Announcements

- Projects: three types
  - Focused literature survey
  - A non-trivial application of ML
  - Theoretical and/or empirical analysis of an advanced ML model/algorithm.

- Write-up (up to 8 pages)

- No collaboration on projects!

- Proposal (2 page) due on or before Nov 22nd.
Review: $k$-means clustering

1. Initialize $k$ means $\mu_1, \ldots, \mu_k$ to random locations.
   - E.g., set to $k$ randomly chosen distinct examples.

2. Repeat until no change in assignment:
   
   **E-step:** Assign each example to the closest mean:
   
   $$ y_i = \arg\min_c \|x_i - \mu_c\|. $$

   **M-step:** Reestimate each mean based only on examples assigned to it:

   Let $N_c = |\{x_i : y_i = c\}|$; $\mu_c = \frac{1}{N_c} \sum_{y_i = c} x_i$.  

$k$-means example

1st iteration

3rd

5th (last)
$k$-means example

1st iteration

3rd

5th (last)
Plan for today

• Other clustering methods:
  – Hierarchical clustering,
  – Spectral clustering.
Vector Quantization

- We can use the cluster mean as a prototype representing all the examples assigned to the cluster.

- **Vector quantization**: construct a codebook using $k$-means.

- Whenever need to transmit $\mathbf{x}$, transmit instead the closest codebook.
  - The bits to transmit: $\log kd$ once + $\log k$ for every message.
Setting $k$

- How can we set $k$?
How can we set $k$? Cross-validation doesn’t work (why?)
Setting $k$

- How can we set $k$? Cross-validation doesn’t work (why?)

- The relevant statistic: *within-class dissimilarity*

  \[ W_k = \sum_{c=1}^{k} \sum_{y_i = y_j = c} \| x_i - x_j \|^2. \]

- A popular (heuristic) strategy: look for an “elbow” in $W_k$
Setting $k$

- How can we set $k$? Cross-validation doesn’t work (why?)

- The relevant statistic: *within-class dissimilarity*

\[
W_k = \sum_{c=1}^{k} \sum_{y_i = y_j = c} \|x_i - x_j\|^2.
\]

- A popular (heuristic) strategy: look for an “elbow” in $W_k$
Mixture of Gaussians EM versus $k$-means

- $k$-means:
  - No probabilistic model $\Rightarrow$ no estimated density.
  - faster to compute (only a single explanation for each data point).
  - Limited by the underlying assumption of spherical clusters

- We can bring back the covariance—get “hard EM”.
  - Still limited by the shape of the covariance (ellipsoid).

- Both EM and $k$-means depend on initialization (can get stuck in local optima).
  - Useful trick: run $k$-means and use the result to initialize EM.
Practical aspects

- Can have empty clusters
  - Take the example with highest distance to its mean, and create a new cluster.

- Many strategies for initialization
  - Start with a random example as $\mu_1$; then,
    \[
    \mu_c = \arg\max_x \min_{j=1,\ldots,c} \|x - \mu_j\|.
    \]

- Robustness: we want to diminish the influence of outliers
  - Set a threshold on the distance;
  - Ignore top percentile of distances in the M-step.
**$k$-medoids clustering**

- A generalization of $k$-means for distances $D(x_1, x_2)$ other than $L_2$ norms
  - E.g., using $L_1$ makes the clustering more robust.

- Also, we often want cluster centers to be valid observations themselves ("prototypes")

- $k$-medoids algorithm: initialize the clusters to randomly selected examples, and iterate:

  **E-step:** for each $i = 1, \ldots, N$

  $$y_i = \arg\min_c D(x_i, m_c)$$

  **M-step:** for each cluster $c = 1, \ldots, k$

  $$i^*_c = \arg\min_i \sum_{j: y_j = c} D(x_i, x_j)$$
Hierarchical structure discovery

- In some cases we want to explore the structure of the similarities in the data beyond partitioning to $k$ groups.

- E.g., a hierarchical structure (subclusters, sub-subclusters etc.)
Hierarchical clustering

• *Hierarchical clustering*: produce hierarchical representation, that can be depicted as a tree—dendrogram.
Bottom-up agglomeration
Bottom-up agglomeration
Bottom-up agglomeration
Bottom-up agglomeration
Bottom-up agglomeration
Bottom-up agglomeration
Bottom-up agglomeration
Bottom-up agglomeration
Agglomerative hierarchical clustering

- Denote by $C^m_i$ the i-th cluster at level $n$.

- Initialize $C^N_i = \{x_i\}$.

- At level $n = N - 1, N - 2, \ldots, 1$:
  - Find two closest clusters $C^{m+1}_i, C^{m+1}_j$:
    \[
    D(C^{m+1}_i, C^{m+1}_j) = \max_{l,m} D(C^{m+1}_l, C^{m+1}_m)
    \]
  - Merge them: $C^n_1 = C^{m+1}_i \cup C^{m+1}_j$. For the rest of the clusters, $C^n_l = C^{m+1}_l$
Linkage schemes

- How do we measure distances between clusters (groups of examples)?
  - **Single linkage** (nearest neighbor)
    \[ D(A, B) = \min_{a \in A, b \in B} D(a, b) \]
  - **Average linkage**:
    \[ D(A, B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} D(a, b) \]
  - **Complete linkage** (furthest neighbor)
    \[ D(A, B) = \max_{a \in A, b \in B} D(a, b) \]
Variations on hierarchical clustering

- *Divisive* clustering: top-down partition instead of bottom-up agglomeration.

- Can apply the furthest neighbor idea:
  - Assign the first cluster to a random example;
  - At each level find the example *farthest* from the current clusters, and assign the new cluster to it.

- By cutting at some levels, we can create partition to $k$ clusters.
What is missing?
Spectral clustering

- Suppose we have a \( N \times N \) distance matrix

- We can represent the data as a graph:
  - \( N \) vertices,
  - edges corresponding to nearest neighbors.
Random walk model

- Assign weights to edges: \( W_{ij} = \exp(-\beta \|x_i - x_j\|) \) (or zero if \( x_i \) and \( x_j \) not connected)

- The weight of a path \( x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_n \) is

\[
W_{12} \cdot W_{23} \cdots W_{n-1,n} = \exp\left( \beta \sum_{i=1}^{n-1} \|x_i - x_{i+1}\| \right)
\]

- The idea behind spectral clustering: imagine a random walk with probability of step \( i \rightarrow j \) given by

\[
P_{ij} = \frac{W_{ij}}{\sum_l W_{il}}.
\]

  - If we start within a cluster, we will likely remain within that cluster for a long time.
Next time

Finish spectral clustering;
Dimensionality reduction.