Weekly report of lessons

Name: Mayank Kumar
Roll No: 19CS30029
The week: 27-09-2021 to 03-10-2021

The topics covered:

Determine optimal K for clustering: Cluster Validity Indices, External indices, Internal indices, Stability check based clustering, Wang's method of cross-validation; Generalizing K-Means: mixture densities: Mixture of Gaussians; Expectation-Maximization (EM) Algorithm: Parameter re-estimation; Hierarchical clustering: Hierarchical clustering algorithm, distance between a pair of clusters; Graph-based approaches: Clique Graphs, Transforming into Clique Graphs, Distance Graphs, Corrupted Cliques Problem, Parallel Classification with Cores (PCC): Algorithm, Time Complexity; Cluster Affinity Search Technique (CAST): Algorithm; DBSCAN

Summary topic wise:

Determine optimal K: By maximizing or minimizing cluster validity index depending upon the nature of metric, or by checking stable clustering results with random initialization.
Cluster Validity Indices: Finds optimal K.

- External indices when reference partitioning information is given.
  Methods - Normalized Mutual Interface (NMI), FM Index, Set matching measures.
  NMI = 2I(Y;C)/(H(Y) + H(C)), where H is entropy, Y is cluster label, C is class label and I(Y;C) = H(Y) - H(Y|C)
- Internal indices from variance distribution and structure of clusters.
  Methods - Silhouette index: average of (a-b)/max(a,b), where a is average intra-cluster distance and b is average nearest cluster distance.
  Calinski-Harabasz (CH) Index: $CH(K) = \frac{(J(1)-J(K))/(K-1)}{J(K)/(n-K)}$; J(i) is SSE (K=i), K is no. of clusters

Stability check based clustering: For appropriate K, similar partitioning is seen for repeated clustering.
Wang's method of cross-validation: Input data is permuted c times and each time, data is divided into S_1, S_2 and S_3 such that $|S_1|=|S_2|=m$. K-means is performed on S_1 and S_2 and tested on S_3. Average of number of disagreements is computed for c observations, and such K is chosen which minimizes this average.

Generalizing K-Means: mixture densities: $P(x) = \sum_{i=1}^{K} P(x|G_i)P(G_i)$, where G_i represents i^{th} cluster and K is number of components (hyperparameter). For multivariate Gaussian distribution, $P(x|G_i) \sim N(\mu_i, \Sigma_i)$

Mixture of Gaussians: The cluster centers are augmented by covariance matrix and their values are re-estimated from corresponding samples.
Mahalanobis distance $d(x, \mu_k; \Sigma_k) = (x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)$; μ_k -> cluster center, Σ_k -> covariance matrix

Parametric pdf $p(x|\{\pi_k, \mu_k, \Sigma_k\}) = \sum_k \pi_k N(x \mid \mu_k, \Sigma_k)$, where π_k is mixing coefficients

Expectation-Maximization Algorithm: Used to find local maximum likelihood parameters iteratively. The E and M steps are integrated till convergence. E-Step: Each x is assigned to cluster whose probability of belongingness for x is maximum. M-Step: Re-estimate parameters from class distribution.
Parameter re-estimation: $z_{ik} = \frac{1}{z_i}\pi_k N(x \mid \mu_k, \Sigma_k)$, Expected no. of pixels in class k $N_k = \Sigma_i z_{ik}$

$$\mu_k = \frac{1}{N_k}\Sigma_i z_{ik}x_i, \qquad \Sigma_k = \frac{1}{N_k}\Sigma_i z_{ik}(x_i - \mu_k)(x_i - \mu_k)^T, \quad \pi_k = \frac{N_k}{N}$$

Hierarchical clustering: A nonprobabilistic approach to build hierarchy of groups from data similarities using distance matrix among the samples.
Hierarchical clustering algorithm: Start with a separate cluster for each element. Iteratively identify a pair of clusters closest together and merge two most similar clusters until all the clusters are merged together.
Distance between a pair of clusters: $d_{min}(C,C^*) = \min d(x,y)\ \forall$ x in C and y in C^*
$d_{avg}(C,C^*) = (1 / (|C^*||C|))\ \Sigma\ d(x,y)\ \forall$ x in C and y in C^*

Graph-based approaches: Connected components, cliques, etc are computed by forming graphs from the given input data

Clique Graphs: Here, each connected component is a clique. A clique has every vertex is connected to every other vertex.
Transforming into Clique Graphs: Add/remove some edges in a graph to transform into a clique graph. For minimum such operations, use Corrupted Cliques Problem.
Distance Graphs: If the distance between two vertices is below a chosen distance threshold θ, an edge is added between them.
Corrupted Cliques Problem: NP-hard problem. Approximate methods: PCC and CAST

Parallel Classification with Cores (PCC): Let $\{C_1, C_2, ..., C_k\}$ be clustering on subset S' of S, j ∈ S-S' and $N(j,C_i)$ be no. of edges from j to C_i. Assign j to cluster with max affinity. $Affinity(j,C_i) = N(j,C_i)/|C_i|$
Algorithm for PCC: PCC(S, G, k)
S -> set of n elements forming vertices of G, G -> distance graph and k -> no. of clusters.
Randomly select subsets S' and S'' from S and S-S' such that |S'| = log(log(n)) and |S''| = log(n). For all k partitions in S', get extended partitions in S through S' -> S'' -> (S - (S' U S'')), and choose the one with a minimum score to get the required Clique graph.
Time Complexity for PCC: $O(n^2 (log\ n)^{log_2 k})$

Cluster Affinity Search Technique (CAST): It is based on the distance of a feature i from a cluster C. The cluster is close if distance d(i, C) is less than the distance threshold θ.
d(i, C) = avg distance between feature i and all other features in C
CAST Algorithm: CAST(S, G, θ)
S -> set of elements, G -> distance graph, θ -> distance threshold
Get a cluster C corresponding to maximal degree vertex in G. For each close feature not in C or distant feature in C, add the nearest close feature i not in C and remove the farthest distant feature in C. Add C to partition P and remove its vertices from G. Repeat until S is empty and return P.

DBSCAN: Density-based spatial clustering of applications with noise
It is a non-parametric algorithm that groups together core samples of high-density points that are closely packed with many nearby neighbors and expand clusters from them. It works well for data that contains clusters of similar density.

Concepts challenging to comprehend:
None

Interesting and exciting concepts:
I found Corrupted Cliques Problem interesting as it was totally new to me and also an NP-hard problem. The two approximation methods were astute for finding a solution to the NP problem.

Concepts not understood:
None

Any novel idea of yours out of the lessons:
The Hierarchical Clustering algorithm provides better clustering results than the K-means algorithm and is easy to implement. But this algorithm seems to involve a lot of arbitrary decisions if the training set has mixed data types or the dataset is large. It might also make a wrong merging choice at an initial stage which can't be undone and hence give a poor outcome in the end. So, there is a need for an even better method for unsupervised learning. An approach could be that instead of finding clusters according to arbitrary distance, we can use a probabilistic model that describes the distribution of the dataset, and the data points are assigned to a cluster based on maximum likelihood estimation.
