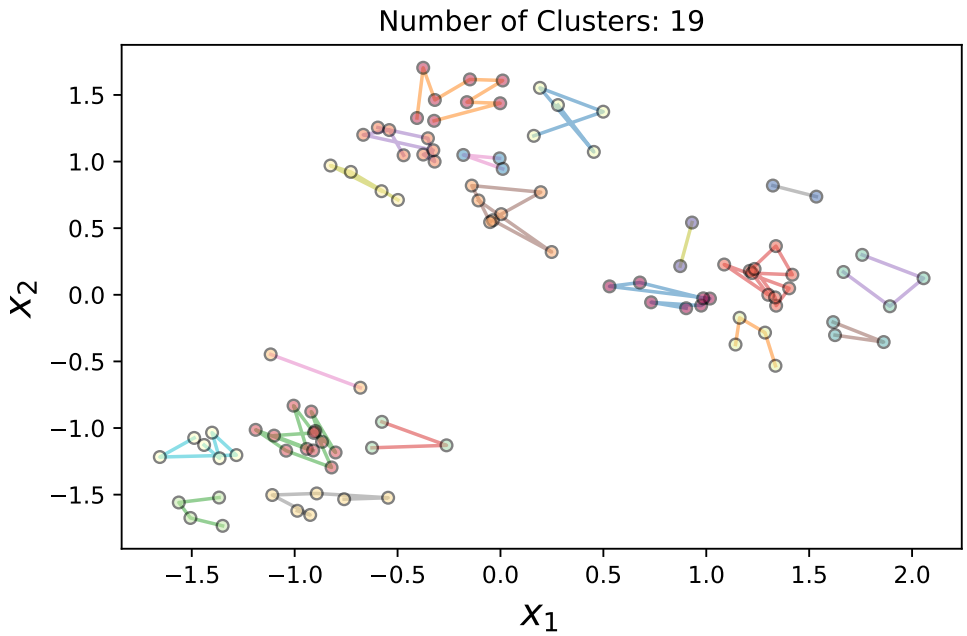
HAC

Example


Number of Clusters: 29
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

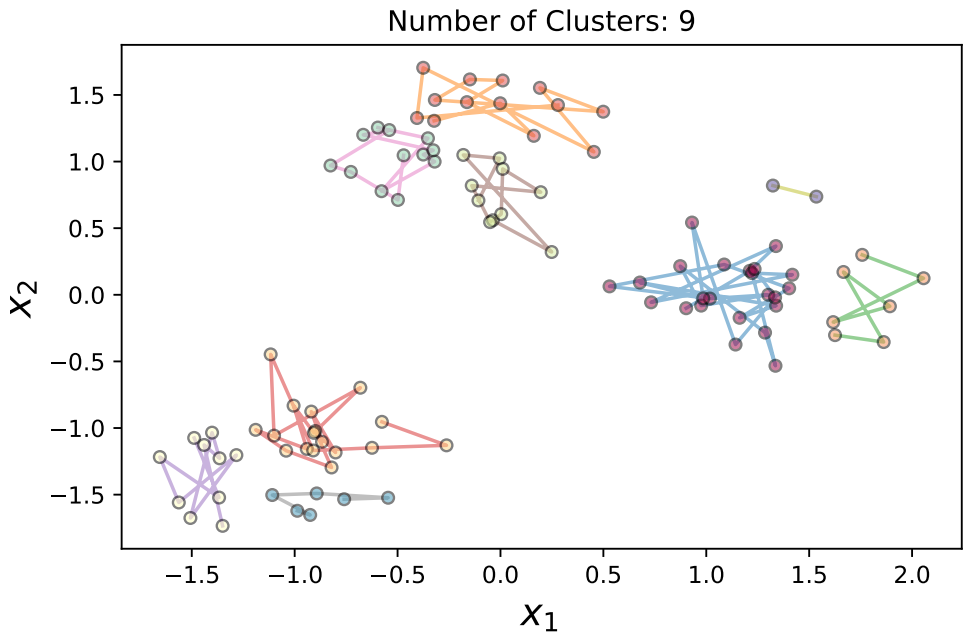
HAC

Example


Number of Clusters: 19
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

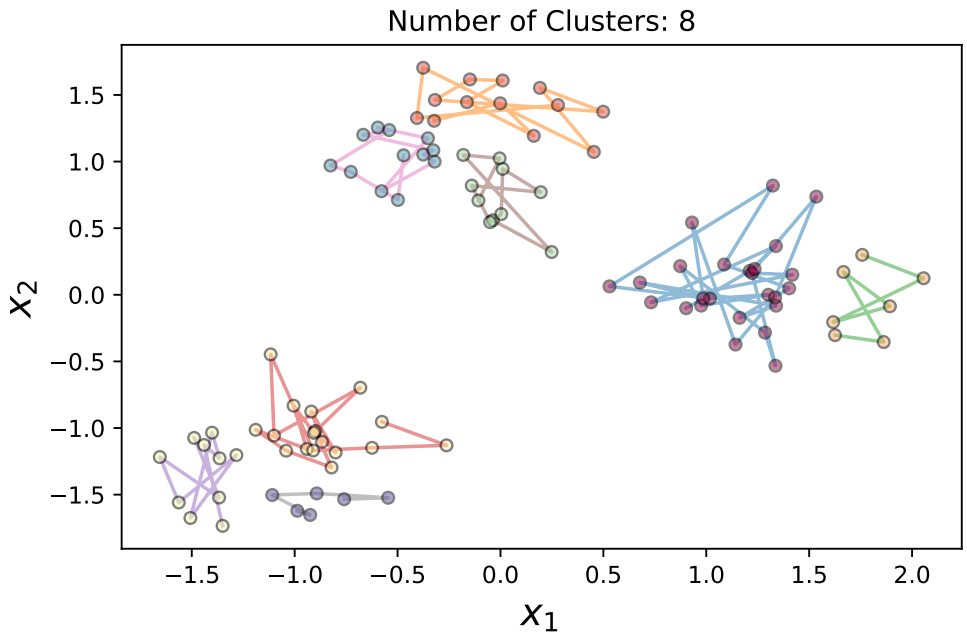
HAC

Example


Number of Clusters: 9
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

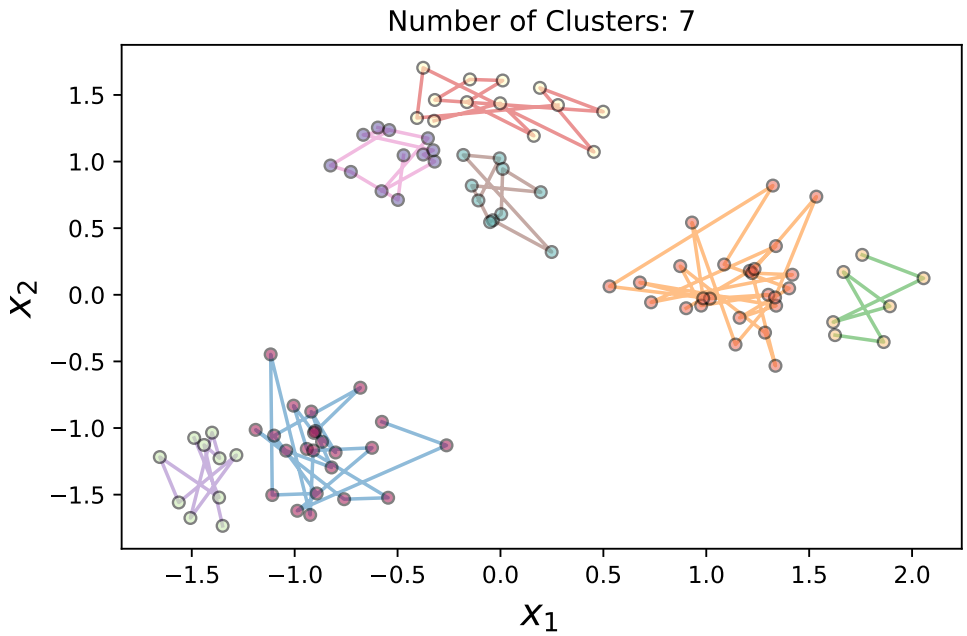
HAC

Example


Number of Clusters: 8
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

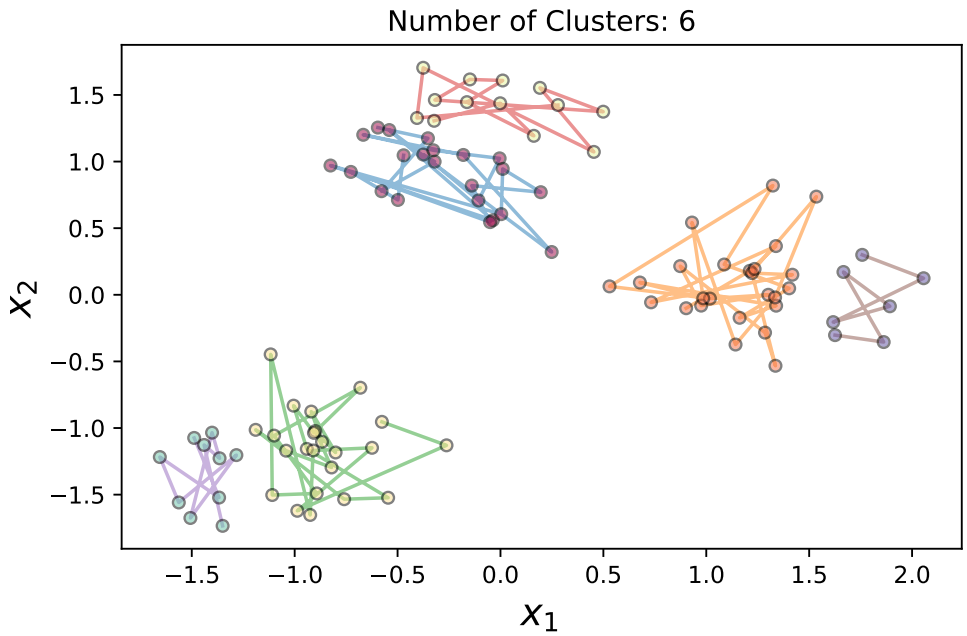
HAC

Example


Number of Clusters: 7
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

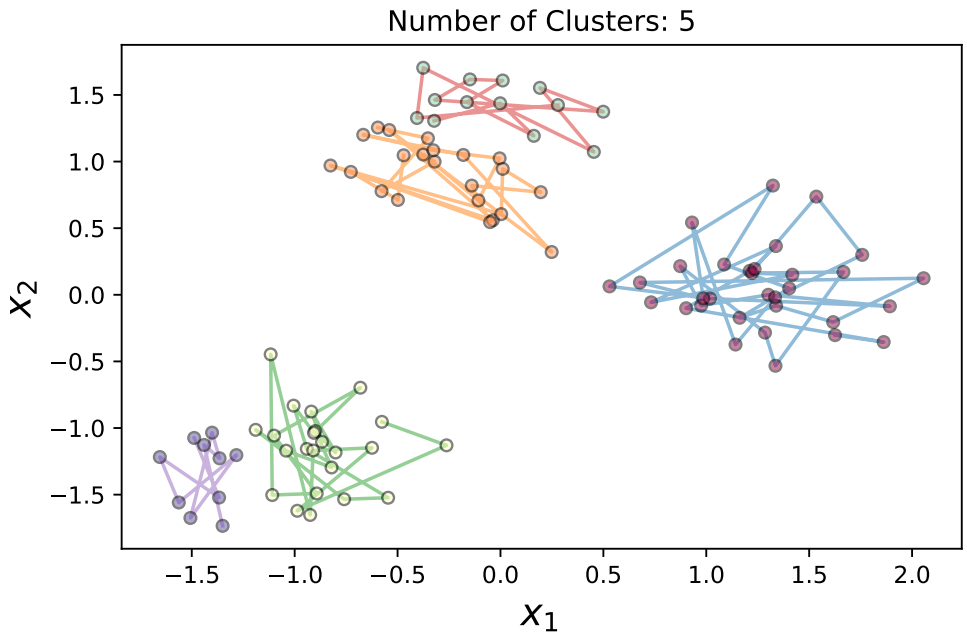
HAC

Example


Number of Clusters: 6
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

HAC

Example


Number of Clusters: 5
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

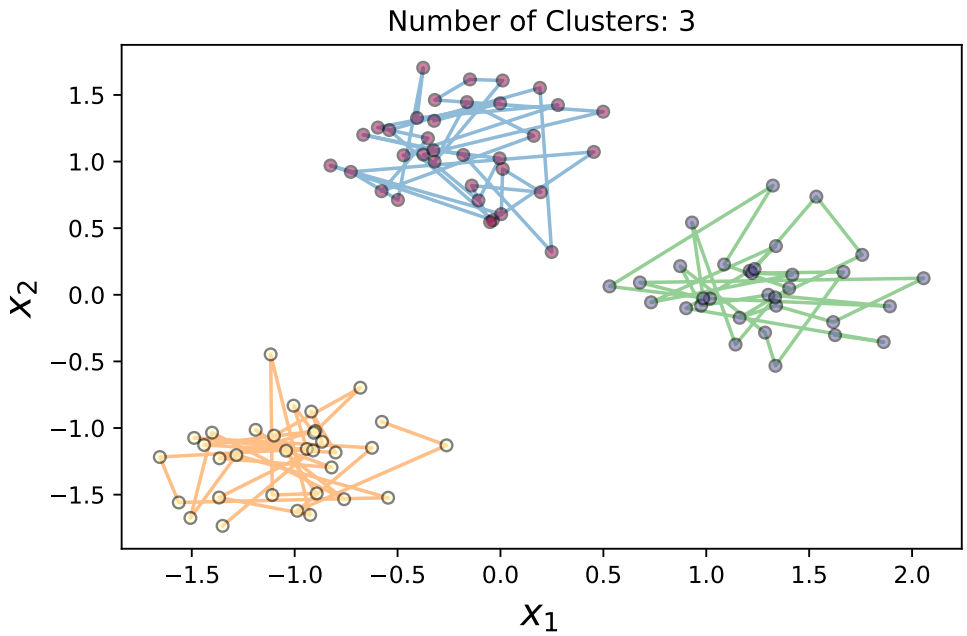
HAC

Example


Number of Clusters: 4
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

HAC

Example


Number of Clusters: 3
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

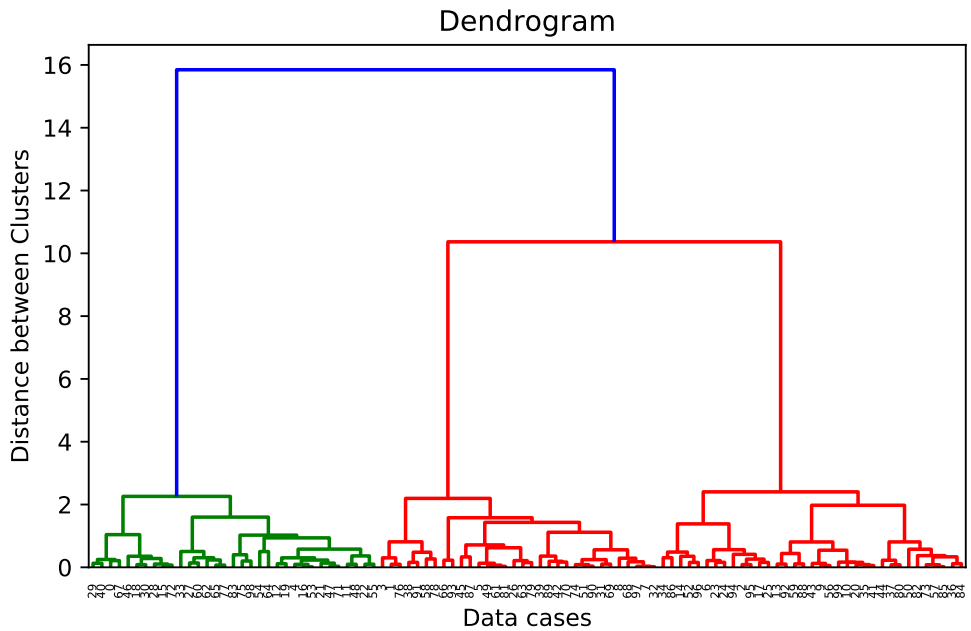
HAC

Example


Number of Clusters: 2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x_1

HAC

Dendrogram



Hierarchical Agglomerative Clustering

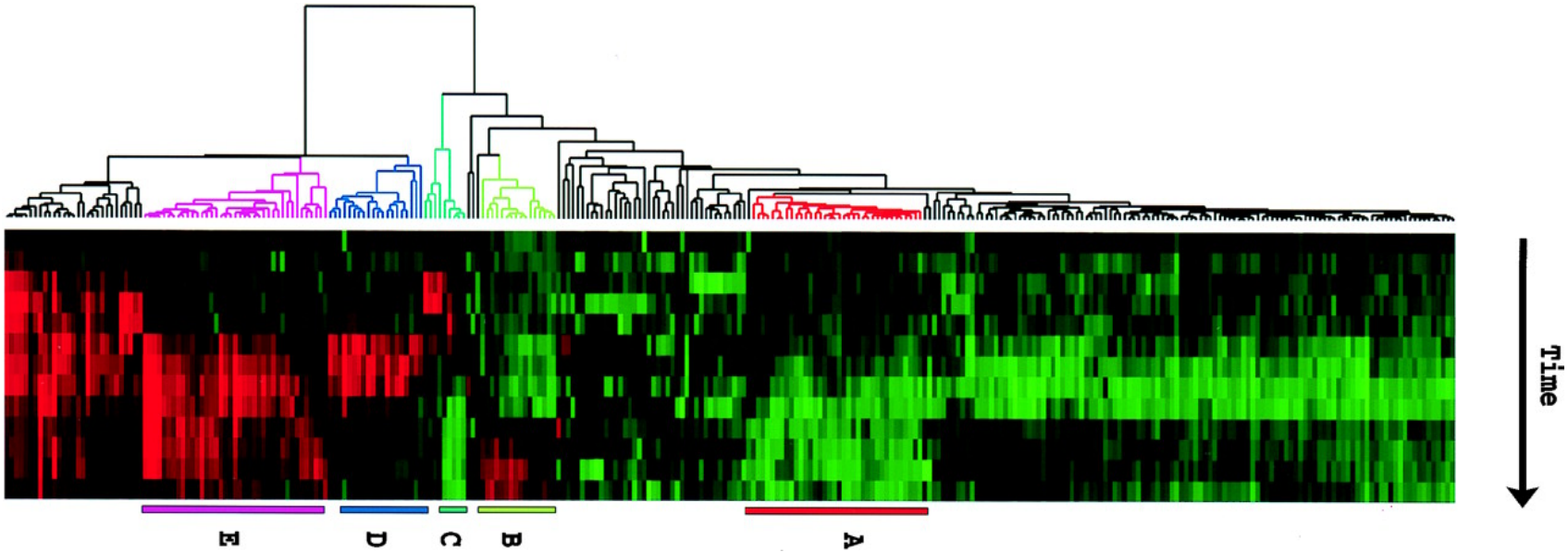
Issues

1. (like K-Means) Need good notion of similarity between clusters
2. Choose good 'linkage' function
3. (like K-Means) Sensitive to data scaling
4. Caution when interpreting results!

Clustering Examples

Gene Expression

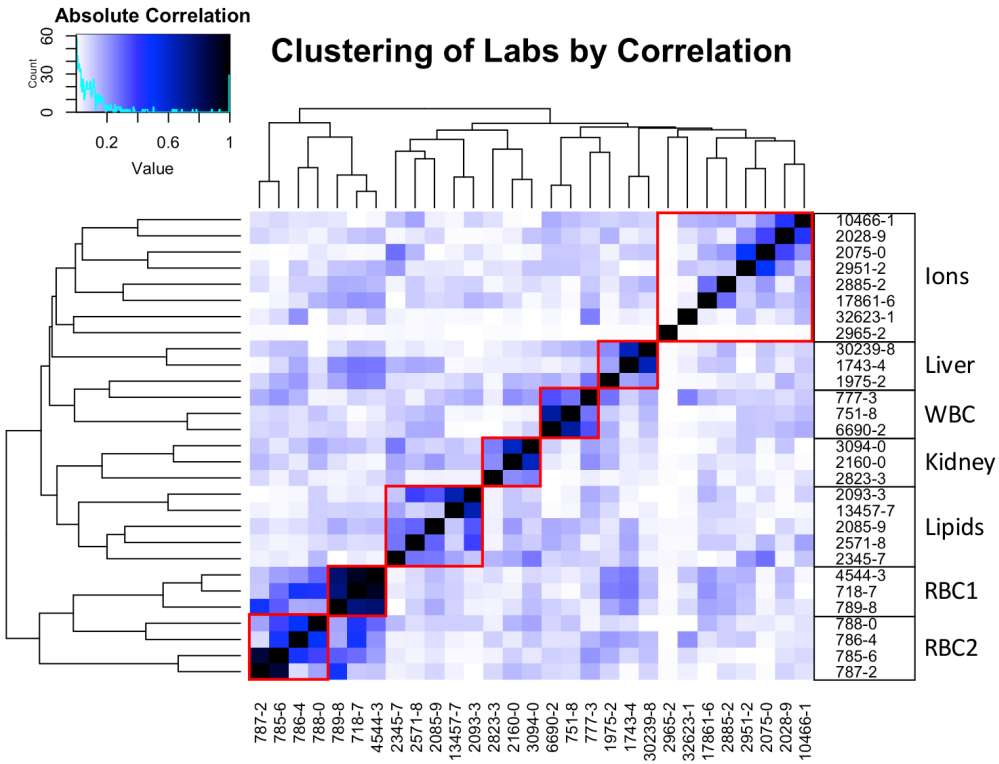
HAC used in gene expression analysis to visualize how genes group



Clustering Examples

Clinical Lab Measures

HAC and K-Means visualize lab measures in Geisinger EHR data (Bauer et. al. 2016)

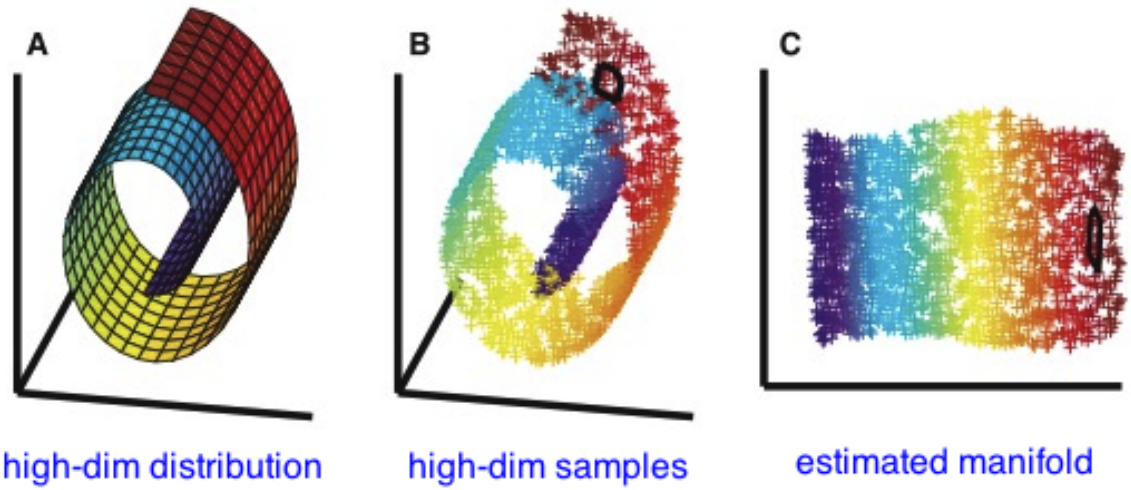


Unsupervised Learning

- Set of data: $\{\mathbf{x_i}, i = 1 \ \ldots \ N\}$ with d features

Definition (Dimensionality Reduction)

Given a set of data $\mathbf{x} \in \mathbb{R}^d$, map the feature vectors into a lower dimensional space $\mathbb{R}^k$ where $k < d$ while preserving certain properties of the data.



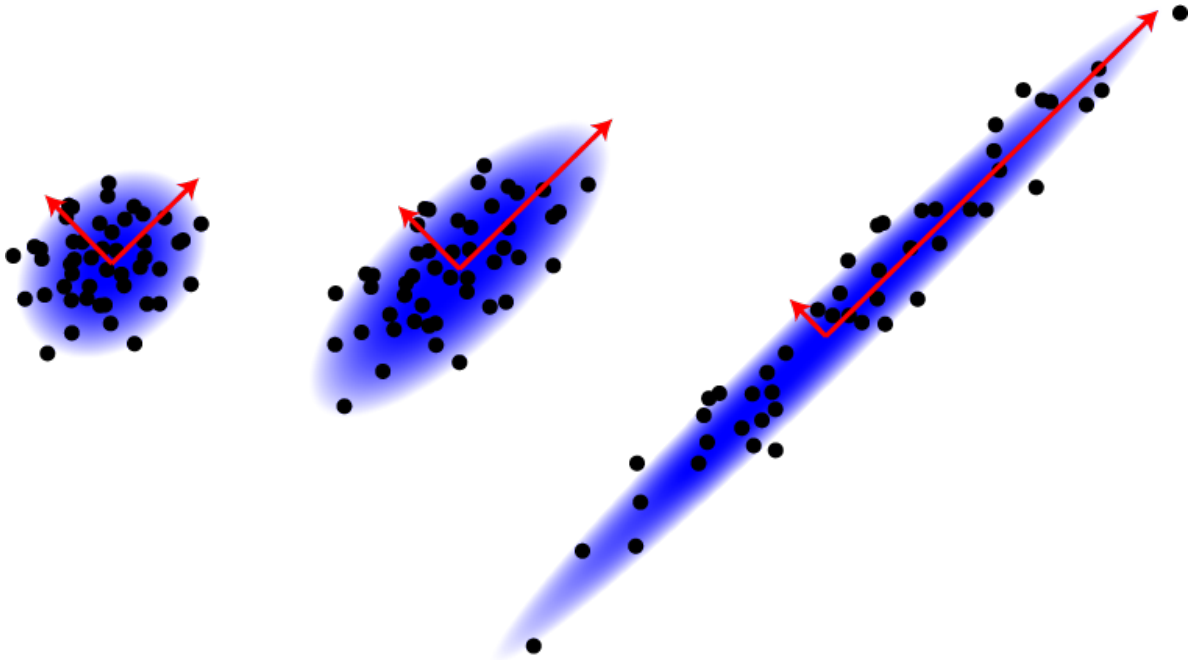
Principal Component Analysis

1. PCA assumes $\mathbf{x}_i \in \mathbb{R}^d$ lies on a k-dimensional linear manifold within $\mathbb{R}^d$.
2. In math,

$$\mathbf{X} = \mathbf{Z} \times \mathbf{B}$$

Principal Component Analysis

PCA identifies directions of maximum variance in the data.



Principal Component Analysis

Steps:

1. Given centered $N \times d$ data matrix $\mathbf{X}$

Principal Component Analysis

Steps:

1. Given centered $N \times d$ data matrix $\mathbf{X}$
2. Compute covariance matrix $\Sigma = \mathbf{X}^T\mathbf{X}$

Principal Component Analysis

Steps:

1. Given centered $N \times d$ data matrix $\mathbf{X}$
2. Compute covariance matrix $\Sigma = \mathbf{X}^T\mathbf{X}$
3. Compute the k leading eigenvectors of Σ, $w_1 \ldots w_k$

Principal Component Analysis

Steps:

1. Given centered $N \times d$ data matrix $\mathbf{X}$
2. Compute covariance matrix $\Sigma = \mathbf{X}^T\mathbf{X}$
3. Compute the k leading eigenvectors of Σ, $w_1 \ldots w_k$
4. Stack the eigenvectors into a $d \times k$ matrix $\mathbf{W}$ where each column is an eigenvector

Principal Component Analysis

Steps:

1. Given centered $N \times d$ data matrix $\mathbf{X}$
2. Compute covariance matrix $\Sigma = \mathbf{X}^T\mathbf{X}$
3. Compute the k leading eigenvectors of Σ, $w_1 \ldots w_k$
4. Stack the eigenvectors into a $d \times k$ matrix $\mathbf{W}$ where each column is an eigenvector
5. Compute the k-dimensional projection $\mathbf{Z} = \mathbf{XW}$

Principal Component Analysis

Steps:

1. Given centered $N \times d$ data matrix $\mathbf{X}$
2. Compute covariance matrix $\Sigma = \mathbf{X}^T\mathbf{X}$
3. Compute the k leading eigenvectors of Σ, $w_1 \ldots w_k$
4. Stack the eigenvectors into a $d \times k$ matrix $\mathbf{W}$ where each column is an eigenvector
5. Compute the k-dimensional projection $\mathbf{Z} = \mathbf{XW}$
6. To reconstruct $\mathbf{X}$ from $\mathbf{Z}$ and $\mathbf{W}$, compute $\hat{\mathbf{X}} = \mathbf{ZW}^T$

Principal Component Analysis

Why does this work?

1. Insight: Any real, symmetric matrix (like $\Sigma = \mathbf{X}^T\mathbf{X}$) can be decomposed into *eigenvectors* with corresponding *eigenvalues*

$$\Sigma = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

Principal Component Analysis

Why does this work?

1. Insight: Any real, symmetric matrix (like $\Sigma = \mathbf{X}^T\mathbf{X}$) can be decomposed into *eigenvectors* with corresponding *eigenvalues*

$$\Sigma = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

2. The maximum direction of variance in $\mathbf{X}$ is the eigenvector of $\mathbf{X}^T\mathbf{X}$ with the largest eigenvalue.

Principal Component Analysis

Why does this work?

1. Insight: Any real, symmetric matrix (like $\Sigma = \mathbf{X}^T\mathbf{X}$) can be decomposed into *eigenvectors* with corresponding *eigenvalues*

$$\Sigma = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

2. The maximum direction of variance in $\mathbf{X}$ is the eigenvector of $\mathbf{X}^T\mathbf{X}$ with the largest eigenvalue.
3. The k biggest directions of variance in $\mathbf{X}$ are the eigenvectors of $\mathbf{X}^T\mathbf{X}$ with the k largest eigenvalues.

Principal Component Analysis

Why does this work?

1. Insight: Any real, symmetric matrix (like $\Sigma = \mathbf{X}^T\mathbf{X}$) can be decomposed into *eigenvectors* with corresponding *eigenvalues*

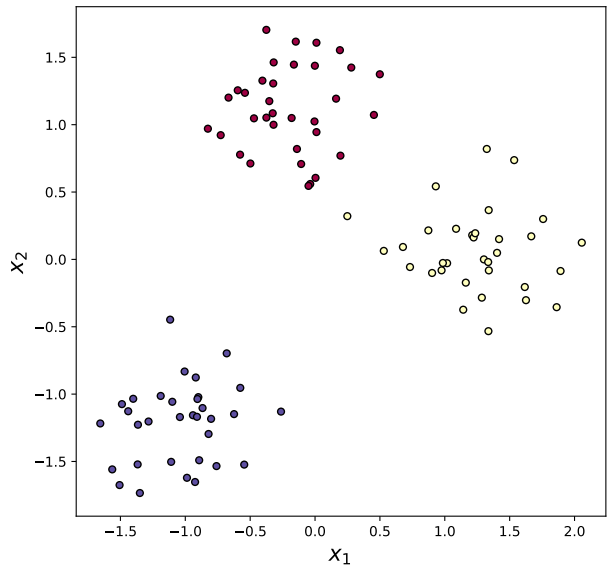
$$\Sigma = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

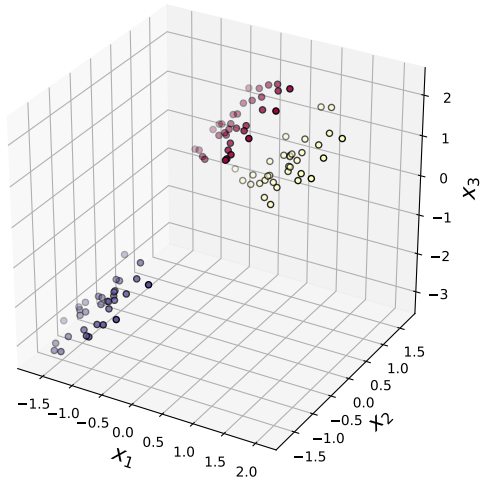
2. The maximum direction of variance in $\mathbf{X}$ is the eigenvector of $\mathbf{X}^T\mathbf{X}$ with the largest eigenvalue.
3. The k biggest directions of variance in $\mathbf{X}$ are the eigenvectors of $\mathbf{X}^T\mathbf{X}$ with the k largest eigenvalues.
4. eigenvectors are orthogonal to each other, so the data projected into $\mathbf{Z}$ will be linearly independent.

Principal Component Analysis

Example

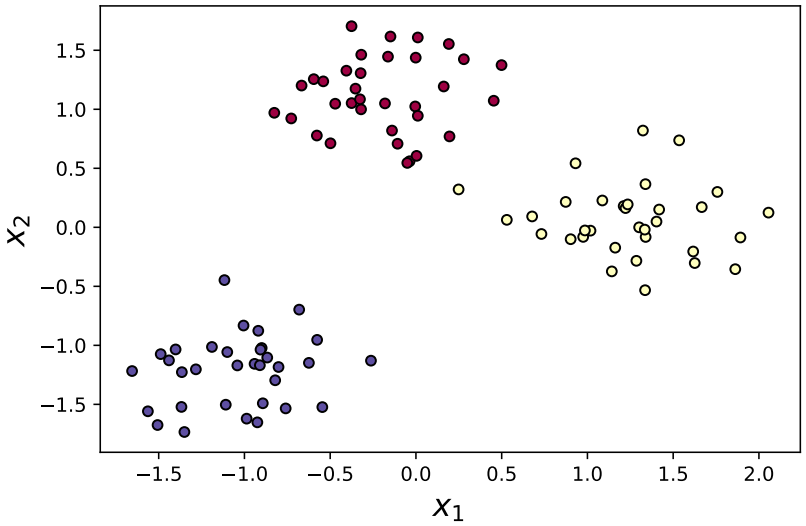
Our cluster data has an extra dimension ($x_3 = x_1 + x_2$)

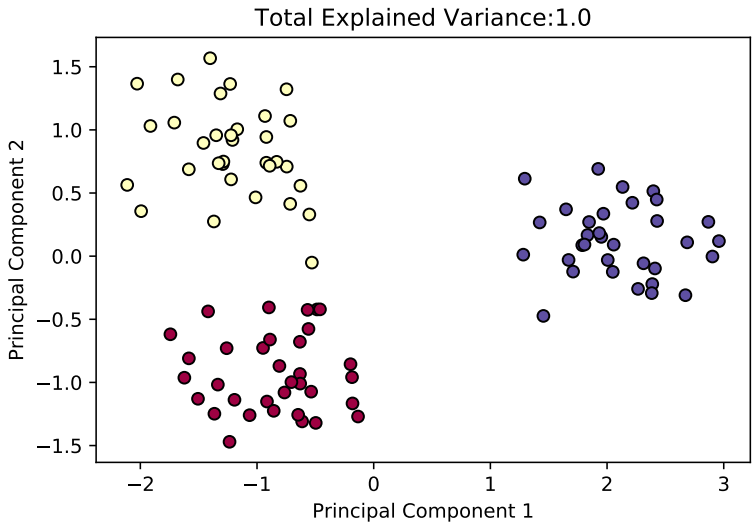




Principal Component Analysis

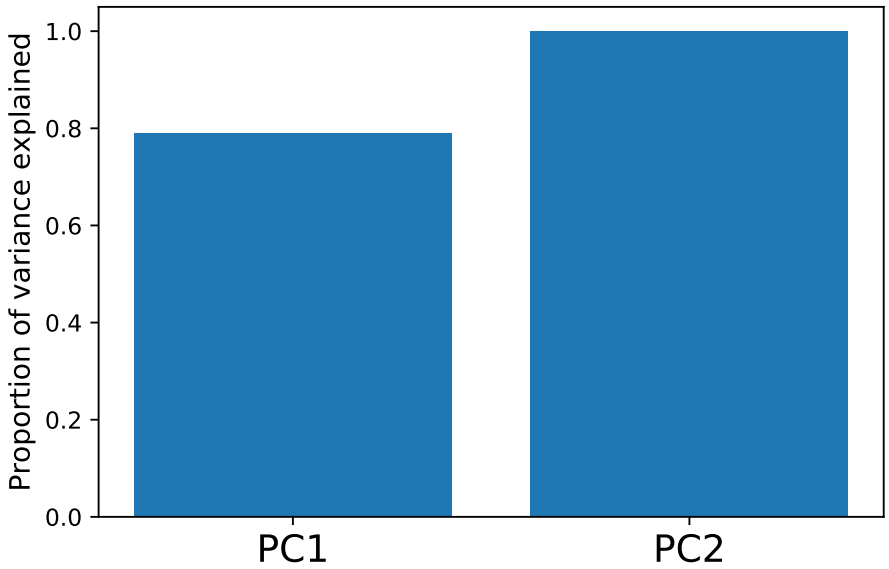
Example





Principal Component Analysis

Example



Principal Component Analysis

Uses

1. Each principal component is a linear combination of columns in $\mathbf{X}$.

Principal Component Analysis

Uses

1. Each principal component is a linear combination of columns in $\mathbf{X}$.
2. PCA is useful for visualization, because it can plot high-dimensional data along its first two directions of maximum variance.

Principal Component Analysis

Uses

1. Each principal component is a linear combination of columns in $\mathbf{X}$.
2. PCA is useful for visualization, because it can plot high-dimensional data along its first two directions of maximum variance.
3. Data should be centered and scaled to unit-variance.

Principal Component Analysis

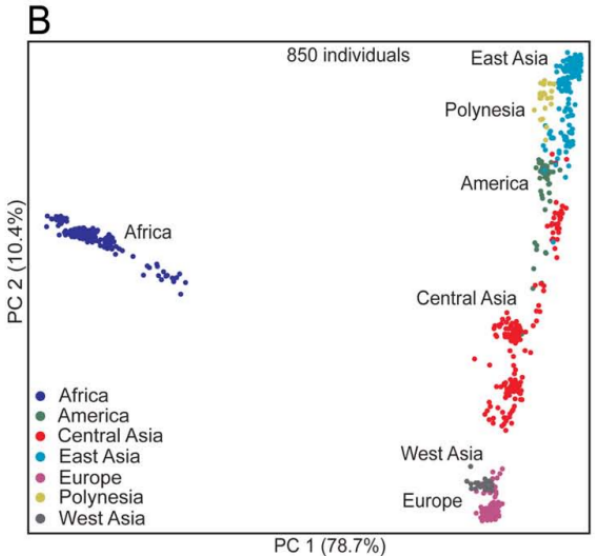
Uses

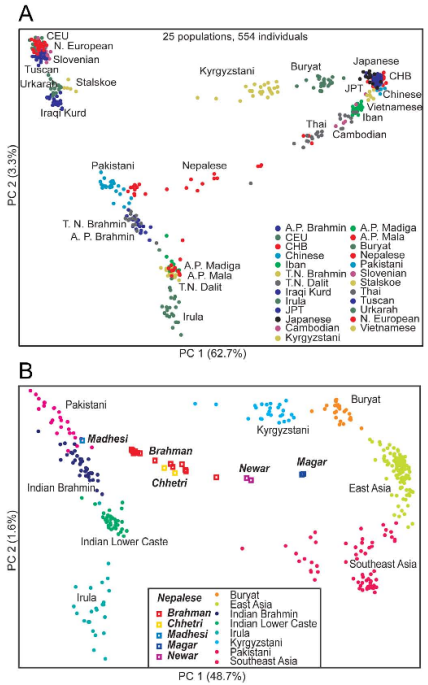
1. Each principal component is a linear combination of columns in $\mathbf{X}$.
2. PCA is useful for visualization, because it can plot high-dimensional data along its first two directions of maximum variance.
3. Data should be centered and scaled to unit-variance.
4. "Variance explained" gives a measure of how well the data dimensionality can be reduced.

PCA Examples

Genomes

Large-scale variation in human genomes corresponds to the history of our species (Xing et. al. 2010)





Non-linear manifolds

Problem

1. What if the low-dimensional manifold is not linear?

Non-linear manifolds

Problem

1. What if the low-dimensional manifold is not linear?
2. Variance in original coordinate system will not capture this.

Non-linear manifolds

Problem

1. What if the low-dimensional manifold is not linear?
2. Variance in original coordinate system will not capture this.
3. How can we preserve information in the data without relying on the original coordinates?

Non-linear manifolds

Solution

- Preserve the relations of samples in original data in the low-dimensional space.
  - Multidimensional Scaling (MDS)
  - Locally linear embedding
  - t-Distributed Stochastic Neighbor Embedding

t-SNE

- Convert distances between samples $\mathbf{x}_i, \mathbf{x}_j$ into conditional probabilities that represent similarities

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma^2)}{\exp(\sum_{k \neq i} ||x_i - x_k||^2/2\sigma^2)}$$

t-SNE

- Measure similarities in mapped points $\mathbf{z}_i, \mathbf{z}_j$ using a similar approach

$$q_{j|i} = \frac{(1 + ||z_i - z_j||^2)^{-1}}{\sum_{k \neq i}(||z_i - z_k||^2)^{-1}}$$

(this assumes a Student t-distribution, hence the name)

t-SNE

- Loss: Kullback-Leibler divergence

$$C = KL(P|Q) = \sum_i \sum_j p_{ij} \log(\frac{p_{ij}}{q_{ij}})$$

(similar to cross-entropy of the distributions)

t-SNE

Putting it all together:

- compute pairwise affinities $p_{i|j}$

t-SNE

Putting it all together:

- compute pairwise affinities $p_{i|j}$
- set $p_{ij} = (p_{i|j} + p_{j|i})/2$

t-SNE

Putting it all together:

- compute pairwise affinities $p_{i|j}$
- set $p_{ij} = (p_{i|j} + p_{j|i})/2$
- choose initial $\mathcal{Z}^0 = \{\mathbf{z}_i \ \ldots \ \mathbf{z}_N\}$

t-SNE

Putting it all together:

- compute pairwise affinities $p_{i|j}$
- set $p_{ij} = (p_{i|j} + p_{j|i})/2$
- choose initial $\mathcal{Z}^0 = \{\mathbf{z}_i \ \ldots \ \mathbf{z}_N\}$
- for T iterations:

t-SNE

Putting it all together:

- compute pairwise affinities $p_{i|j}$
- set $p_{ij} = (p_{i|j} + p_{j|i})/2$
- choose initial $\mathcal{Z}^0 = \{\mathbf{z}_i \ \ldots \ \mathbf{z}_N\}$
- for T iterations:
  - compute low-dimensional affinities q_{ij}

t-SNE

Putting it all together:

- compute pairwise affinities $p_{i|j}$
- set $p_{ij} = (p_{i|j} + p_{j|i})/2$
- choose initial $\mathcal{Z}^0 = \{\mathbf{z}_i \ \ldots \ \mathbf{z}_N\}$
- for T iterations:
  - compute low-dimensional affinities q_{ij}
  - compute gradient $\delta\mathcal{C}/\delta\mathcal{Z}$

t-SNE

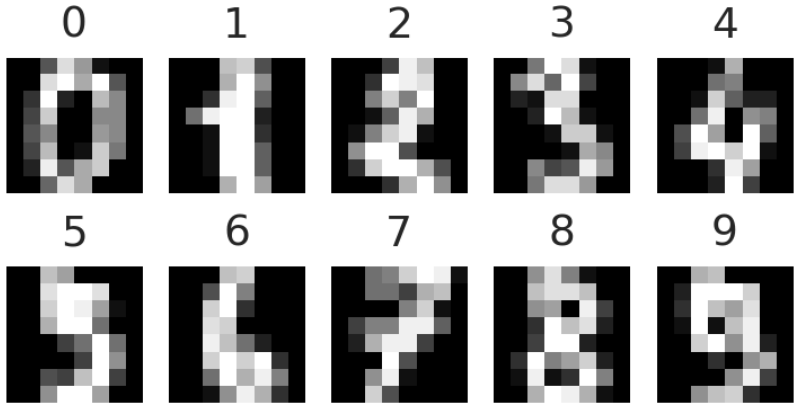
Putting it all together:

- compute pairwise affinities $p_{i|j}$
- set $p_{ij} = (p_{i|j} + p_{j|i})/2$
- choose initial $\mathcal{Z}^0 = \{\mathbf{z}_i \ \ldots \ \mathbf{z}_N\}$
- for T iterations:
  - compute low-dimensional affinities q_{ij}
  - compute gradient $\delta\mathcal{C}/\delta\mathcal{Z}$
  - set $\mathcal{Z}^{(t)} = \mathcal{Z}^{(t-1)} + \nu\delta\mathcal{C}/\delta\mathcal{Z} + \alpha(\mathcal{Z}^{(t-1)} - \mathcal{Z}^{(t-1)})$

t-SNE

Example

Hand-written digits

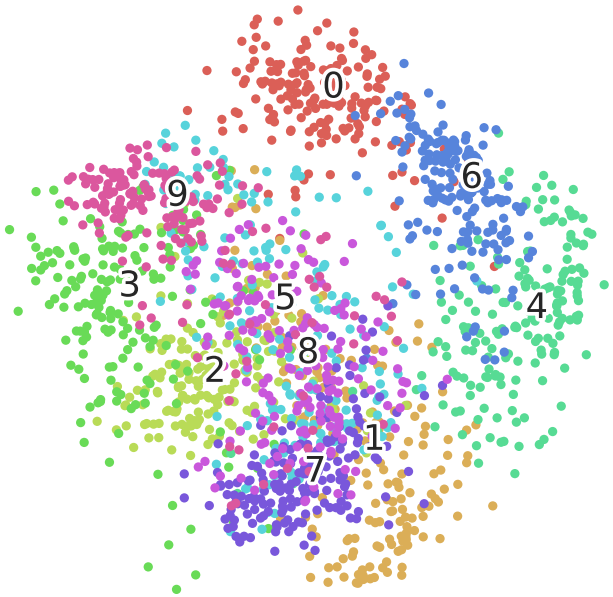


- 8x8 images, 64 attributes

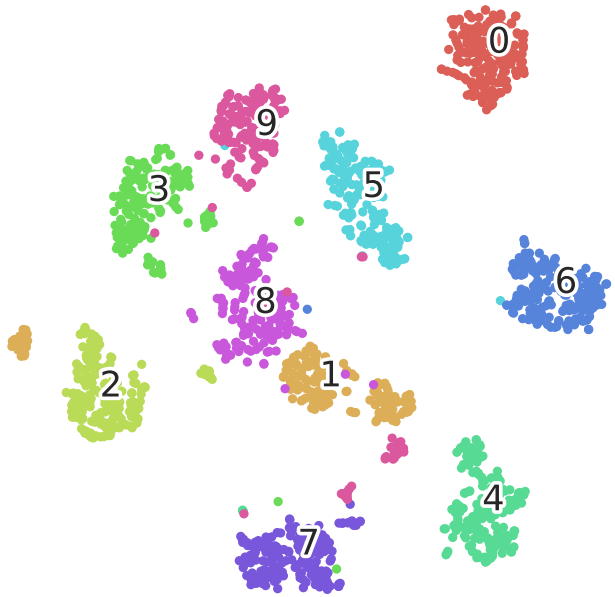
t-SNE

Example

PCA



t-SNE



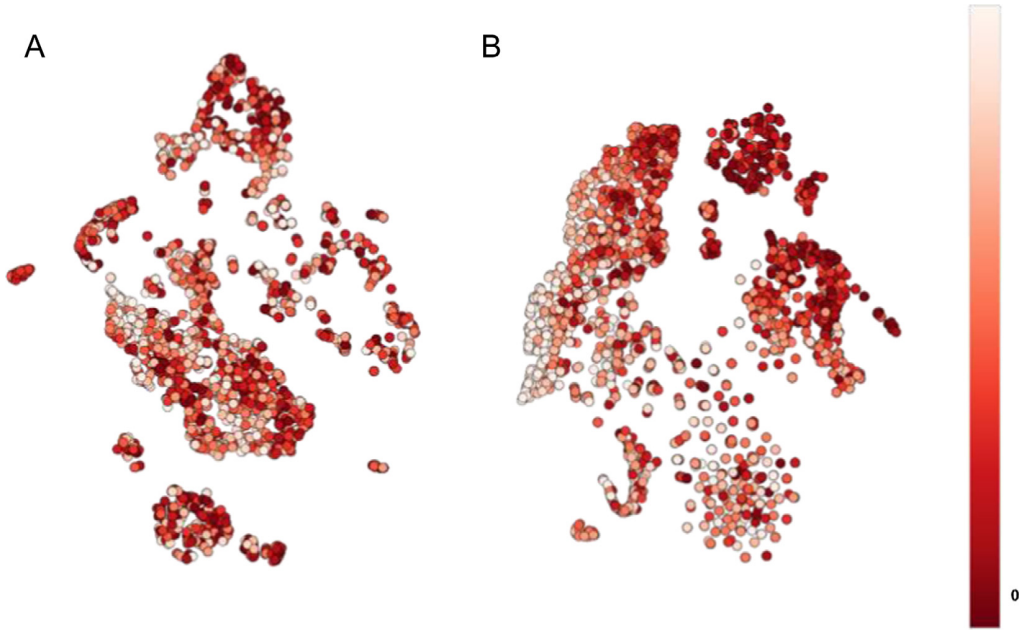
t-SNE

Animation

Examples

ALS

Learning low-order representations can improve supervised learning methods (Beaulieu-Jones et. al. 2016)



A: PCA followed by t-SNE. B: t-SNE of a 250 node auto-encoder. Color: days survived.

t-SNE

Conclusions

- t-SNE can help visualize high-dimensional data that lies on a low-dimensional, non-linear manifold.
- the result is a set of points
- good for visualization, but otherwise uninterpretable

Conclusions

- Other unsupervised learning algorithms
  - Mixture Models
  - Multidimensional Scaling (MDS)
  - Non-negative Matrix Factorization (NMF)
  - Auto-encoders (Deep Learning)
- mail me for lecture examples or questions
  `lacava@upenn.edu`