Clustering and dictionary learning

Clustering

Unsupervised classification / clustering

Unsupervised classification

- **Input**: $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output**: function $f \colon \mathbb{R}^d \to \{1, 2, \ldots, k\} =: [k]$.
- **Typical semantics**: hidden subpopulation structure.

Clustering

- **Input**: $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output**: partitioning of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ into k groups.
- Often done via unsupervised classification;
  $\Rightarrow$ "clustering" often synonymous with "unsupervised classification".
- Sometimes also have a "representative" $\boldsymbol{c}_j \in \mathbb{R}^d$ for each $j \in [k]$
  (e.g., average of the $\boldsymbol{x}_i$ in jth group) $\longrightarrow$ **quantization**.

Uses of clustering: feature representations

"One-hot" / "dummy variable" encoding of $f(\boldsymbol{x})$

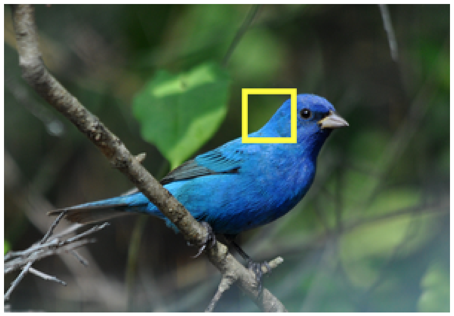
(Often used together with other features.)

$$\phi(\boldsymbol{x}) = \begin{bmatrix}0\\ \vdots\\ 0\\ 1\\ 0\\ \vdots\\ 0\end{bmatrix} \begin{array}{l} \\ \\ \\ \longleftarrow f(\boldsymbol{x}) \text{ position} \\ \\ \\ \\ \end{array}$$

Uses of clustering: feature representations

Histogram representation

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



Uses of clustering: compression

Quantization

Replace each $\boldsymbol{x}_i$ with its representative

$$\boldsymbol{x}_i \;\mapsto\; \boldsymbol{c}_{f(\boldsymbol{x}_i)} \,.$$

Example: quantization at image patch level.

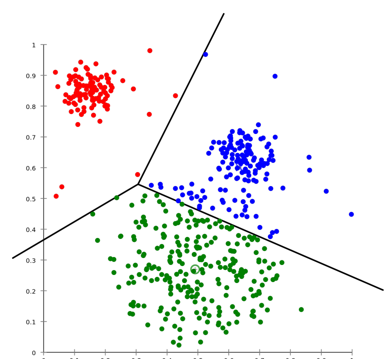
k-means clustering

k-means clustering

Problem

- **Input**: $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output**: k representatives ("centers", "means") $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \mathbb{R}^d$.
- **Objective**: choose $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \mathbb{R}^d$ to minimize

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



Natural assignment function

$$f(\boldsymbol{x}) \;:=\; \operatorname*{arg\,min}_{j \in [k]} \|\boldsymbol{x} - \boldsymbol{c}_j\|_2^2 \,.$$

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

The easy cases

k-means clustering for $k = 1$

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

$$\sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{c}\|_2^2 .$$

Solution: "bias/variance decomposition"

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

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

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

k-means clustering for $d = 1$

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

Alternating optimization algorithm

Assignment variables

For each data point $\boldsymbol{x}_i$, let $\boldsymbol{\phi}_i \in \{0,1\}^k$ denote its "one-hot" representation:

$$\phi_{i,j} \;=\; \mathbb{1}\{\boldsymbol{x}_i \text{ is assigned to cluster } j\} .$$

Objective becomes (for optimal setting of $\boldsymbol{\phi}_i$s)

$$\sum_{i=1}^{n} \min_{j \in [k]} \|\boldsymbol{x}_i - \boldsymbol{c}_j\|_2^2 \;=\; \sum_{i=1}^{n} \left\{ \sum_{j=1}^{k} \phi_{i,j} \|\boldsymbol{x}_i - \boldsymbol{c}_j\|_2^2 \right\} .$$

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

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

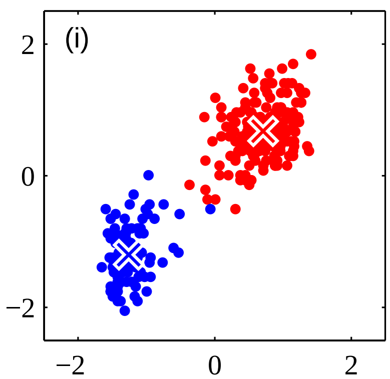
- Holding $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k$ fixed, pick optimal $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n$.

  Set $\boldsymbol{\phi}_i$ so $\boldsymbol{x}_i$ is assigned to closest $\boldsymbol{c}_j$.

- Holding $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n$ fixed, pick optimal $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k$.

  Set $\boldsymbol{c}_j$ to be the average of the $\boldsymbol{x}_i$ assigned to cluster j.

Sample run of Lloyd's algorithm



Arbitrary initialization of $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$.

Initializing Lloyd's algorithm

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

Farthest-first traversal

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

- Pick $\boldsymbol{c}_j \in \mathbb{R}^d$ from among $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ farthest from previously chosen $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{j-1}$.

  ($\boldsymbol{c}_1$ chosen arbitrarily.)

But this can be thrown off by outliers...

A better idea:

D^2 sampling (a.k.a. "k-means++")

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

- Randomly pick $\boldsymbol{c}_j \in \mathbb{R}^d$ from among $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ according to distribution

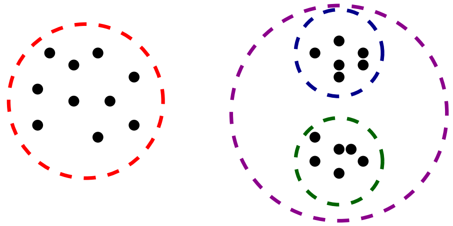
  $$P(\boldsymbol{x}_i) \;\propto\; \min_{\ell=1,2,\ldots,j-1} \|\boldsymbol{x}_i - \boldsymbol{c}_\ell\|_2^2 .$$

  (Uniform distribution when $j = 1$.)

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.

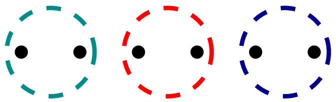
Clustering at multiple scales



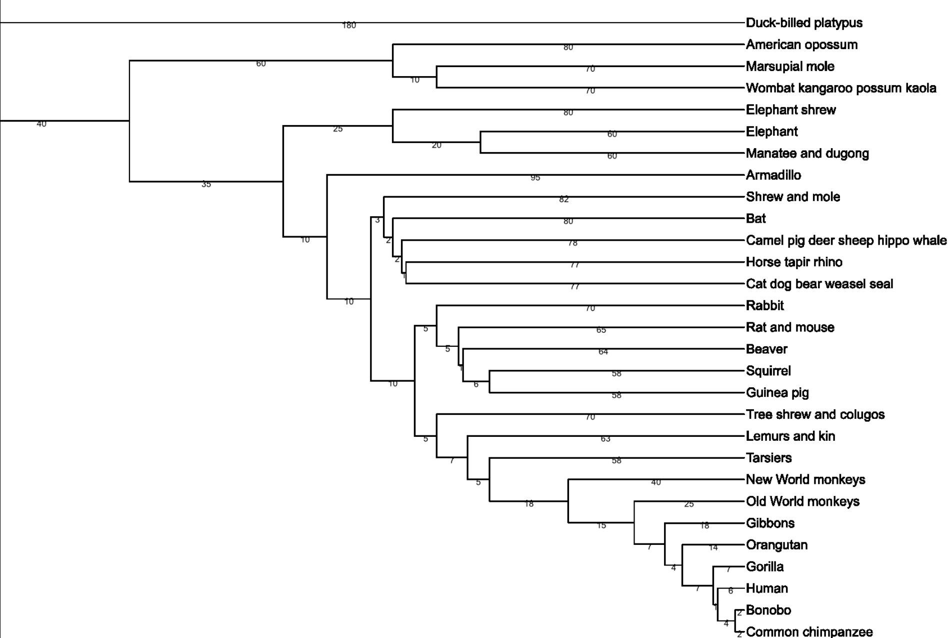
$k = 2$ or $k = 3$?

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

Caveat: not always possible.



Example: phylogenetic tree



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 $\boldsymbol{x}_i$ in its own cluster.
- Repeatedly merge "closest" pair of clusters.

  Example: *Ward's average linkage method*

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

(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 $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \mathbb{R}^d$ such that each $\boldsymbol{x}_i$ is "well-represented" by a linear combination of $\leq s$ such representatives $\boldsymbol{c}_j$.

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

Generalizing k-means

k-means objective

$$\min_{\boldsymbol{C}, \boldsymbol{\Phi}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{C}\boldsymbol{\phi}_i\|_2^2$$

- $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 | \boldsymbol{\phi}_2 | \cdots | \boldsymbol{\phi}_n] \in \{0,1\}^{k \times n}$ are the cluster assignments.
- $\boldsymbol{C} = [\boldsymbol{c}_1 | \boldsymbol{c}_2 | \cdots | \boldsymbol{c}_k] \in \mathbb{R}^{d \times k}$ are the cluster representatives.

Lloyd's algorithm:
Initialize $\boldsymbol{C}$ somehow. Then repeat:

- Holding $\boldsymbol{C}$ fixed, pick optimal $\boldsymbol{\Phi}$.
- Holding $\boldsymbol{\Phi}$ fixed, pick optimal $\boldsymbol{C}$.

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

Dictionary learning

Common dictionary learning objective

$$\min_{\boldsymbol{C}, \boldsymbol{\Phi}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{C}\boldsymbol{\phi}_i\|_2^2 .$$

Generalization of Lloyd's algorithm:
Initialize $\boldsymbol{C}$ somehow. Then repeat:

- Holding $\boldsymbol{C}$ fixed, pick (near) optimal $\boldsymbol{\Phi}$.

  n sparse regression problems (use Lasso, forward stepwise regression, ...)

- Holding $\boldsymbol{\Phi}$ fixed, pick optimal $\boldsymbol{C}$.

  Ordinary least squares solution:

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

  where i-th row of $\boldsymbol{X}$ is $\boldsymbol{x}_i^\top$.

Typical initialization: random (e.g., i.i.d. $\mathrm{N}(0,1)$ 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	$\cdots$
aardvark	3	7	2	$\cdots$
abacus	0	0	4	$\cdots$
abalone	0	4	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

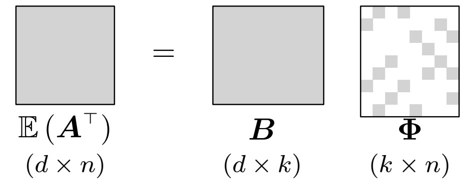
Modeling assumption:

- k "topics", each represented by a distributions over vocabulary words $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_k \in \mathbb{R}^d$.
- Each document i is associated with $\leq s$ topics.

  Document i's count vector is drawn from a multinomial distribution with probabilities given by $\sum_{t=1}^{k} w_{i,t} \boldsymbol{\beta}_t$ where $\boldsymbol{w}_i$ is a probability vector with $\leq s$ non-zero entries.

Example: mixed-membership model

In expectation:



- $\boldsymbol{\phi}_{i,t} = w_{i,t} \times$ length of document i.
- $\boldsymbol{\beta}_t = t$-th column of $\boldsymbol{B}$

Applying dictionary learning:

Identify $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_k$ as "representatives" $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \mathbb{R}^d$ ...

Recap

- 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.