Subspace Clustering

Hemant Banke, Borish Jha

Indian Statistical Institute

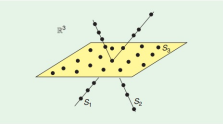
March 22, 2025

Introduction and Background

Motivation

- Points in the dataset are equidistant from each other in high dimensions.
- The data points could be drawn from multiple subspaces.
- A video sequence could contain several moving objects, and different subspaces might be needed to describe the motion of different objects in the scene.
- Therefore, there is a need to simultaneously cluster the data into multiple subspaces and find the subset of features leading to each subspace. This is the problem of subspace clustering.

The Subspace Clustering Problem



Let $\left\{x_j \in \mathbb{R}^D\right\}_{j=1}^N$ be a given set of points drawn from an unknown union of $n \geq 1$ linear or affine subspaces $\{S_i\}_{i=1}^n$ of unknown dimensions $d_i = \dim\left(S_i\right), 0 < d_i < D, i = 1, \ldots, n$. The subspaces can be described as $S_i = \left\{x \in \mathbb{R}^D : \boldsymbol{x} = \boldsymbol{\mu}_i + U_i y\right\}, \quad i = 1, \ldots, n,$ where $\boldsymbol{\mu}_i \in \mathbb{R}^D$ is an arbitrary point in subspace S_i that can be chosen as $\mu_i = \mathbf{0}$ for linear subspaces, $U_i \in \mathbb{R}^{D \times d_i}$ is a basis for subspace S_i, and $y \in \mathbb{R}^{d_i}$ is a low-dimensional representation for point x. The goal of subspace clustering is to find the number of subspaces n, their dimensions $\{d_i\}_{i=1}^n$, the subspace bases $\{U_i\}_{i=1}^n$, the points $\{\boldsymbol{\mu}_i\}_{i=1}^n$, and the segmentation of the points according to the subspaces.

Types

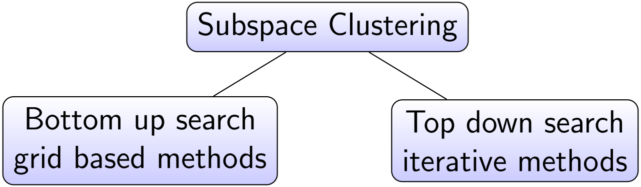


Figure: Types based on the search strategy

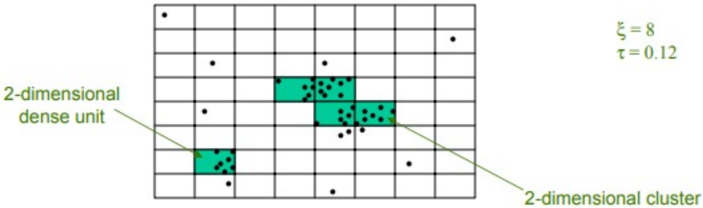
Bottom up search

- Starts with 1-dimensional subspaces and merges lower dimensional subspaces to compute higher dimensional ones.
- A unit is considered dense if fraction of all data points in it exceeds a predetermined density threshold.
- If there are dense units in k dimensions, then all $(k-1)$ dimensional projections are also dense. Candidate subspaces in k dimensions can then be formed using only the dense units in $(k-1)$ dimensions.

CLIQUE I

- Each dimension is partitioned into ϵ equi-sized intervals called units.
- Having found dense units in $(k-1)$ dimensions, dense units in k dimensions are found by considering only those whose every projected $(k-1)$ dimensional unit is dense.
- Adjacent dense grids are then combined to form clusters.
- The points that don't fall into dense grids are considered outliers.

CLIQUE II



Bottom up search II

- Bottom-up approach often leads to overlapping clusters.
- Obtaining meaningful results is dependent on the proper tuning of the grid size and the density threshold parameters.
- Some methods allow for adaptive grid generation that stabilize the results across a range of density thresholds.

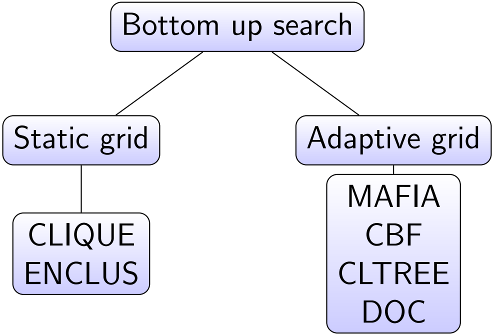
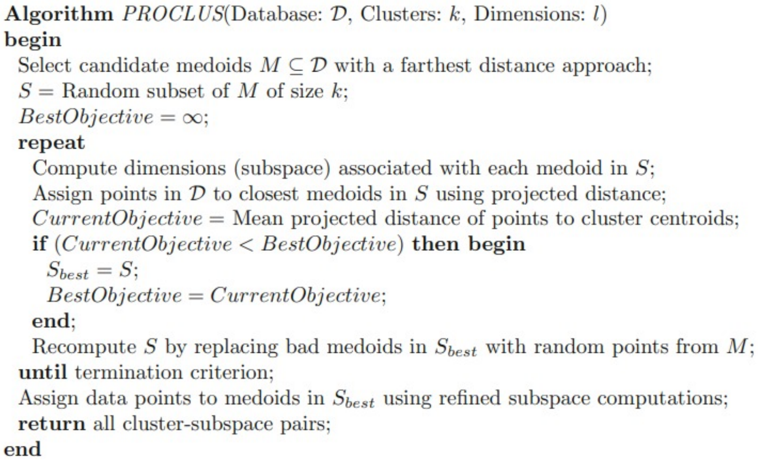


Figure: Types based on type of grid used

Top down search

- The search starts in the full d-dimensional space and iteratively learns for each point or each cluster the correct subspace.
- Dimensions are weighted for each cluster or instance to indicate the relevant subspace.

PROCLUS I



```
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
```

PROCLUS II

- The sum of the total number of dimensions associated with the different medoids must be equal to kl. An additional constraint is that the number of dimensions associated with a medoid must be at least 2.
- The medoid of the cluster with the least number of points is bad. In addition, the medoid of any cluster with less than $(n/k) \cdot minDeviation$ points is bad, where $minDeviation$ is a input parameter smaller than 1.

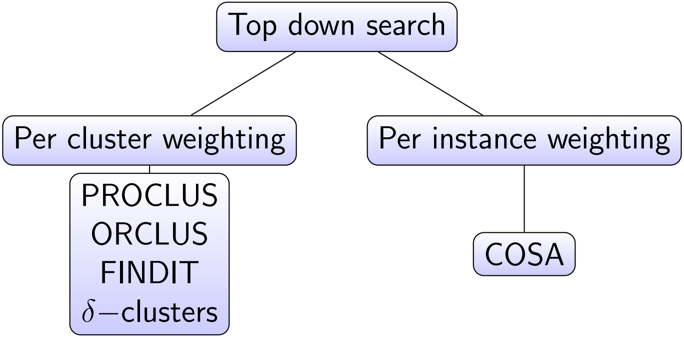


Figure: Types based on type of grid used

- Per cluster weighting:
  - Learns the subspace of a **cluster** starting with full dimensional clusters.
  - Iteratively refines the cluster memberships of points and the subspaces of the cluster.
- Per instance weighting:
  - Learns for **each point** its subspace preference in the full dimensional data space.
  - The subspace preference specifies the subspace in which each point clusters best.
  - Merges points having similar subspace preferences to generate clusters.

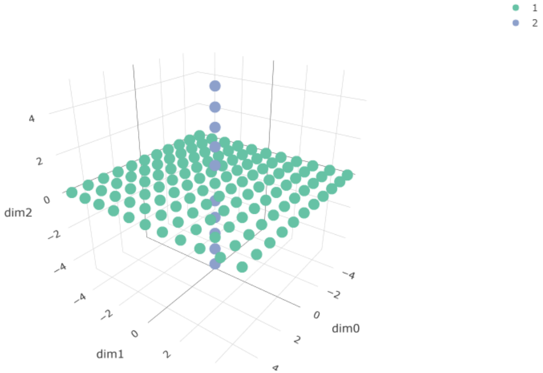
Top down search II

- Clusters formed are partitions of the dataset, many methods allowing for an additional group of outliers.
- Many use sampling to improve performance. Often the most critical parameters for top-down algorithms is the number of clusters and the size of the subspaces, which are often very difficult to determine ahead of time.

Examples

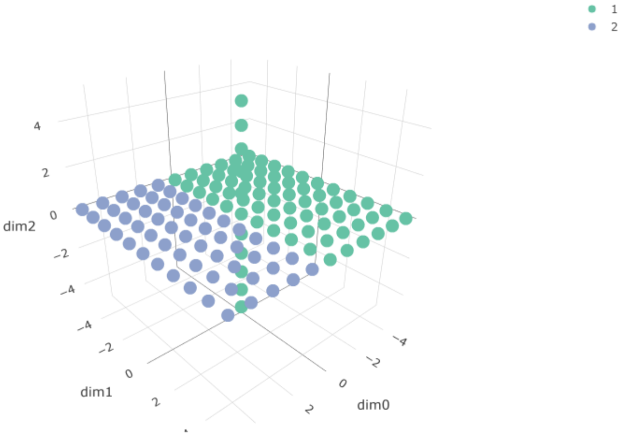
Example 1

True Cluster 1 : Points on the x-y plane
True Cluster 2 : Points along the z-axis



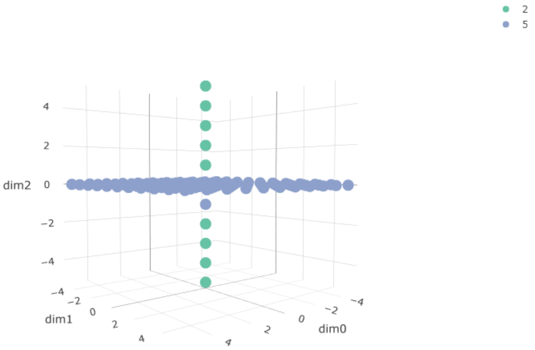
Example 1 - K-Means

Accuracy $= 0.9236641$



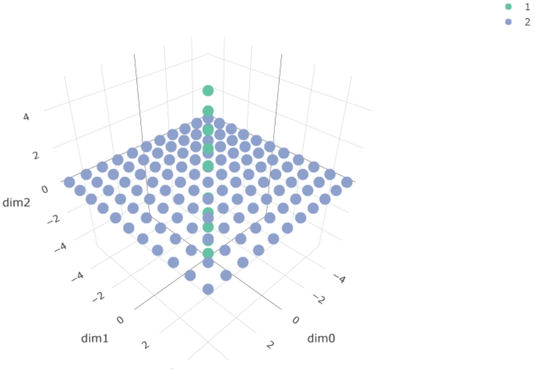
Example 1 - CLIQUE($x_i = 5, \tau = 0.14$)

Accuracy $= 0.9923664$



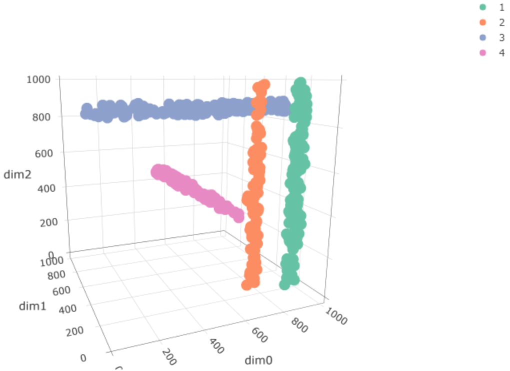
Example 1 - ProClus($k = 2, l = 1.5$)

Accuracy $= 1$

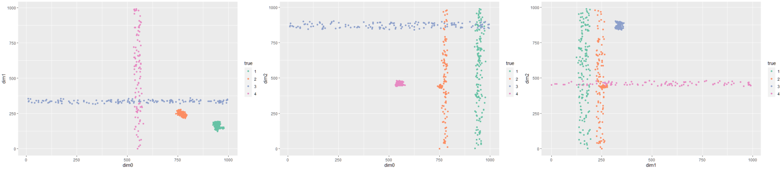


Example 2

Points in the 4 clusters are distributed along lines parallel to axis'.



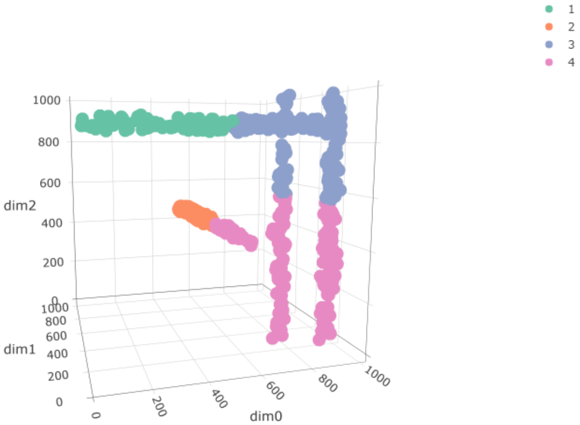
Example 2



(a) dim0 vs dim1 (b) dim0 vs dim2 (c) dim1 vs dim2

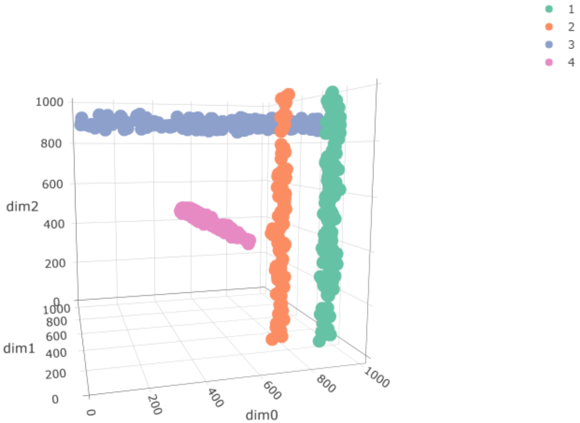
Example 2 - K-Means

Accuracy = 0.615245



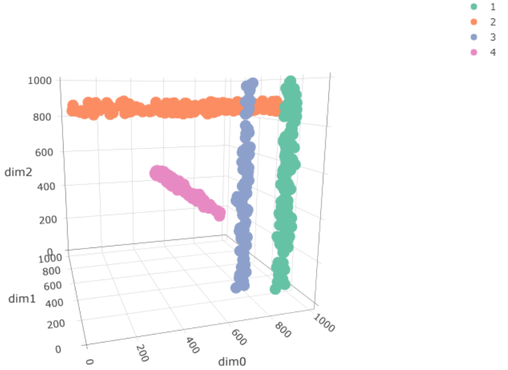
Example 2 - Single Linkage H.C.

Accuracy $= 1$



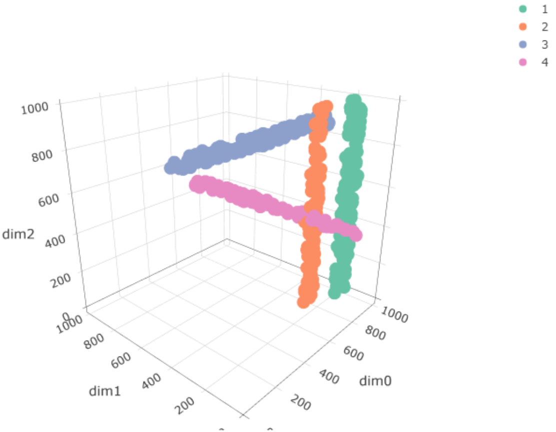
Example 2 - ProClus($k = 4, l = 2$)

Accuracy $= 1$



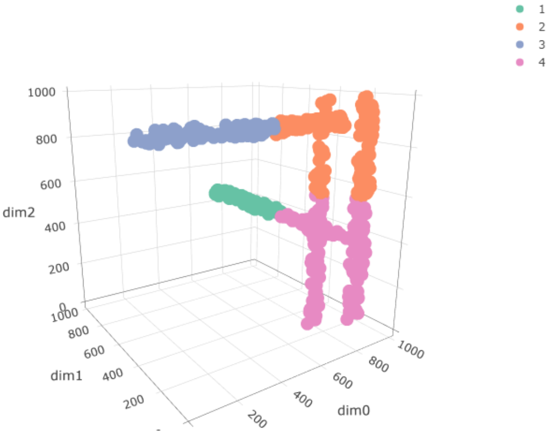
Example 3

Now we introduce noise in the data by touching cluster 4 with cluster 2.



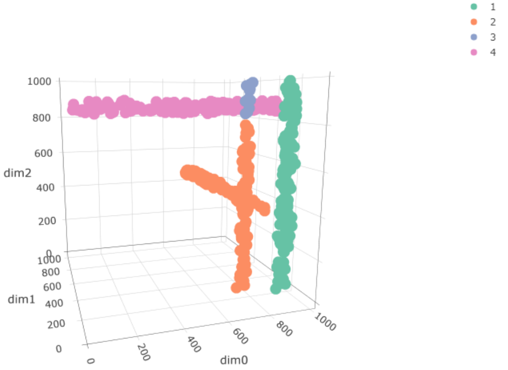
Example 3 - K-Means

Accuracy $= 0.6061706$



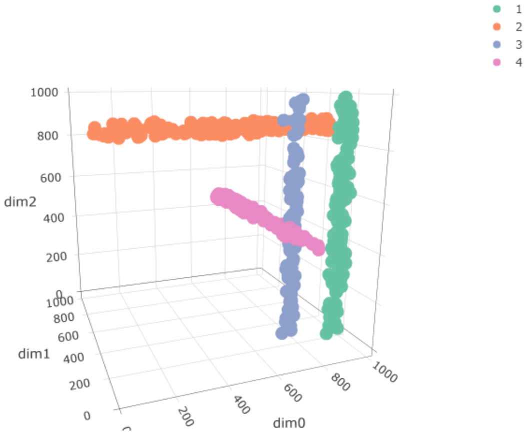
Example 3 - Single Linkage H.C.

Accuracy $=0.8185118$



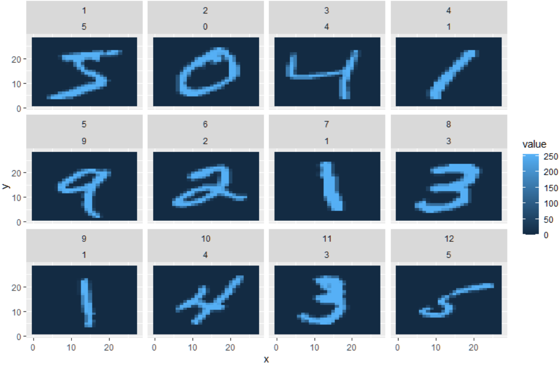
Example 3 - ProClus($k = 4, l = 2$)

Accuracy $= 0.9201452$



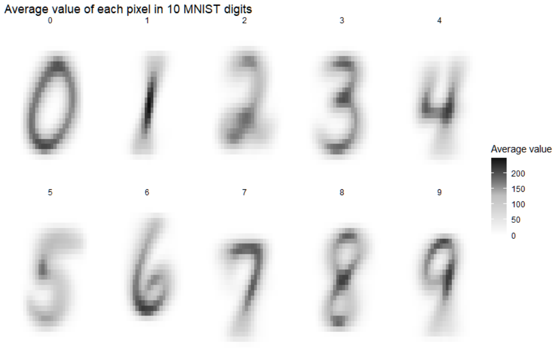
MNIST Dataset

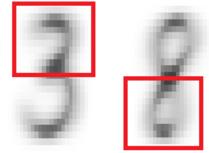
We can show the effectiveness of subspace clustering in MNIST Handwritten Digits Dataset. Each image is 28x28 wide (i.e. 784 dimensions)



MNIST Dataset

Intuitively, considering and characterizing small segments of picture can lead to a discriminator of the handwritten digits.





MNIST Dataset

Using this instinct, performing $PROCLUS(k = 10, l = 100)$ gives Accuracy **0.631**.

To compare this with k-Means, we first performed PCA to reduce dimensions from 784 to 200 (explains about 90% variance). The Accuracy obtained was **0.5106** .

Other Applications

Subspace clustering can be leveraged to uncover complex relationships found in data of DNA Micro-arrays in identification and characterization of disease sub types.