2109567

slido

Join at slido.com
#2109567

Click **Present with Slido** or install our Chrome extension to display joining instructions for participants while presenting.

CC BY NC SA

Final Thought - Data 100 Course Evaluation

2109567

- The final course survey is intended to help us improve future offerings of the course.

- If you complete our **internal anonymous survey** **AND** **the official Course Evaluation** (http://course-evaluations.berkeley.edu) **you will get 0.5% extra credit points** to your grade. If **90%** of the class also submits both, then everyone who **submitted both** will receive **an additional 0.5% percentage point** on their final grade.

- Internal course survey link: **https://tinyurl.com/sp25finalsurvey**
- Internal course survey QR code:

CC BY NC SA

2109567

LECTURE 26

Clustering

Using clustering algorithms to identify patterns in our data.

Data 100/Data 200, Spring 2025 @ UC Berkeley

Narges Norouzi and Josh Grossman

Content credit: Acknowledgments

CC BY NC SA

2109567

Today's Roadmap

Lecture 26, Data 100 Spring 2025

Introduction to Clustering

Taxonomy of Clustering Approaches

K-Means Clustering

Minimizing Inertia

Hierarchical Agglomerative Clustering

Picking K

CC BY NC SA

Review: Taxonomy of Machine Learning

2109567

Labeled Data

Unlabeled Data

Supervised Learning

Quantitative Response

Categorical Response

Regression

Classification

Data 8: Nearest Neighbors
Earlier: Logistic Regression

Unsupervised Learning

Dimensionality Reduction

Last week: Principal Component Analysis

Clustering

Today: **KMeans Clustering**
Agglomerative Clustering

CC BY NC SA

Review: Taxonomy of Machine Learning

2109567

Labeled Data

Supervised Learning

Quantitative Response

Categorical Response

Regression

Classification

In “Supervised Learning”:

- Goal is to create a function that maps inputs to outputs.
- Model is learned from example input/output pairs. Each pair consists of:
  - Input vector.
  - Output value (**label**).
- Regression: Output value is quantitative.
- Classification: Output value is categorical.

CC BY NC SA

Review: Taxonomy of Machine Learning

2109567

In “Unsupervised Learning”:

- Goal is to identify patterns in **unlabeled** data.
  - We do not have input/output pairs.
  - Note: Sometimes we may have labels, but we’re just choosing to ignore them (e.g. PCA on labeled data).

Unlabeled Data

Unsupervised Learning

Dimensionality Reduction

Clustering

Problem: Dimensionality Reduction.

- Solution: PCA.

Problem: Clustering.

CC BY NC SA

Image Compression with Clustering

2109567

- Why image Compression?
  - Digital images consume significant storage and bandwidth.
  - Reducing the number of colors simplifies images while retaining their visual appeal.

- Clustering can group similar colors (pixels) in an image into clusters.
- Each cluster represents a color centroid (mean color of the group).
- Replace each pixel in the image with the color of its cluster centroid, reducing the number of unique colors.

- Other image compression techniques:
  - Transform-based
  - Prediction-based
  - Deep Learning-based

CC BY NC SA

Clustering in Social Networks

2109567

In social network clustering, we identify groups (clusters or communities) of individuals who interact more frequently with each other.

Why Is It Useful?

- Helps analyze network behavior, predict relationships, and recommend connections.
- Applications:
  - Detecting communities on social network platforms.
  - Understanding collaboration in organizations.
  - Optimizing targeted marketing strategies.

Shown here is a synthetic graph representing interactions in a karate club.

- Nodes = Members of the club.
- Edges = Interactions between members.

CC BY NC SA

Clustering in Climate Science

2109567

Clustering in climate science helps

- Identify patterns in large, complex datasets.
- Provide insights into global and regional climate trends.

Examples:

- Climate zone identification
- Weather pattern analysis
- Environmental monitoring

In this example, we grouped regions based on average temperatures to understand climate zones.

CC BY NC SA

2109567

Taxonomy of Clustering Approaches

Lecture 26, Data 100 Spring 2025

Introduction to Clustering

Taxonomy of Clustering Approaches

K-Means Clustering

Minimizing Inertia

Hierarchical Agglomerative Clustering

Picking K

CC BY NC SA

Taxonomy of Clustering Approaches

2109567

BY NC SA

Taxonomy of Clustering Approaches

2109567

CC BY NC SA

2109567

K-Means Clustering

Lecture 26, Data 100 Spring 2025

Introduction to Clustering

Taxonomy of Clustering Approaches

K-Means Clustering

Minimizing Inertia

Hierarchical Agglomerative Clustering

Picking K

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - Move center for each color to center of points with that color.

Raw data →

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - Move center for each color to center of points with that color.

Iteration: 0

Initial random placement of two centers

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center**.
  - Move center for each color to center of points with that color.

Iteration: 0

Data colored by closest center

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary k, and randomly place k “centers”, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Iteration: 0

Where should the centers go next?

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Iteration: 1

Centers moved to their new homes

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center**.
  - Move center for each color to center of points with that color.

Iteration: 1

Data colored by closest center (in new position)

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Iteration: 2

Centers moved to new position

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center**.
  - Move center for each color to center of points with that color.

Iteration: 3

Data colored by closest center (in new position)

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Iteration: 4

Centers moved to new position

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center**.
  - Move center for each color to center of points with that color.

Iteration: 4

Data colored by closest center (in new position)

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - Color points according to the closest center.
  - **Move center for each color to center of points with that color.**

Iteration: 5

Centers moved to new position

CC BY NC SA

K-Means Clustering

2109567

Most popular clustering approach. The algorithm:

- Pick an arbitrary **k**, and randomly place **k "centers"**, each a different color.
- Repeat until convergence:
  - **Color points according to the closest center**.
  - Move center for each color to center of points with that color.

Iteration: 5

Data colored by closest center (in new position)

CC BY NC SA

K-Means Clustering

2109567

Above, we see the results after iteration 4 and 5:

- Centers moved slightly between iteration 4 and 5.
- But no points changed color.
- Are we done?

CC BY NC SA

K-Means Clustering

2109567

Above, we see the results after iteration 4 and 5:

- Centers moved slightly between iteration 4 and 5.
- But no points changed color.
- Are we done?
  - Yes! If we tried iteration 6, we'd see that centers don't move at all.

CC BY NC SA

K-Means vs. K-Nearest Neighbors

2109567

Quick note: **K-Means** is a totally different algorithm than "**K-Nearest Neighbors**".

- **K-Means**: For clustering:
  - Assigns each point to one of K clusters.
- **K-Nearest Neighbors**: For Classification (or less often, Regression):
  - Prediction is the most common class among the k-nearest data points in the training set.
  - Covered in Data 8.

The names may be similar, but there isn't really anything in common.

CC BY NC SA

2109567

Minimizing Inertia

Lecture 26, Data 100 Spring 2025

Introduction to Clustering

Taxonomy of Clustering Approaches

K-Means Clustering

Minimizing Inertia

Hierarchical Agglomerative Clustering

Picking K

CC BY NC SA

K-Means Clustering for K = 4

2109567

Below is an example of an output for K=4:

CC BY NC SA

K-Means Clustering for K = 4

2109567

Each time you run K-Means, you get a different output, depending on where centers started.

`random.seed(25)`

`random.seed(29)`

`random.seed(40)`

Which is best?

CC BY NC SA

slido

2109567

With K = 4, which of the following options shows a better clustering output?

ⓘ Start presenting to display the poll results on this slide.

CC BY NC SA

K-Means Clustering for K = 4

2109567

Each time you run K-Means, you get a different output, depending on where centers started.

`random.seed(25)`

`random.seed(29)`

`random.seed(40)`

Which is best?

- One approach: Define some sort of loss function.

CC BY NC SA

K-Means Clustering for K = 4

2109567

Each time you run K-Means, you get a different output, depending on where centers started.

`random.seed(25)`

`random.seed(29)`

`random.seed(40)`

Which is best?

- One approach: Define some sort of loss function.

Goal: Come up with a loss function for clustering.

- What is “good” about the leftmost clustering, but “bad” about the right clustering?

CC BY NC SA

slido

2109567

What are some good evaluation metrics to compare different clustering results?

ⓘ Start presenting to display the poll results on this slide.

CC BY NC SA

K-Means Clustering for K = 4

2109567

To evaluate different clustering results, we need a loss function.

Two common loss functions:

- **Inertia**: Sum of squared distances from each data point to its center.
- **Distortion**: Weighted sum of squared distances from each data point to its center.

Example:

- Inertia: $0.47^2 + 0.19^2 + 0.34^2 + 0.25^2 + 0.58^2 + 0.36^2 + 0.44^2$
- Distortion: $(0.47^2 + 0.19^2 + 0.34^2)/3 + (0.25^2 + 0.58^2 + 0.36^2 + 0.44^2)/4$

CC BY NC SA

K-Means Clustering for K = 4

2109567

Each time you run K-Means, you get a different output:

`random.seed(25)`

`random.seed(29)`

`random.seed(40)`

Among these three choices, our **inertia** loss function says that the leftmost clustering is best (inertia: 44.88) and rightmost clustering (inertia: 54.27) is worst.

CC BY NC SA

K-Means and Inertia

2109567

It turns out that the function K-Means is trying to minimize is **inertia**…

… but often fails to find global optimum. Why?

Can think of K-means as a pair of optimizers that take turns:

- First optimizer:
  - Holds center positions constant.
  - Optimizes data colors.
- Second optimizer:
  - Holds data colors constant.
  - Optimizes center positions.
- Neither gets total control!

CC BY NC SA

Optimizing Inertia

2109567

Hard problem: Give an algorithm that optimizes inertia FOR A GIVEN K. K is picked in advance.

- Your algorithm should return the EXACT best centers and colors.
- Don't worry about runtime.

This is a bit of a CS61B/CS70/CS170 problem...

CC BY NC SA

Optimizing Inertia

2109567

Hard problem: Give an algorithm that optimizes inertia FOR A GIVEN K. K is picked in advance.

- Your algorithm should return the EXACT best centers and colors.
- Don't worry about runtime.

A "coloring" is just a choice of color for every point, e.g. point 1 = red, point 2 = green, point 3 = orange, point 4 = blue

Algorithm:

- For all possible k^n **colorings**:
  - Compute the k centers for that coloring.
  - Compute the inertia for the k centers.
    - If current inertia is better than best known, write down the current centers and coloring and call that the new best known.

No better algorithm has been found for solving the problem of minimizing exactly.

For those who know what this means:
K-Means is known to be an NP-hard problem

CC BY NC SA

K-Means

2109567

Which clustering result do you like better?

CC BY NC SA

slido

2109567

Which clustering result do you like better?

ⓘ Start presenting to display the poll results on this slide.

CC BY NC SA

K-Means

2109567

Which clustering result do you like better?

K-Means likes the one on the right better because it has lower inertia: sum of squared distances from each data point to its center.

- Why is the inertia on the right lower?
- Is clustering on the right “wrong”?

CC BY NC SA

K-Means

2109567

Which clustering result do you like better?

K-Means likes the one on the right better because it has lower inertia: sum of squared distances from each data point to its center.

- Why is the inertia on the right lower? K-Means optimizes for distance, not “blobbiness”.
- Is clustering on the right “wrong”? Good question!

CC BY NC SA

Interlude

Lecture 26, Data 100 Spring 2025

2109567

CC BY NC SA

2109567

Hierarchical Agglomerative Clustering

Lecture 26, Data 100 Spring 2025

Introduction to Clustering

Taxonomy of Clustering Approaches

K-Means Clustering

Minimizing Inertia

Hierarchical Agglomerative Clustering

Picking K

CC BY NC SA

Taxonomy of Clustering Approaches

2109567

CC BY NC SA

Hierarchical Agglomerative Clustering

2109567

- We start with every data point in a separate cluster.
- We keep merging the most similar pairs of data points/clusters until we have one big cluster left.
- This is called a **bottom-up** or **agglomerative** method.
- There are various ways to decide the order of combining clusters called **Linkage Criterion**.

Source: https://www.drive5.com/usearch/manual/agg.html

CC BY NC SA

Approaches for Merging Clusters: Linkage Criteria

2109567

- **Single linkage:** Similarity of the most similar: the distance between two clusters as the **minimum** distance between a point in the first cluster and a point in the second.

Single-Link Example

CC BY NC SA

Approaches for Merging Clusters: Linkage Criteria

2109567

- **Single linkage:** Similarity of the most similar: the distance between two clusters as the **minimum** distance between a point in the first cluster and a point in the second.
- **Average linkage:** **Average** similarity of pairs of points in clusters.
- **Complete linkage:** Similarity of the least similar: the distance between two clusters as the **maximum** distance between a point in the first cluster and a point in the second.

Single-Link Example

Complete-Link Example

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

When the algorithm starts, every data point is in its own cluster.

- Below, 12 data points, so 12 clusters.
- Closest clusters are 10 and 11, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

When the algorithm starts, every data point is in its own cluster.

- Below, 12 data points, so 12 clusters.
- Closest clusters are 10 and 11, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 0 and 4, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

At this point we have 10 clusters:

- 8 with a single point {1, 2, 3, 5, 6, 7, 8, 9}
- 2 with two points {0/0, 10/10}

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 1 and 5, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 7 and 9, so merge them.

- What is the distance between clusters 0 and 3? Since we use the Complete-Link approach, we should consider the **max** distance amongst pairs.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 6 and 8, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 0 and 3, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 1 and 10, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 0 and 7, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 0 and 6, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Next two closest are 0 and 2, so merge them.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

Which finally leads to having two clusters.

CC BY NC SA

Agglomerative Clustering - Complete Linkage

2109567

On the full dataset, our agglomerative clustering algorithm gets the “correct” output.

CC BY NC SA

Clustering and Dendrograms

2109567

Agglomerative clustering is one form of “hierarchical clustering.”

- Can keep track of when two clusters got merged.
  - Each cluster is a tree.
- Can visualize merging hierarchy, resulting in a “dendrogram.”
  - Won’t discuss any further, but you might see these in the wild.

CC BY NC SA

How to Use Dendrogram

2109567

- Observe the Vertical Gaps:
  - Look for the largest vertical distance (gap) between horizontal lines as you move upward in the dendrogram.
  - This gap indicates a significant jump in distance and suggests that merging clusters beyond this point combines dissimilar groups.
- Cut the Dendrogram:
  - Draw a horizontal line just below this large gap.
  - The number of vertical lines the horizontal line crosses represents the optimal K.

CC BY NC SA

2109567

Picking K

Lecture 26, Data 100 Spring 2025

Introduction to Clustering

Taxonomy of Clustering Approaches

K-Means Clustering

Minimizing Inertia

Hierarchical Agglomerative Clustering

Picking K

CC BY NC SA

Picking K

2109567

The algorithms we've discussed today require us to pick a K before we start.

- But how do we pick K?

Often, best K is subjective:

- Example: State plot.
- How many clusters are there here?

CC BY NC SA

Picking K: Elbow Method

2109567

For K-Means, one approach is to plot inertia versus many different K values.

- Pick the K in the “elbow”, where we get diminishing returns afterwards.
- Note: Big complicated data often lacks an elbow.

CC BY NC SA

Silhouette Scores

2109567

To evaluate how “well-clustered” a specific data point is, we can use the “silhouette score”, a.k.a. the “silhouette width”.

- **High score**: Near the other points in its cluster.
- **Low score**: Far from the other points in its cluster.

For a data point X, score S is:

- A = avg distance to other points in cluster.
- B = avg distance to points in closest cluster.
- S = (B - A) / max(A, B)

CC BY NC SA

slido

2109567

What is the highest value of S, given the information in the image?

ⓘ Start presenting to display the poll results on this slide.

CC BY NC SA

Silhouette Scores

2109567

For a data point X:

- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B).

What is the highest possible S?

- How can this happen?

Highest possible S is 1.

- This happens if every point in X's cluster is right on top of X.
  - Average distance to other points in X's cluster is 0 (all equal to zero!), so A = 0.
  - B / max(0, B) = B / B = 1
  - Or B >> A

CC BY NC SA

slido

2109567

Can the value of S be negative?

ⓘ Start presenting to display the poll results on this slide.

CC BY NC SA

Silhouette Scores

2109567

For a data point X:

- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B).

Can S be negative?

- Yes. If the average distance to X's clustermates is larger than distance to the closest cluster.

CC BY NC SA

Silhouette Scores

2109567

For a data point X:

- A = average distance to other points in X's cluster.
- B = average distance to points in closest cluster.
- S = (B - A) / max(A, B).

Can S be negative?

- Yes. If the average distance to X's clustermates is larger than distance to the closest cluster.
- Example: The "low score" point on the right has S = -0.13

CC BY NC SA

Evaluating Clustering

2109567

To evaluate different clustering results, we need a loss function.

We discussed:

- **Inertia**: Sum of squared distances from each data point to its center.
  - Lower inertia indicates tighter clusters.
  - Inertia always decreases as k increases, so it doesn't penalize overfitting.
  - Inertia doesn't consider whether clusters are well-separated or non-spherical.
  - Use the elbow method to find optimal number of clusters.
- **Silhouette Score**: Silhouette score evaluates how well a point fits within its cluster versus how poorly it fits into neighboring clusters.
  - A higher silhouette score indicates better-defined and separated clusters.
  - Useful when comparing clustering results for different values of k.
  - Negative silhouette scores can highlight misclassified points or overlapping clusters.
  - Assumes convex clusters and computationally expensive.

CC BY NC SA

Silhouette Plot

2109567

We can plot the **Silhouette Scores** for all of our data points.

- Points with large silhouette widths are deeply embedded in their cluster.
- Red dotted line shows the average.

CC BY NC SA

Silhouette Scores, K = 2 vs. K = 3

2109567

Average silhouette score is lower.

Picking K: Real World Metrics (Example via Andrew Ng)

2109567

Sometimes you can rely on real world metrics to guide your choice of K. Assume that you have a T-shirt production line and you want to decide about the different T-shirt sizes you need to produce.

Perform 2 clusterings:

- Cluster heights and weights of customers with K = 3 to design Small, Medium, and Large shirts.
- Cluster heights and weights of customers with K = 5 to design XS, S, M, L, and XL shirts.

To pick K:

- Consider projected costs and sales for the 2 different K's.
- Pick the one that maximizes profit.

CC BY NC SA

Summary

2109567

Today we discussed a new machine learning goal: Clustering.

Saw two solutions:

- **K-Means**
  - Tries to optimize a loss function called inertia -- no known algorithm to find the optimal answer in an efficient manner.
- **Hierarchical Agglomerative**

Our version of these algorithms required a hyperparameter k.

- 4 ways to pick k: Intuitively, elbow method, silhouette scores, harnessing real world metrics.

CC BY NC SA

2109567

LECTURE 26

Clustering

Data 100/Data 200, Spring 2025 @ UC Berkeley

Narges Norouzi and Josh Grossman

Content credit: Acknowledgments

CC BY NC SA