

COMS 4771 Lecture 21

1. k -means clustering
2. Dictionary learning

CLUSTERING

Unsupervised classification

- ▶ **Input:** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- ▶ **Output:** function $f: \mathbb{R}^d \rightarrow \{1, 2, \dots, k\} =: [k]$.
- ▶ **Typical semantics:** hidden subpopulation structure.

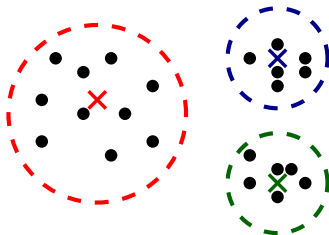
Unsupervised classification

- ▶ **Input:** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- ▶ **Output:** function $f: \mathbb{R}^d \rightarrow \{1, 2, \dots, k\} =: [k]$.
- ▶ **Typical semantics:** hidden subpopulation structure.

Clustering

- ▶ **Input:** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- ▶ **Output:** partitioning of $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ into k groups.
- ▶ Often done via unsupervised classification;
⇒ “clustering” often synonymous with “unsupervised classification”.
- ▶ Sometimes also have a “representative” $\mathbf{c}_j \in \mathbb{R}^d$ for each $j \in [k]$
(e.g., average of the $\mathbf{x}^{(i)}$ in j th group) → **quantization**.

UNSUPERVISED CLASSIFICATION / CLUSTERING



Clustering

- ▶ **Input:** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- ▶ **Output:** partitioning of $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ into k groups.
- ▶ Often done via unsupervised classification;
 $\Rightarrow$ “clustering” often synonymous with “unsupervised classification”.
- ▶ Sometimes also have a “representative” $\mathbf{c}_j \in \mathbb{R}^d$ for each $j \in [k]$
(e.g., average of the $\mathbf{x}^{(i)}$ in j th group) $\longrightarrow$ **quantization**.

USES OF CLUSTERING: FEATURE REPRESENTATIONS

“One-hot” / “dummy variable” encoding of $f(\mathbf{x})$

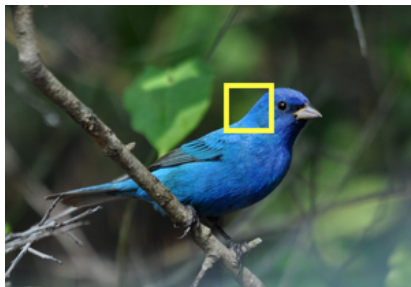
$$\phi(\mathbf{x}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \leftarrow f(\mathbf{x}) \text{ position}$$

(Often used together with other features.)

USES OF CLUSTERING: FEATURE REPRESENTATIONS

Histogram representation

- ▶ Cut up each $\mathbf{x}^{(i)} \in \mathbb{R}^d$ into different parts $\mathbf{x}^{(i,1)}, \mathbf{x}^{(i,2)}, \dots, \mathbf{x}^{(i,m)} \in \mathbb{R}^p$ (e.g., small patches of an image) .
- ▶ Cluster all the parts $\mathbf{x}^{(i,j)}$: get k representatives $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^p$.
- ▶ Represent $\mathbf{x}^{(i)}$ by a histogram over $\{1, 2, \dots, k\}$ based on assignments of $\mathbf{x}^{(i)}$'s parts to representatives.



USES OF CLUSTERING: COMPRESSION

Quantization

Replace each $\mathbf{x}^{(i)}$ with its representative

$$\mathbf{x}^{(i)} \mapsto \mathbf{c}_{f(\mathbf{x}^{(i)})}.$$

Example: quantization at image patch level.



k -MEANS CLUSTERING

k -MEANS CLUSTERING

Problem

- ▶ **Input:** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- ▶ **Output:** k representatives (“centers”, “means”) $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$.
- ▶ **Objective:** choose $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$ to minimize

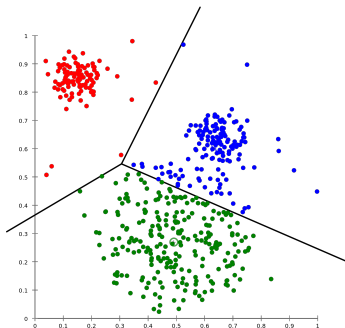
$$\sum_{i=1}^n \min_{j \in [k]} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2.$$

k -MEANS CLUSTERING

Problem

- ▶ **Input:** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- ▶ **Output:** k representatives (“centers”, “means”) $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$.
- ▶ **Objective:** choose $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^n \min_{j \in [k]} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2.$$



Natural assignment function

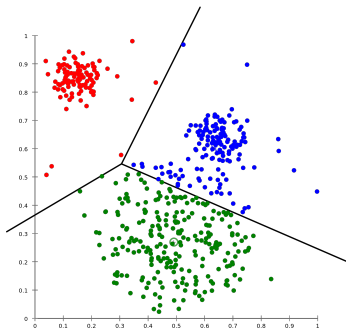
$$f(\mathbf{x}) := \arg \min_{j \in [k]} \|\mathbf{x} - \mathbf{c}_j\|_2^2.$$

k -MEANS CLUSTERING

Problem

- ▶ **Input:** $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- ▶ **Output:** k representatives (“centers”, “means”) $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$.
- ▶ **Objective:** choose $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^n \min_{j \in [k]} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2.$$



Natural assignment function

$$f(\mathbf{x}) := \arg \min_{j \in [k]} \|\mathbf{x} - \mathbf{c}_j\|_2^2.$$

NP-hard, even if $k = 2$ or $d = 2$.

THE EASY CASES

k -means clustering for $k = 1$

Problem: Pick $\mathbf{c} \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^n \|\mathbf{x}^{(i)} - \mathbf{c}\|_2^2.$$

THE EASY CASES

k -means clustering for $k = 1$

Problem: Pick $\mathbf{c} \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^n \|\mathbf{x}^{(i)} - \mathbf{c}\|_2^2.$$

Solution: “bias/variance decomposition”

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}^{(i)} - \mathbf{c}\|_2^2 = \|\boldsymbol{\mu} - \mathbf{c}\|_2^2 + \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}^{(i)} - \boldsymbol{\mu}\|_2^2$$

where $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}^{(i)}$.

THE EASY CASES

k -means clustering for $k = 1$

Problem: Pick $\mathbf{c} \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^n \|\mathbf{x}^{(i)} - \mathbf{c}\|_2^2.$$

Solution: “bias/variance decomposition”

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}^{(i)} - \mathbf{c}\|_2^2 = \|\boldsymbol{\mu} - \mathbf{c}\|_2^2 + \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}^{(i)} - \boldsymbol{\mu}\|_2^2$$

where $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}^{(i)}$.

Therefore, optimal choice for $\mathbf{c}$ is $\boldsymbol{\mu}$.

THE EASY CASES

k -means clustering for $k = 1$

Problem: Pick $\mathbf{c} \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^n \|\mathbf{x}^{(i)} - \mathbf{c}\|_2^2.$$

Solution: “bias/variance decomposition”

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}^{(i)} - \mathbf{c}\|_2^2 = \|\boldsymbol{\mu} - \mathbf{c}\|_2^2 + \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}^{(i)} - \boldsymbol{\mu}\|_2^2$$

where $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}^{(i)}$.

Therefore, optimal choice for $\mathbf{c}$ is $\boldsymbol{\mu}$.

k -means clustering for $d = 1$

Dynamic programming in time $O(n^2k)$.

ALTERNATING OPTIMIZATION ALGORITHM

Assignment variables

For each data point $\mathbf{x}^{(i)}$, let $\phi^{(i)} \in \{0, 1\}^d$ denote its “one-hot” representation:

$$\phi_j^{(i)} = \mathbb{1}\{\mathbf{x}^{(i)} \text{ is assigned to cluster } j\}.$$

ALTERNATING OPTIMIZATION ALGORITHM

Assignment variables

For each data point $\mathbf{x}^{(i)}$, let $\phi^{(i)} \in \{0, 1\}^d$ denote its “one-hot” representation:

$$\phi_j^{(i)} = \mathbb{1}\{\mathbf{x}^{(i)} \text{ is assigned to cluster } j\}.$$

Objective becomes (for optimal setting of $\phi^{(i)}$ s)

$$\sum_{i=1}^n \min_{j \in [k]} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 = \sum_{i=1}^n \left\{ \sum_{j=1}^k \phi_j^{(i)} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 \right\}.$$

ALTERNATING OPTIMIZATION ALGORITHM

Assignment variables

For each data point $\mathbf{x}^{(i)}$, let $\phi^{(i)} \in \{0, 1\}^d$ denote its “one-hot” representation:

$$\phi_j^{(i)} = \mathbb{1}\{\mathbf{x}^{(i)} \text{ is assigned to cluster } j\}.$$

Objective becomes (for optimal setting of $\phi^{(i)}$ s)

$$\sum_{i=1}^n \min_{j \in [k]} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 = \sum_{i=1}^n \left\{ \sum_{j=1}^k \phi_j^{(i)} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 \right\}.$$

Lloyd's algorithm (sometimes called *the* k -means algorithm)

Initialize $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$ somehow. Then repeat until convergence:

- ▶ Holding $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ fixed, pick optimal $\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(n)}$.
- ▶ Holding $\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(n)}$ fixed, pick optimal $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$.

ALTERNATING OPTIMIZATION ALGORITHM

Assignment variables

For each data point $\mathbf{x}^{(i)}$, let $\phi^{(i)} \in \{0, 1\}^d$ denote its “one-hot” representation:

$$\phi_j^{(i)} = \mathbb{1}\{\mathbf{x}^{(i)} \text{ is assigned to cluster } j\}.$$

Objective becomes (for optimal setting of $\phi^{(i)}$ s)

$$\sum_{i=1}^n \min_{j \in [k]} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 = \sum_{i=1}^n \left\{ \sum_{j=1}^k \phi_j^{(i)} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 \right\}.$$

Lloyd's algorithm (sometimes called *the* k -means algorithm)

Initialize $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$ somehow. Then repeat until convergence:

- Holding $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ fixed, pick optimal $\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(n)}$.

Set $\phi^{(i)}$ so $\mathbf{x}^{(i)}$ is assigned to closest $\mathbf{c}_j$.

- Holding $\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(n)}$ fixed, pick optimal $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$.

ALTERNATING OPTIMIZATION ALGORITHM

Assignment variables

For each data point $\mathbf{x}^{(i)}$, let $\phi^{(i)} \in \{0, 1\}^d$ denote its “one-hot” representation:

$$\phi_j^{(i)} = \mathbb{1}\{\mathbf{x}^{(i)} \text{ is assigned to cluster } j\}.$$

Objective becomes (for optimal setting of $\phi^{(i)}$ s)

$$\sum_{i=1}^n \min_{j \in [k]} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 = \sum_{i=1}^n \left\{ \sum_{j=1}^k \phi_j^{(i)} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2 \right\}.$$

Lloyd's algorithm (sometimes called *the* k -means algorithm)

Initialize $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$ somehow. Then repeat until convergence:

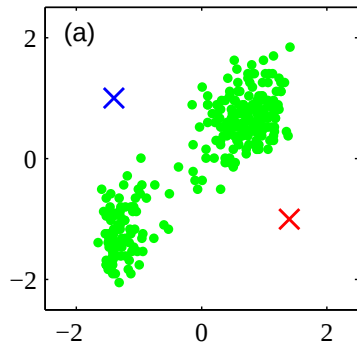
- ▶ Holding $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ fixed, pick optimal $\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(n)}$.

Set $\phi^{(i)}$ so $\mathbf{x}^{(i)}$ is assigned to closest $\mathbf{c}_j$.

- ▶ Holding $\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(n)}$ fixed, pick optimal $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$.

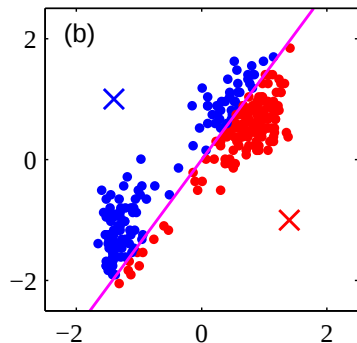
Set $\mathbf{c}_j$ to be the average of the $\mathbf{x}^{(i)}$ assigned to cluster j .

SAMPLE RUN OF LLOYD'S ALGORITHM



Arbitrary initialization of c_1 and c_2 .

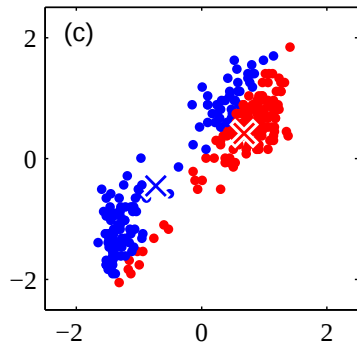
SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 1

Optimize assignments $\phi^{(i)}$.

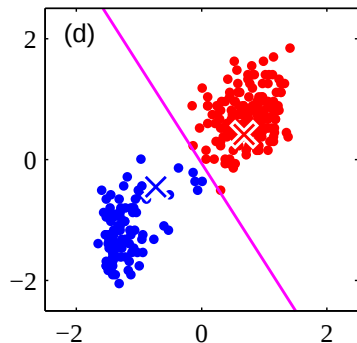
SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 1

Optimize representatives c_j .

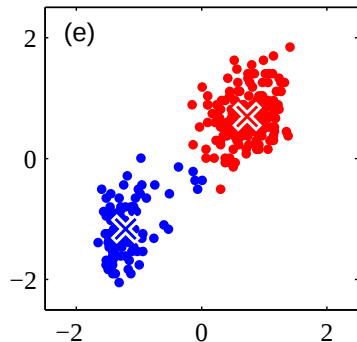
SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 2

Optimize assignments $\phi^{(i)}$.

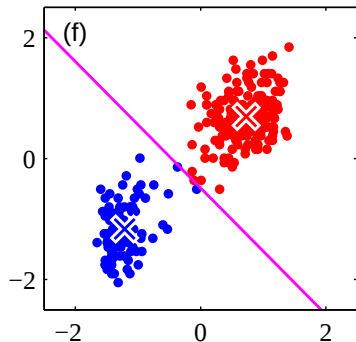
SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 2

Optimize representatives c_j .

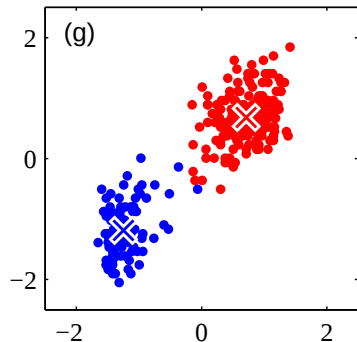
SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 3

Optimize assignments $\phi^{(i)}$.

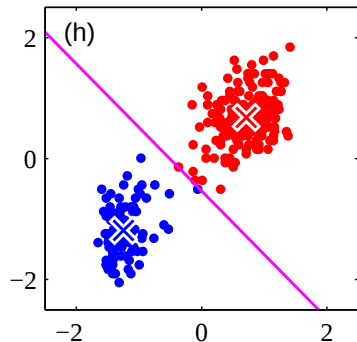
SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 3

Optimize representatives c_j .

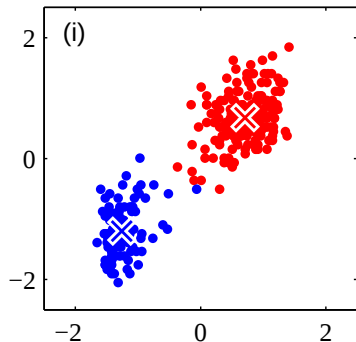
SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 4

Optimize assignments $\phi^{(i)}$.

SAMPLE RUN OF LLOYD'S ALGORITHM



Iteration 4

Optimize representatives c_j .

INITIALIZING LLOYD'S ALGORITHM

Basic idea: Choose initial centers to have good coverage of the data points.

Farthest-first traversal

For $j = 1, 2, \dots, k$:

- Pick $c_j \in \mathbb{R}^d$ from among $x^{(1)}, x^{(2)}, \dots, x^{(n)}$ farthest from previously chosen $c_1, c_2, \dots, c_{j-1}$.
(c_1 chosen arbitrarily.)

INITIALIZING LLOYD'S ALGORITHM

Basic idea: Choose initial centers to have good coverage of the data points.

Farthest-first traversal

For $j = 1, 2, \dots, k$:

- Pick $\mathbf{c}_j \in \mathbb{R}^d$ from among $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ farthest from previously chosen $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{j-1}$.
($\mathbf{c}_1$ chosen arbitrarily.)

But this can be thrown off by outliers...

INITIALIZING LLOYD'S ALGORITHM

Basic idea: Choose initial centers to have good coverage of the data points.

Farthest-first traversal

For $j = 1, 2, \dots, k$:

- Pick $\mathbf{c}_j \in \mathbb{R}^d$ from among $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ farthest from previously chosen $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{j-1}$.
($\mathbf{c}_1$ chosen arbitrarily.)

But this can be thrown off by outliers...

A better idea:

D^2 sampling (a.k.a. “ k -means++”)

For $j = 1, 2, \dots, k$:

- Randomly pick $\mathbf{c}_j \in \mathbb{R}^d$ from among $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$ according to distribution

$$\Pr(\mathbf{c}_j = \mathbf{x}^{(i)}) \propto \min_{j' < j} \|\mathbf{x}^{(i)} - \mathbf{c}_{j'}\|_2^2.$$

(Uniform distribution when $j = 1$.)

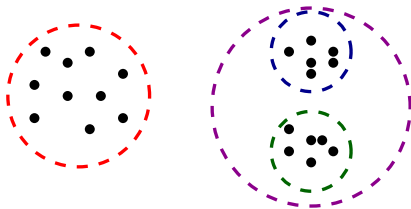
CHOOSING k

- ▶ Usually by hold-out validation / cross-validation on auxiliary task (e.g., supervised learning task).

CHOOSING k

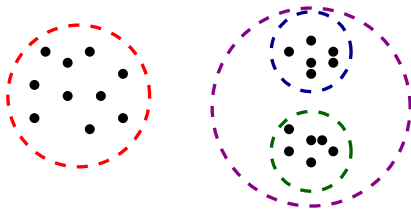
- ▶ Usually by hold-out validation / cross-validation on auxiliary task (e.g., supervised learning task).
- ▶ *Heuristic*: Find large gap between $k - 1$ -means cost and k -means cost.

CLUSTERING AT MULTIPLE SCALES



$k = 2$ or $k = 3$?

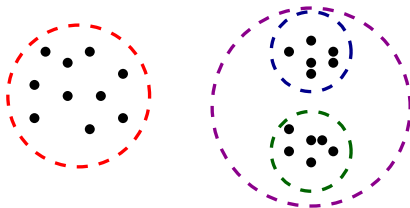
CLUSTERING AT MULTIPLE SCALES



$k = 2$ or $k = 3$?

Hierarchical clustering: encode clusterings for all values of k in a tree.

CLUSTERING AT MULTIPLE SCALES



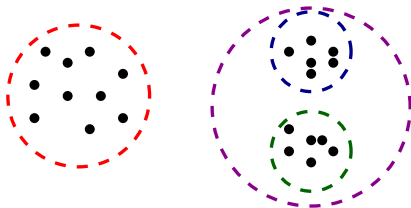
$k = 2$ or $k = 3$?

Hierarchical clustering: encode clusterings for all values of k in a tree.

Caveat: not always possible.



CLUSTERING AT MULTIPLE SCALES



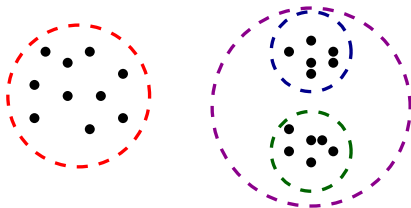
$k = 2$ or $k = 3$?

Hierarchical clustering: encode clusterings for all values of k in a tree.

Caveat: not always possible.



CLUSTERING AT MULTIPLE SCALES



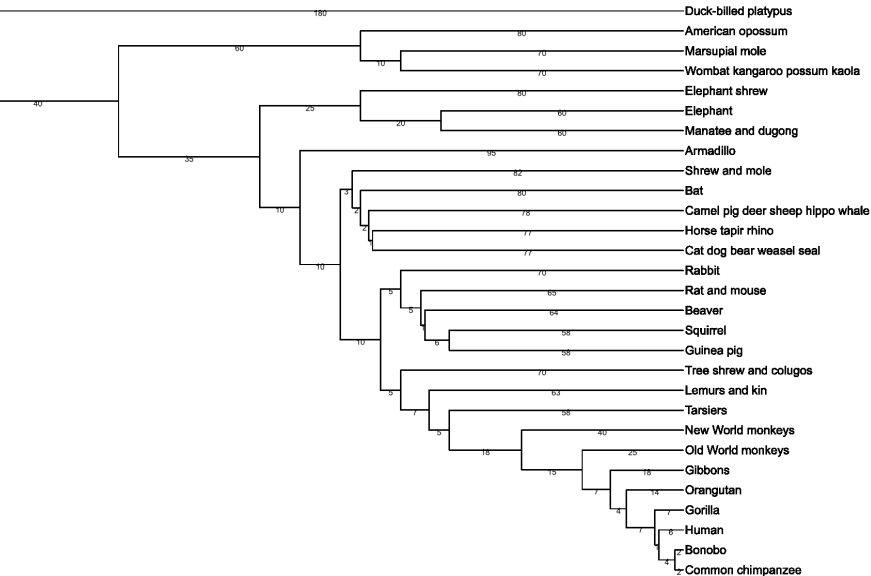
$k = 2$ or $k = 3$?

Hierarchical clustering: encode clusterings for all values of k in a tree.

Caveat: not always possible.



EXAMPLE: PHYLOGENETIC TREE



Divisive (top-down) clustering

- ▶ Partition data into two groups (e.g., via k -means clustering with $k = 2$).
- ▶ Recurse on each part.

HIERARCHICAL CLUSTERING

Divisive (top-down) clustering

- ▶ Partition data into two groups (e.g., via k -means clustering with $k = 2$).
- ▶ Recurse on each part.

Agglomerative (bottom-up) clustering

- ▶ Start with every point $\mathbf{x}^{(i)}$ in its own cluster.
- ▶ Repeatedly merge “closest” pair of clusters.

HIERARCHICAL CLUSTERING

Divisive (top-down) clustering

- ▶ Partition data into two groups (e.g., via k -means clustering with $k = 2$).
- ▶ Recurse on each part.

Agglomerative (bottom-up) clustering

- ▶ Start with every point $\mathbf{x}^{(i)}$ in its own cluster.
- ▶ Repeatedly merge “closest” pair of clusters.

Example: *Ward's average linkage method*

$$\text{dist}(C, \tilde{C}) := \frac{|C| \cdot |\tilde{C}|}{|C| + |\tilde{C}|} \|\text{mean}(C) - \text{mean}(\tilde{C})\|_2^2$$

(the increase in k -means cost caused by merging C and $\tilde{C}$).

DICTIONARY LEARNING (A.K.A. SPARSE CODING)

DICTIONARY LEARNING

Goal: Find representatives $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d$ such that each $\mathbf{x}^{(i)}$ is “well-represented” by a linear combination of $\leq s$ such representatives $\mathbf{c}_j$.

Special case: $s = 1 \implies$ clustering/quantization.

GENERALIZING k -MEANS

k -means objective

$$\min_{\mathbf{C}, \Phi} \sum_{i=1}^n \left\| \mathbf{x}^{(i)} - \mathbf{C} \phi^{(i)} \right\|_2^2$$

- ▶ $\Phi = [\phi^{(1)} | \phi^{(2)} | \dots | \phi^{(n)}] \in \{0, 1\}^{k \times n}$ are the cluster assignments.
- ▶ $\mathbf{C} = [\mathbf{c}_1 | \mathbf{c}_2 | \dots | \mathbf{c}_k] \in \mathbb{R}^{d \times k}$ are the cluster representatives.

GENERALIZING k -MEANS

k -means objective

$$\min_{\mathbf{C}, \Phi} \sum_{i=1}^n \left\| \mathbf{x}^{(i)} - \mathbf{C} \phi^{(i)} \right\|_2^2$$

- ▶ $\Phi = [\phi^{(1)} | \phi^{(2)} | \dots | \phi^{(n)}] \in \{0, 1\}^{k \times n}$ are the cluster assignments.
- ▶ $\mathbf{C} = [\mathbf{c}_1 | \mathbf{c}_2 | \dots | \mathbf{c}_k] \in \mathbb{R}^{d \times k}$ are the cluster representatives.

Lloyd's algorithm:

Initialize $\mathbf{C}$ somehow. Then repeat:

- ▶ Holding $\mathbf{C}$ fixed, pick optimal Φ .
- ▶ Holding Φ fixed, pick optimal $\mathbf{C}$.

GENERALIZING k -MEANS

k -means objective

$$\min_{\mathbf{C}, \Phi} \sum_{i=1}^n \left\| \mathbf{x}^{(i)} - \mathbf{C} \phi^{(i)} \right\|_2^2$$

- ▶ $\Phi = [\phi^{(1)} | \phi^{(2)} | \dots | \phi^{(n)}] \in \{0, 1\}^{k \times n}$ are the cluster assignments.
- ▶ $\mathbf{C} = [\mathbf{c}_1 | \mathbf{c}_2 | \dots | \mathbf{c}_k] \in \mathbb{R}^{d \times k}$ are the cluster representatives.

Lloyd's algorithm:

Initialize $\mathbf{C}$ somehow. Then repeat:

- ▶ Holding $\mathbf{C}$ fixed, pick optimal Φ .
- ▶ Holding Φ fixed, pick optimal $\mathbf{C}$.

Generalization

Permit each $\phi^{(i)}$ to have up to s non-zero entries (not necessarily equal to 1).

DICTIONARY LEARNING

Common dictionary learning objective

$$\min_{C, \Phi} \sum_{i=1}^n \left\| \mathbf{x}^{(i)} - C\phi^{(i)} \right\|_2^2.$$

Generalization of Lloyd's algorithm:

Initialize C somehow. Then repeat:

- ▶ Holding C fixed, pick (near) optimal Φ .
- ▶ Holding Φ fixed, pick optimal C .

Common dictionary learning objective

$$\min_{C, \Phi} \sum_{i=1}^n \left\| \mathbf{x}^{(i)} - C\phi^{(i)} \right\|_2^2.$$

Generalization of Lloyd's algorithm:

Initialize C somehow. Then repeat:

- Holding C fixed, pick (near) optimal Φ .

n sparse regression problems: use Lasso, forward stepwise regression, ...

- Holding Φ fixed, pick optimal C .

DICTIONARY LEARNING

Common dictionary learning objective

$$\min_{C, \Phi} \sum_{i=1}^n \left\| \mathbf{x}^{(i)} - C\phi^{(i)} \right\|_2^2.$$

Generalization of Lloyd's algorithm:

Initialize C somehow. Then repeat:

- Holding C fixed, pick (near) optimal Φ .

n sparse regression problems: use Lasso, forward stepwise regression, ...

- Holding Φ fixed, pick optimal C .

Ordinary least squares solution:

$$C^\top := (\Phi\Phi^\top)^{-1}\Phi X$$

where i -th row of X is $\mathbf{x}^{(i)\top}$.

Common dictionary learning objective

$$\min_{C, \Phi} \sum_{i=1}^n \left\| \mathbf{x}^{(i)} - C\phi^{(i)} \right\|_2^2.$$

Generalization of Lloyd's algorithm:

Initialize C somehow. Then repeat:

- Holding C fixed, pick (near) optimal Φ .

n sparse regression problems: use Lasso, forward stepwise regression, ...

- Holding Φ fixed, pick optimal C .

Ordinary least squares solution:

$$C^\top := (\Phi\Phi^\top)^{-1}\Phi X$$

where i -th row of X is $\mathbf{x}^{(i)\top}$.

Typical initialization: random (e.g., i.i.d. $N(0,1)$ entries), or D^2 sampling.

EXAMPLE: MIXED-MEMBERSHIP MODEL

Represent corpus of documents by counts of words they contain:

	doc. 1	doc. 2	doc. 3	...
aardvark	3	7	2	...
abacus	0	0	4	...
abalone	0	4	0	...
⋮	⋮	⋮	⋮	

EXAMPLE: MIXED-MEMBERSHIP MODEL

Represent corpus of documents by counts of words they contain:

	doc. 1	doc. 2	doc. 3	...
aardvark	3	7	2	...
abacus	0	0	4	...
abalone	0	4	0	...
⋮	⋮	⋮	⋮	

Modeling assumption:

- ▶ k "topics", each represented by a distributions over vocabulary words $\beta_1, \beta_2, \dots, \beta_k \in \mathbb{R}^d$.
- ▶ Each document i is associated with $\leq s$ topics.

Document i 's count vector is drawn from a multinomial distribution with probabilities given by $\sum_{t=1}^k w_t^{(i)} \beta_t$ where $w^{(i)}$ is a probability vector with $\leq s$ non-zero entries.

EXAMPLE: MIXED-MEMBERSHIP MODEL

In expectation:

$$\begin{array}{ccc} \begin{array}{|c|} \hline \square \\ \hline \end{array} & = & \begin{array}{|c|} \hline \square \\ \hline \end{array} \begin{array}{|c|} \hline \square \\ \hline \end{array} \\ \mathbb{E}(\mathbf{A}^\top) & & \mathbf{B} \quad \mathbf{\Phi} \\ (d \times n) & & (d \times k) \quad (k \times n) \end{array}$$

- ▶ $\phi_t^{(i)} = w_t^{(i)} \times \text{length of document } i$.
- ▶ $\beta_t = t$ -th column of $\mathbf{B}$

EXAMPLE: MIXED-MEMBERSHIP MODEL

In expectation:

$$\begin{array}{ccc} \begin{array}{|c|} \hline \square \\ \hline \end{array} & = & \begin{array}{|c|} \hline \square \\ \hline \end{array} \begin{array}{|c|} \hline \square \\ \hline \end{array} \\ \mathbb{E}(\mathbf{A}^\top) & & \mathbf{B} \quad \mathbf{\Phi} \\ (d \times n) & & (d \times k) \quad (k \times n) \end{array}$$

- ▶ $\phi_t^{(i)} = w_t^{(i)} \times \text{length of document } i.$
- ▶ $\beta_t = t\text{-th column of } \mathbf{B}$

Applying dictionary learning:

Identify $\beta_1, \beta_2, \dots, \beta_k$ as “representatives” $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k \in \mathbb{R}^d \dots$

- ▶ Uses of clustering:
 - ▶ Unsupervised classification (“hidden subpopulations”).
 - ▶ Quantization
 - ▶ ...
- ▶ k -means clustering: popular objective for clustering and quantization.
- ▶ Lloyd’s algorithm: alternating optimization, needs good initialization.
- ▶ Hierarchical clustering: clustering at multiple levels of granularity.
- ▶ Dictionary learning/sparse coding: generalization of clustering.