Machine Learning (CS60050) – Weekly Report

Kaushal Banthia (19CS10039)

Week 8: 29th – 1st October, 2021

Topics Covered:

- Cluster Validity Indices and Stability Check Based Clustering
- Generalizing K – Means
- Expectation Maximization Algorithm (EM)
- Hierarchical Clustering
- Graph Based Approaches
- Clique Graphs
- DBSCAN (Density-based spatial clustering of applications with noise)

Summary (Topic Wise):

- Cluster Validity Indices and Stability Check Based Clustering

  - External indices using a reference partitioning information.

    - Normalized Mutual Information (NMI) $= \frac{2I(Y;C)}{H(Y)+H(C)}$,
      $$where\ I(Y;C) = H(Y) - H(Y|C)$$
    - Fraction of same pairs in same clusters (FM index)
    - Set matching measures (Finding matching partition pairs and maximal common coverage).

  - Internal indices by Looking at variance distribution, structure of clusters

    - Silhouette index. The higher, the better. It exists between $[-1,1]$
      $$s(x) = \frac{a(x) - b(x)}{max\big(a(x), b(x)\big)}$$
      $$where\ a(x) = \ the\ average\ distance\ of\ points\ within\ the\ cluster\ from\ x$$
      $$b(x) = Minimum\ average\ distance\ of\ points\ of\ other\ clusters\ from\ x$$

    - Calinski-Harabasz (CH) Index. The higher, the better.
      $$CH(K) = \frac{(J(1) - J(K))/(K-1)}{J(K)/(n-K)}, where\ J(i) = SSE\ with\ K = i$$

  - Repeated clustering should have similar partitioning for an appropriate K.
  - For that, we use Wang's method of cross-validation

    - Permute the input data c times.
    - Each time divide into three parts, S1, S2, and S3, such that |S1|=|S2|=m
    - Perform k-means on S1 and S2 and test on S3 for both the cases, to find the cluster numbers. Then compute the number of disagreement (i.e., a pair being in the same or different clusters).
    - Take the average over c observations.
    - Choose K minimizing average number of disagreements.

- Generalizing K – Means

  - $P(x) = \sum_{i=1}^{K} P(x|G_i) * P(G_i)$, where G_i defines the i[th] group or cluster and K is a hyper – parameter (denoting the number of components).
  - Mixture of Gaussians: Each cluster center is augmented by a covariance matrix, whose values are re – estimated from corresponding samples.
  - Mahalanobis distance function = $d(x, \mu_k; \sum_k = (x - \mu_k)^T * \sum_k^{-1}(x - \mu_k)$, *where* μ_k *is the cluster centre and* $\sum_k$ *is the covariance matrix*
  - This technique could be refined by computing probabilities of belongingness to a cluster. Parametric PDF = $p(x|\{\pi_k.\mu_k, \sum_k\}) = \sum_k \pi_k N(x|\mu_k, \sum_k)$

- Expectation Maximization Algorithm (EM)

1. Start with initial set: $\{\pi_k.\mu_k, \sum_k\}$
2. E – Step (Expectation Stage)
   a. Compute probability (z_{ik}) of x belonging to the k[th] Gaussian cluster.
   b. (OPTIONAL) Assign x to the m[th] cluster whose probability is maximum.
3. M – Step (Maximization Stage)
   a. Re – estimate parameters ($\{\pi_k.\mu_k, \sum_k\}$) from class distribution.

$$z_{ik} = \frac{1}{z_i}\pi_k N(x|\mu_k, \sum_k)$$

$$N_k = \sum_i z_{ik}$$

$$\mu_k = \frac{1}{N_k}\sum_i z_{ik} x_i$$

$$\sum_k = \frac{1}{N_k}\sum_i z_{ik}(x_i - \mu_k)(x_i - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

4. Iterate above 2 steps, till convergence.

- Hierarchical Clustering

  - Builds hierarchy of groups, usually using a bottom-up approach. Uses a distance matrix among the samples. Also, explicit feature representation may not be required, as it uses a non – probabilistic approach.
  - This clustering algorithm takes an n x n distance matrix d of pairwise distances between points as input.

```
1. Form n clusters each with 1 element.
2. Initialize a graph T with a vertex for each cluster.
3. while (number of clusters > 1)
       a) Find the 2 closest clusters C1 and C2 and merge them into C, having
          |C1|+|C2| elements.
       b) Compute the distance from C to all the other clusters.
       c) Add a new vertex C to T and connect to vertices C1 and C2.
       d) Remove rows and columns of d corresponding to C1 and C2.
       e) Add a row and column to d corresponding to the new cluster C.
4. Return T
```

  - Note that there are different ways of defining distances between clusters, which may lead to different clustering.
  - $d_{min}(C, C^*) = min\ d(x, y)\ \forall\ x\ in\ C\ and\ y\ in\ C^*$
  - Thus, the distance between two clusters is the smallest distance between any pair of their elements.
  - $d_{avg}(C, C^*) = \left(\frac{1}{|C^*||C|}\right)\sum d(x, y)\ \forall\ x\ in\ C\ and\ y\ in\ C^*$

- Thus, the distance between two clusters is the average distance between all pairs of their elements.

- Graph Based Approaches

  - Form a graph from the input data (may not be explicit).
  - Compute cliques, connected components, etc.

- Clique Graphs

  - A clique is a graph with every vertex connected to every other vertex.
  - A clique graph is a graph where each connected component is a clique.
  - An arbitrary graph can be transformed into a clique graph by adding or removing edges.
  - Corrupted Cliques Problem:

    - Input: A graph G
    - Output: The smallest number of additions and removals of edges that will transform G into a clique graph.

  - Distance Graphs: The feature vectors are represented as vertices in the graph.

    - We choose a distance threshold θ. If the distance between two vertices is below θ, then draw an edge between them.
    - The resulting graph may contain cliques. These cliques represent clusters of closely located data points

  - Although the Corrupted Cliques problem is NP-Hard, some heuristics exist to approximately solve it:
  - Two approximate methods:

    - Parallel Classification with Cores (PCC)

      1. Suppose S' is a subset of S.
      2. Let, $\{C1, C2, \ldots, Ck\}$ be a clustering on S'.
      3. Let $j \epsilon S - S'$ and $N(j, Ci)$ be number of edges from j to Ci
      4. $Affinity(j, Ci) = \frac{N(j,Ci)}{|Ci|}$
      5. Assign j to the cluster which has maximum affinity.

    - Algorithm for $PCC(S, G, k)$, where S: Set of n elements (feature vectors forming vertices of G), G: Distance graph and k: Number of clusters

      1. Randomly select S', a subset from S, and S'', a subset from $S - S'$, such that $|S'| = log(log(n))$ and $|S''| = log(n)$
      2. For all k partitions in S'
         a) Obtain extended partition in S by 2 stages of extensions i.e., $S' \rightarrow S'' \rightarrow (S - (S' \cup S''))$
         b) Choose the one which has minimum score, i.e., the number of edges required to be added or removed from G to get a Clique graph as per the partition.

    - Time Complexity of PCC:

      - Number of partitions in $S' = k^{|S'|} = k^{\log(\log(n))} = (\log(n))^{log_2(k)}$

- In each iteration $O(n^2)$ operations for extension and score computation.
- Thus, total time complexity = $O(n^2 \left(log(n)\right)^{log_2 k})$

- Cluster Affinity Search Technique (CAST)
  - It is a practical and a fast algorithm.
  - CAST is based on the notion of features close to cluster C or distant from cluster C.
  - Distance between feature i and cluster C = $d(i, C)$ = average distance between feature i and all other features in C
  - Gene i is close to cluster C, if $d(i, C) < \theta$ and distant otherwise.

- Algorithm for $CAST(S, G, \theta)$, where S: set of elements, G: distance graph and θ: distance threshold

```
1. P ← Ø
2. while (S ≠ Ø)
      • V ← vertex of maximal degree in the distance graph G.
      • C ← {v}
      • while (a close feature i not in C or distant feature
        i in C exists)
          • Find the nearest close feature i not in C and
            add it to C.
          • Remove the farthest distant feature i in C.
      • Add cluster C to partition P.
      • S ← S \ C
      • Remove vertices of cluster C from the distance graph
        G.
3. return P
```

- DBSCAN (Density-based spatial clustering of applications with noise)
  - No explicit computation of distance graph.
  - Grow regions of connected core points from a seed.
  - A neighbor, which is not a core point, is called a border point.

Concepts Challenging to Comprehend: None yet.

Interesting and Exciting Concepts: Hierarchical Clustering and Clique Graphs

Concepts not understood: None yet.

A novel idea: The Cluster Validity Indices (Internal) are dependent on distance metrics to calculate the indices. Since distance can be defined in various ways, like the Euclidean Distance, Manhattan Distance etc., the output could be different and thus prone to judgement error, unless the specific distance calculation method is known. Though nothing significant can be done when the ground truth is absent, but when the ground truth is present, we can approximate the cluster's ground truth by the mode of the ground truths of the individual data points. Then we can calculate the cluster error as a normal classification problem, where the true value is the ground truth of the individual points and the predicted value is the ground truth of the cluster.