Chapter 7

Cluster Analysis Advanced Concept

Outlines

- Introduction;
- Clustering Categorical Data;
- Scalable Data Clustering;
- High Dimensional Clustering;
- Semisupervised Clustering; and
- Human and Visually Supervised Clustering; and
- Applications.

1.Introduction

Clustering Aspects (Difficult Clustering Scenarios), (Advanced insights)

1- Categorical data clustering;
2- Scalable Clustering; and
3- High-Dimensional Clustering;

Because clustering is unsupervised problem, the quality of the clusters may be difficult to evaluate in many real scenarios (noise).

1- Semisupervised clustering;
2- Interactive and visual clustering; and
3-Ensemble Clustering.

2.Clustering Categorical Date

2.1. Representative-Based Algorithm

2.2. Hierarchical Algorithm

Agglomerative bottom-up algorithm have been successfully used for categorical data.

2.2.1. ROCK (RObust Clustering using linKs)

- It is based on bottom-up approach.
- Merge clusters based on Similarity , shared nearest neighbor metric.
- Algorithm Steps:
  1. Convert the categorical Data to binary representation.
  2. Apply Jaccard Coefficient.
  3. If the similarity of two data set is grater than a defined threshold θ , then these two data sets are neighbors and should merged.

2.Clustering Categorical Date

2.3. Probabilistic Algorithm

- Generating probability Distribution for each mixture component.
- Define E- and M- Steps, (EM) is the corresponding expectation-maximization approach.
- If k components of the mixture be denoted by $G_1, \ldots, G_k$, then the generative process for each point in the data set D uses the following two steps:
  1. Select a mixture component with prior probability a_i, where i belong to {1,…,k}.
  2. If the mth component of the mixture was selected in the first step, then generate data point from G_m .

One reasonable choice for the discrete probability distribution of G_m is to assume that the jth categorical value of ith attributes is independently generated by mixture component (cluster) m with probability p_{ijm}.

2.Clustering Categorical Date

2.3. Graph-Based Algorithm

- Because graph based methods are meta-algorithms the description of these algorithms remains virtually the same for categorical data as for numeric data.
- The only difference is in terms of how the edges and values on the similarity graph are connected.
- Determining the k-nearest neighbors of each data record, and subsequent assigning of similarity values to edges.
- One of the major advantages of graph-based algorithms is that they can be leveraged for virtually any kind of data type as long as similarity function can be defined on that data type.

3.Scalable Data Clustering

In many applications, the size of the data is very large and they cannot be stored in main memory.

3.1. CLARA and CLARANS

Are two scalable implementation of the k-medoids algorithm for clustering.

CLARA (Clustering LARg Applications)

- It is based on a particular instantiation of the k-medoids method know as Partitioning Around Medoids (PAM).
- All possible non-medoids are exchanged by medoids representative to improve the Obj Fun.
- A problem, is when each of the preselected samples does not include good choices of medoids.

CLARANS (Clustering Larg Applications based on RANdomized Search) :

- It works with the full data set for the clustering in order to avoid the problem with preselected samples, by exchanging between random medoids with random non-medoids iteratively.
- Finding local optimal is repeated after number of iterations, to get Maxlocal.
- Evaluating the Maxlocal locality.
- The best among these local optima is selected as the optimal solution.
- Dvantage of CLARANS over CLARA is that a greater diversity of the search space is explored.

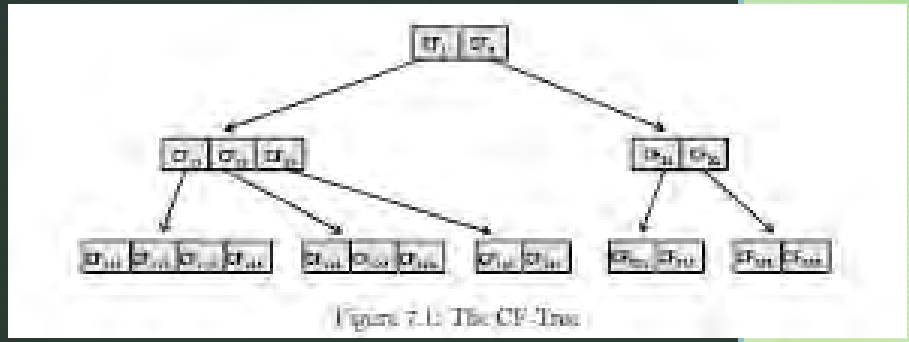
3.Scalable Data Clustering

3.2. BIRCH (Balanced Iterative Reducing and Clustering Hierarchies)

It is a combination of top-down hierarchical generalization and the k-means clustering.

Hierarichal data structure, known as CF-Tree.

Observe the summary of each cluster “Cluster feature (CF)”.



- Each cluster feature can represented as linear sum of the cluster features of the individual data points.
- The cluster features can used to compute useful properties of a cluster.
- Each leaf node in the CF-Tree has a diameter threshold T.
- Low T: will result in a larger number of fine-grained clusters.
- Large T: used for large data sets, to balance the greater need for memory.
- BRICH Algorithm is very fast, because the basic approach requires only one scan over the data, and each insertion is an efficient operation.

3.Scalable Data Clustering

3.3. CURE (Clustering Using Representatives)

It is a bottom-up agglomerative approach for clustering and can discover clusters of arbitrary shapes.

- Set of representatives.

- 1st representative: choose data point that is farthest form the center of the cluster.
- 2nd representative: it should be farthest to the first.
- 3rd representative: is chosen to be the one that has the largest distance to the closet of two representatives , and so on.
- The rth representative is a data point that has the largest distance to the closest of the current set of (r-1) representative and then shrunk toward the cluster center.

- The clusters are merged using agglomerative bottom-up approach.
- Minimum distance between any pair of representative data point is used.

4.High-Dimensional Clustering

4.1. CLIQUE (CLusteirng In QUEst)

Subspace clustering method.

- Close to pattern mining; and
- Number of subspace clusters can be very large, where it depends on the density threshold.

Algorithm *CLIQUE*(Data: $\mathcal{D}$, Ranges: p, Density: τ)
begin
  Discretize each dimension of data set $\mathcal{D}$ into p ranges;
  Determine dense combinations of grid cells at minimum support τ
    using any frequent pattern mining algorithm;
  Create graph in which dense grid combinations are
    connected if they are adjacent;
  Determine connected components of graph;
  return (point set, subspace) pair for each connected component;
end

Figure 7.3: The *CLIQUE* algorithm

4.High-Dimensional Clustering

4.2. PROCLUS (PROjected CLUStering)

- It uses a medoid-bases approach to clustering.
- The algorithm proceeds in three phases: an initialization phase, an iterative phase, and a cluster refinement phase.

Algorithm *PROCLUS*(Database: $\mathcal{D}$, Clusters: k, Dimensions: l)
begin
  Select candidate medoids $M \subseteq \mathcal{D}$ with a farthest distance approach;
  S = Random subset of M of size k;
  $BestObjective = \infty$;
  repeat
    Compute dimensions (subspace) associated with each medoid in S;
    Assign points in $\mathcal{D}$ to closest medoids in S using projected distance;
    $CurrentObjective$ = Mean projected distance of points to cluster centroids;
    if ($CurrentObjective < BestObjective$) **then begin**
      $S_{best} = S$;
      $BestObjective = CurrentObjective$;
    end;
    Recompute S by replacing bad medoids in S_{best} with random points from M;
  until termination criterion;
  Assign data points to medoids in S_{best} using refined subspace computations;
  return all cluster-subspace pairs;
end

4.High-Dimensional Clustering

4.3. ORCLUS (arbitrary Oriented projected CLUStering)

- Combination of Hierarchical and k-means clustering in conjunction with subspace refinement.

- The algorithm consist of a number of iterations.
- In each iteration, the merging operation is alternated with a k-mean and projected distance.
- α factor to reduce the number of clusters. $\alpha = 0.5$
- β is used to reduce the cluster dimensionality.

- The overall procedure using subspace re-computation, and merging in each iteration.

4.High-Dimensional Clustering

4.3. ORCLUS (arbitrary Oriented projected CLUStering)

Algorithm *ORCLUS*(Data: $\mathcal{D}$, Clusters: k, Dimensions: l)
begin
  Sample set S of $k_0 > k$ points from $\mathcal{D}$;
  $k_c = k_0$; $l_c = d$;
  Set each $\mathcal{E}_i$ to the full data dimensionality;
  $\alpha = 0.5$; $\beta = e^{-\log(d/l) \cdot \log(1/\alpha)/\log(k_0/k)}$;
  while $(k_c > k)$ **do**
  begin
    Assign each data point in $\mathcal{D}$ to closest seed in S using
      projected distance in $\mathcal{E}_i$ to create $\mathcal{C}_i$;
    Re-center each seed in S to centroid of cluster $\mathcal{C}_i$;
    Use PCA to determine subspace $\mathcal{E}_i$ associated with $\mathcal{C}_i$ by selecting
      smallest l_c eigenvectors of covariance matrix of $\mathcal{C}_i$;
    $k_c = \max\{k, k_c \cdot \alpha\}$; $l_c = \max\{l, l_c \cdot \beta\}$;
    Repeatedly merge clusters to reduce number of clusters to
      the new reduced value of k_c;
  end;
  Perform final assignment pass of points to clusters;
  return cluster-subspace pairs $(\mathcal{C}_i, \mathcal{E}_i)$ for each $i \in \{1 \ldots k\}$;
end

5. Semisupervised Clustering

Relies on external application-specific criteria to guide the clustering process.

5.1. pointwise supervision

Labels are associated with individual data points and provide information about the category (or cluster) of the object. Related to data classification.

5.2. pairwise supervision

Constraint Clustering.

"Must-link" and "cannot-link" constraints are provided for the individual data points.

These constraints provide information about cases where pairs of objects are allowed to be in the same cluster OR are forbidden to be in the same cluster, respectively.

A,B, and C data points such that (A,B) and (A,C) are both "must-link" pairs. whereas (B,C) is a "cannot-link" pair.

6. Human and Visually Supervised Clustering

6.1. Modification of Existing Clustering Algorithm

1. Modification to k-means and related methods.
2. Modification to hierarchical methods.

6.2. Visual Clustering

It is a combination of the computational ability of a computer and the intuitive feedback of the user.

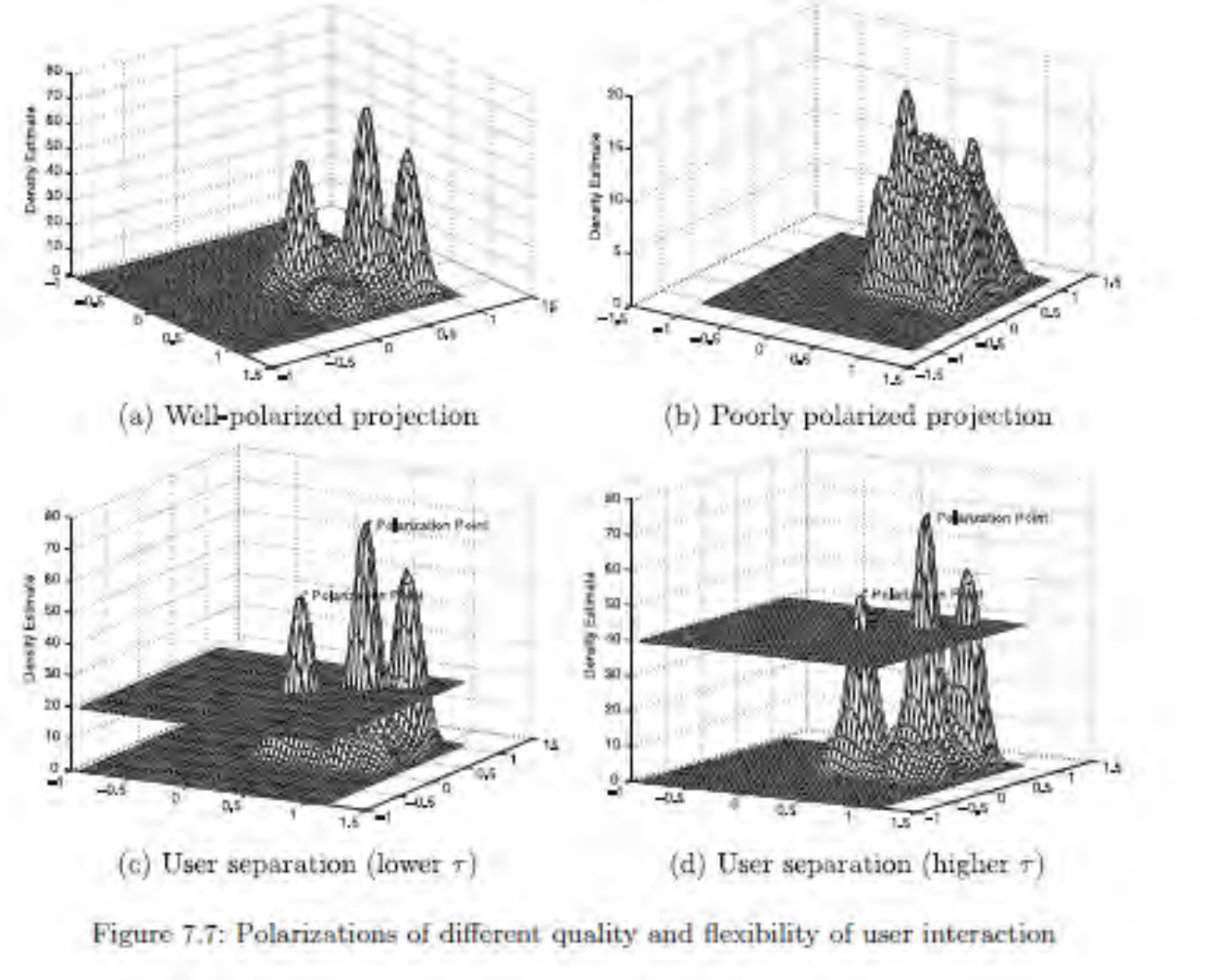
IPCLUS (Interactive Project CLUStering algorithm).

Algorithm *IPCLUS*(Data Set: $\mathcal{D}$, Polarization Points: k)
begin
  while not(termination_criterion) **do**
  begin
    Randomly sample k points $\overline{Y_1} \ldots \overline{Y_k}$ from $\mathcal{D}$;
    Compute 2-dimensional subspace $\mathcal{E}$ polarized around $\overline{Y_1} \ldots \overline{Y_k}$;
    Generate density profile in $\mathcal{E}$ and present to user;
    Record membership statistics of clusters based on
        user-specified density-based feedback;
  end;
  return consensus clusters from membership statistics;
end

Figure 7.6: The *IPCLUS* algorithm

6. Human and Visually Supervised Clustering

6.2. Visual Clustering



(a) Well-polarized projection

(b) Poorly polarized projection

(c) User separation (lower τ)

(d) User separation (higher τ)

Figure 7.7: Polarizations of different quality and flexibility of user interaction

Applications

- Applications to other Data Mining Problems;
- Data Summarization;
- Outlier Analysis;
- Classification;
- Dimensionality Reduction;
- Similarity Search and Indexing;
- Customer Segmentation and Collaborative Filtering;
- Text Application;
- Multimedia Applications ;
- Temporal and Sequence Applications; and
- Social Network Analysis.

Thank You