### Clustering and dictionary learning

#### Unsupervised classification / clustering

**Input:** $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.

**Output:** function $f: \mathbb{R}^d \rightarrow \{1, 2, \ldots, k\} =: [k]$.

**Typical semantics:** hidden subpopulation structure.

#### Clustering

**Input:** $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.

**Output:** partitioning of $x_1, x_2, \ldots, x_n$ into $k$ groups.

Often done via unsupervised classification; ⇒ “clustering” often synonymous with “unsupervised classification”.

Sometimes also have a “representative” $c_j \in \mathbb{R}^d$ for each $j \in [k]$ (e.g., average of the $x_i$ in $j$th group) → quantization.

### Uses of clustering: feature representations

“One-hot” / “dummy variable” encoding of $f(x)$

$$
\phi(x) = \begin{bmatrix}
0 \\
\vdots \\
0 \\
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
$$

(Often used together with other features.)

$f(x)$ position
### Uses of Clustering: Feature Representations

**Histogram representation**

- Cut up each $x_i \in \mathbb{R}^d$ into different parts $x_{i,1}, x_{i,2}, \ldots, x_{i,m} \in \mathbb{R}^p$ (e.g., small patches of an image).
- Cluster all the parts $x_{i,j}$: get $k$ representatives $c_1, c_2, \ldots, c_k \in \mathbb{R}^p$.
- Represent $x_i$ by a histogram over $\{1, 2, \ldots, k\}$ based on assignments of $x_i$'s parts to representatives.

### Uses of Clustering: Compression

**Quantization**

Replace each $x_i$ with its representative

$$x_i \mapsto c_{f(x_i)}.$$

**Example:** quantization at image patch level.

### k-Means Clustering

**Problem**

- **Input:** $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output:** $k$ representatives ("centers", "means") $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$.
- **Objective:** choose $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$ to minimize

$$\sum_{i=1}^{n} \min_{j \in [k]} \|x_i - c_j\|_2^2.$$ 

**Natural assignment function**

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

**NP-hard, even if $k = 2$ or $d = 2$.**
**The easy cases**

$k$-means clustering for $k = 1$

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

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

**Solution:** “bias/variance decomposition”

$$\frac{1}{n} \sum_{i=1}^{n} \|x_i - c\|^2 = \|\mu - c\|^2 + \frac{1}{n} \sum_{i=1}^{n} \|x_i - \mu\|^2_2$$

where $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Therefore, optimal choice for $c$ is $\mu$.

$k$-means clustering for $d = 1$

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

---

**Alternating optimization algorithm**

**Assignment variables**

For each data point $x_i$, let $\phi_i \in \{0, 1\}^k$ denote its “one-hot” representation:

$$\phi_{i,j} = 1 \{x_i \text{ is assigned to cluster } j\}.$$ 

Objective becomes (for optimal setting of $\phi_i$’s)

$$\sum_{i=1}^{n} \min_{j \in [k]} \|x_i - c_j\|^2_2 = \sum_{i=1}^{n} \left\{ \sum_{j=1}^{k} \phi_{i,j} \|x_i - c_j\|^2_2 \right\}.$$ 

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

Initialize $c_1, c_2, \ldots, c_k \in \mathbb{R}^d$ somehow. Then repeat until convergence:

- Holding $c_1, c_2, \ldots, c_k$ fixed, pick optimal $\phi_1, \phi_2, \ldots, \phi_n$.
  - Set $\phi_i$ so $x_i$ is assigned to closest $c_j$.

- Holding $\phi_1, \phi_2, \ldots, \phi_n$ fixed, pick optimal $c_1, c_2, \ldots, c_k$.
  - Set $c_j$ to be the average of the $x_i$ assigned to cluster $j$.

**Sample run of Lloyd’s algorithm**

Arbitrary initialization of $c_1$ and $c_2$.

---

**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 $c_j \in \mathbb{R}^d$ from among $x_1, x_2, \ldots, x_n$ farthest from previously chosen $c_1, c_2, \ldots, c_{j-1}$.
  - ($c_1$ 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 $c_j \in \mathbb{R}^d$ from among $x_1, x_2, \ldots, x_n$ according to distribution
  $$\Pr(c_j = x_i) \propto \min_{j' < j} \|x_i - c_{j'}\|^2_2.$$ 

  (Uniform distribution when $j = 1$.)
**Choosing \( k \)**

- Usually by hold-out validation / cross-validation on auxiliary task (e.g., supervised learning task).
- **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 \( x_i \) in its own cluster.
- Repeatedly merge “closest” pair of clusters.

Example: **Ward’s average linkage method**

\[
\text{dist}(C, \tilde{C}) := \frac{|C| \cdot |\tilde{C}|}{|C| + |\tilde{C}|} \left\| \text{mean}(C) - \text{mean}(\tilde{C}) \right\|^2
\]

(the increase in \( k \)-means cost caused by merging \( C \) and \( \tilde{C} \)).
Dictionary learning
(a.k.a. sparse coding)

**Goal**: Find representatives \(c_1, c_2, \ldots, c_k \in \mathbb{R}^d\) such that each \(x_i\) is “well-represented” by a linear combination of \(\leq s\) such representatives \(c_j\).

**Special case**: \(s = 1 \implies\) clustering/quantization.

### Generalizing \(k\)-means

**\(k\)-means objective**

\[
\min_{C, \Phi} \sum_{i=1}^{n} \|x_i - C\Phi_i\|_2^2
\]

- \(\Phi = [\phi_1| \phi_2| \cdots| \phi_n] \in \{0, 1\}^{k \times n}\) are the cluster assignments.
- \(C = [c_1| c_2| \cdots| c_k] \in \mathbb{R}^{d \times k}\) are the cluster representatives.

**Lloyd’s algorithm**:

Initialize \(C\) somehow. Then repeat:

- Holding \(C\) fixed, pick (near) optimal \(\Phi\).
- Holding \(\Phi\) fixed, pick optimal \(C\).

**Generalization**

Permit each \(\phi_i\) to have up to \(s\) non-zero entries (not necessarily equal to 1).

### Dictionary learning

**Common dictionary learning objective**

\[
\min_{C, \Phi} \sum_{i=1}^{n} \|x_i - C\Phi_i\|_2^2
\]

**Generalization of Lloyd’s algorithm**:

Initialize \(C\) somehow. Then repeat:

- Holding \(C\) fixed, pick (near) optimal \(\Phi\).
- Holding \(\Phi\) fixed, pick optimal \(C\).

**Ordinary least squares solution**:

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

where \(i\)-th row of \(X\) is \(x_i^\top\).

**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:

<table>
<thead>
<tr>
<th></th>
<th>aardvark</th>
<th>abacus</th>
<th>abalone</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc. 1</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>doc. 2</td>
<td>7</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>doc. 3</td>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Modeling assumption:

- \( k \) “topics”, each represented by a distribution over vocabulary words \( \beta_1, \beta_2, \ldots, \beta_k \in \mathbb{R}^d \).
- Each document \( i \) is associated with \( \leq s \) topics.
  
  Document \( i \)'s count vector is drawn from a multinomial distribution with probabilities given by \( \sum_{t=1}^k w_{i,t} \beta_t \) where \( w_i \) is a probability vector with \( \leq s \) non-zero entries.

In expectation:

\[
\mathbb{E}(A^\top) = \Phi
\]

\( A \) \( (d \times n) \)

\( \Phi \) \( (k \times n) \)

- \( \phi_{i,t} = w_{i,t} \times \text{length of document } i \).
- \( \beta_t = t\text{-th column of } B \)

Applying dictionary learning:

Identify \( \beta_1, \beta_2, \ldots, \beta_k \) as “representatives” \( c_1, c_2, \ldots, c_k \in \mathbb{R}^d \).

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.