Clustering

COMS 4771

1. Clustering

Unsupervised classification / clustering

- **Input**: $x_1, \ldots, x_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output**: function $f: \mathbb{R}^d \rightarrow \{1, \ldots, k\} =: [k]$.
- **Typical semantics**: hidden subpopulation structure.

Clustering

- **Input**: $x_1, \ldots, x_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output**: partitioning of $x_1, \ldots, x_n$ 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 $f(x)$

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

(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 $x_{i,1}, \ldots, x_{i,m} \in \mathbb{R}^p$ (e.g., small patches of an image).
- Cluster all the parts $x_{i,j}$: get $k$ representatives $c_1, \ldots, c_k \in \mathbb{R}^p$.
- Represent $x_i$ by a histogram over $\{1, \ldots, k\}$ 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.

---

2. **$k$-means**

- **Input**: $x_1, \ldots, x_n \in \mathbb{R}^d$, target cardinality $k \in \mathbb{N}$.
- **Output**: $k$ “means” $c_1, \ldots, c_k \in \mathbb{R}^d$.
- **Objective**: choose $c_1, \ldots, c_k \in \mathbb{R}^d$ 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

▶ \textit{k-means clustering for } k = 1

\textbf{Problem:} Pick } \mathbf{c} \in \mathbb{R}^d \text{ to minimize } \sum_{i=1}^{n} \| \mathbf{x}_i - \mathbf{c} \|^2_2.

\textbf{Solution:} Using \textquote{bias/variance decomposition}, best choice is } \mathbf{c} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i.

▶ \textit{k-means clustering for } d = 1

Dynamic programming in time } \mathcal{O}(n^2k).

Lloyd\textquotesingle s algorithm is a popular heuristic for general \textit{k} and \textit{d}.

(There are many other algorithms for \textit{k}-means clustering.)

---

Lloyd\textquotesingle s algorithm

▶ Start with some initial \textquote{means} } \mathbf{c}_1, \ldots, \mathbf{c}_k.

▶ Repeat:

▶ Partition } \mathbf{x}_1, \ldots, \mathbf{x}_n \text{ into } k \text{ clusters } \mathcal{C}_1, \ldots, \mathcal{C}_k, \text{ based on distance to current \textquote{means}}:

\[ \mathbf{x}_i \mapsto \arg \min_{j \in [k]} \| \mathbf{x}_i - \mathbf{c}_j \|^2_2, \quad i \in [n], \]

\text{breaking ties according to any fixed rule.}

▶ Update \textquote{means}:

\[ \mathbf{c}_j := \text{mean}(\mathcal{C}_j), \quad j \in [k]. \]

---

Sample run of Lloyd\textquotesingle s algorithm

![Sample run of Lloyd's algorithm](image)

Arbitrary initialization of } \mathbf{c}_1 \text{ and } \mathbf{c}_2.

---

Initializing Lloyd\textquotesingle s algorithm

\textbf{Basic idea:} Choose initial centers to have good coverage of the data points.

\textbf{Farthest-first traversal}

For } j = 1, \ldots, k:

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

\text{\quad (} \mathbf{c}_1 \text{ chosen arbitrarily.)}

But this can be thrown off by outliers... \dots

\textbf{D^2 sampling (a.k.a. \textquote{\textit{k-means++}})}

For } j = 1, \ldots, k:

▶ Randomly pick } \mathbf{c}_j \in \mathbb{R}^d \text{ from among } \mathbf{x}_1, \ldots, \mathbf{x}_n \text{ according to distribution}

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

\text{(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.

**Example: Pearson’s crabs (1894)**

**Data**: ratio of forehead-width to body-length for 1000 crabs.

Maybe the sample is comprised of two different sub-species of crab?

---

3. Gaussian mixture model

\[
Y_i \sim \text{Discrete}(\pi_1, \ldots, \pi_K) \\
X_i \mid Y_i = j \sim N(\mu_j, \Sigma_j)
\]

Model is for data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times [K]$.
But we still use it even though we only observe the $x_i$. 

---

Gaussian mixture model
Parameter estimation for Gaussian mixtures

Maximum likelihood estimation

Likelihood of \( \theta = (\pi_1, \mu_1, \Sigma_1, \ldots, \pi_K, \mu_K, \Sigma_K) \) given obs. \( \hat{x}_1, \ldots, \hat{x}_n \):

\[
\arg\max_{\theta} \sum_{i=1}^{n} \ln \left\{ \sum_{j=1}^{K} \pi_j \cdot \frac{1}{(2\pi)^{d/2} \det(\Sigma_j)^{1/2}} \exp \left( -\frac{1}{2} (\hat{x}_i - \mu_j)^T \Sigma_j^{-1} (\hat{x}_i - \mu_j) \right) \right\}.
\]

The \( \ln \left\{ \sum_{j=1}^{K} \cdots \right\} \) does not simplify nicely!

Objective function is not concave function of model parameters.

Local maxima not guaranteed to be global maxima; could be arbitrarily far from / worse than the MLE.

Local optimization

- For the purpose of modeling the density of \( X \), a "good enough" local maximizer could be sufficient.
- If the data are actually generated by a Gaussian mixture distribution \( P_{\theta_{\ast}} \), then \( \theta_{\ast} \) may be close to some local maximizer of the likelihood.
Expectation-Maximization (E-M) algorithm

- Start with some initial parameters $\theta = (\hat{\pi}_1, \hat{\mu}_1, \hat{\Sigma}_1, \ldots, \hat{\pi}_K, \hat{\mu}_K, \hat{\Sigma}_K)$.
- Repeat:
  - **(E-step)** Compute “soft assignments” of data to components, using current parameters $\theta$:
    \[ w_{i,j} := \frac{\hat{\pi}_j}{\sum_{j'=1}^K \frac{\hat{\pi}_{j'}}{\det(\hat{\Sigma}_{j'})^{1/2}}} \exp \left( -\frac{1}{2} (\tilde{x}_i - \hat{\mu}_j)^T \hat{\Sigma}_j^{-1} (\tilde{x}_i - \hat{\mu}_j) \right). \]
  - **(M-step)** Update parameters:
    \[ \hat{\pi}_j := \frac{1}{n} \sum_{i=1}^n w_{i,j} \]
    \[ \hat{\mu}_j := \frac{1}{n \hat{\pi}_j} \sum_{i=1}^n w_{i,j} \tilde{x}_i \]
    \[ \hat{\Sigma}_j := \frac{1}{n \hat{\pi}_j} \sum_{i=1}^n w_{i,j} (\tilde{x}_i - \hat{\mu}_j)(\tilde{x}_i - \hat{\mu}_j)^T. \]

Some details

- **No step sizes** (unlike gradient ascent).
- **Initialization**: a bit of an art. E.g., choose $\mu_1, \ldots, \mu_K$ from among $\tilde{x}_1, \ldots, \tilde{x}_n$.
- **Starved clusters**: problems can occur if $\hat{\pi}_j$ becomes too small (e.g., $\hat{\Sigma}_j$ could be near singular). Remove/replace such components.
- **Log-likelihood of E-M iterates is non-decreasing**: converges to a stationary point.
  
  \[ \therefore \text{Run E-M from many random initializations; pick the result with highest likelihood.} \]
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

Agglomerative clustering

- Start with every point $x_i$ in its own cluster.
- Repeatedly merge “closest” pair of clusters, until only one cluster remains.

**Single-linkage**

$$\text{dist}(C, C') := \min_{a \in C, a' \in C'} \| a - a' \|^2.$$  

**Complete-linkage**

$$\text{dist}(C, C') := \max_{a \in C, a' \in C'} \| a - a' \|^2.$$  

**Average-linkage**

Many variants. E.g., Ward’s average linkage

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

Key takeaways

- Uses of clustering:
  - Unsupervised classification (“hidden subpopulations”).
  - Quantization
  - . . .
  - $k$-means clustering: popular objective for clustering and quantization.
  - Lloyd’s algorithm: alternating optimization.
  - Gaussian mixture models and EM algorithm.
  - Hierarchical clustering: clustering at multiple levels of granularity.