Constrained Clustering

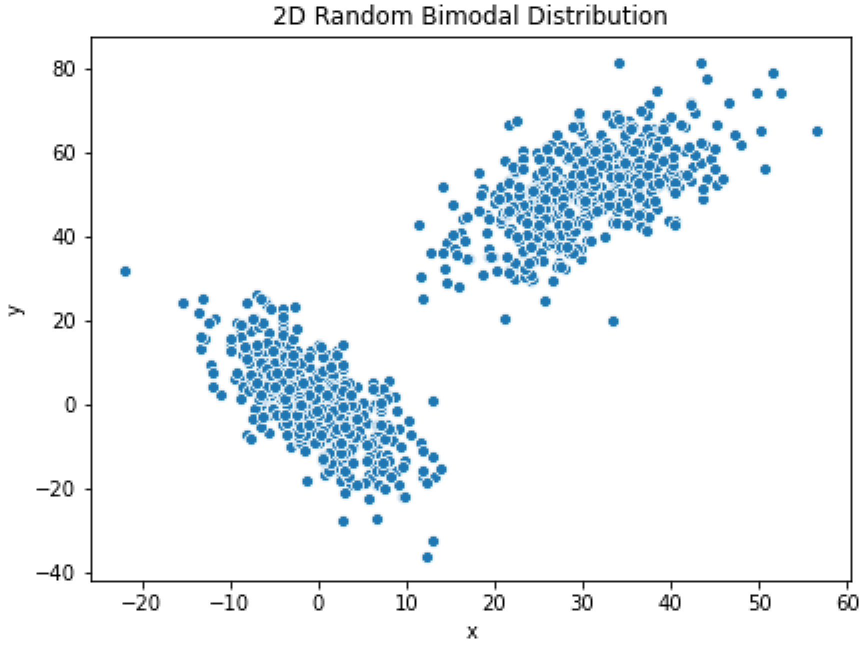
Krishna Teja

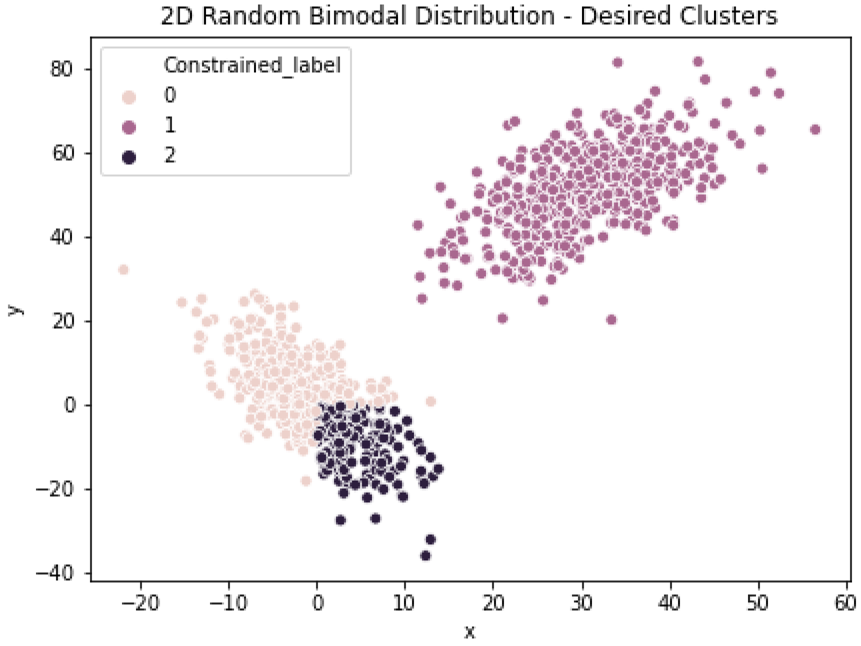
Problem Statement

- How to enforce known rules/domain knowledge into the unsupervised learning and convert it to a semi-supervised learning?
    - Say we know some conditions about data points,
    - some information which could typically be used for the betterment of unsupervised clustering?

Random Data Generation

1. Generate 2D Multivariate normal data
2. Simple start - 2 clusters, easily separable
3. Establish GroundTruth based on the desired conditions (x>0 & y<0)
4. Use Ground truth for constraint generation or evaluate Performance Metrics





Metrics

- NMI - Normalised Mutual Information
- Rand Score - proxy for Accuracy

NMI Explained

- NMI tells about the reduction in the entropy of the class labels we get if we know the cluster labels

- Normalized Mutual Information:

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{[H(Y) + H(C)]}$$

where,

1) Y = class labels
2) C = cluster labels
3) H(.) = Entropy
4) I(Y;C) = Mutual Information b/w Y and C

Note: All logs are base-2.

NMI Explained (contd.)

I(Y;C)= Mutual Information

- Mutual information is given as:
  - $I(Y;C) = H(Y) - H(Y|C)$
  - We already know H(Y)
  - H(Y|C) is the entropy of class labels within each cluster, how do we calculate this??

Python to the rescue

- Sklearn - **normalized_mutual_info_score**
- NMI - [0,1], higher the better

Existing Literature

- 71 papers proposing algorithms

Category	Method
k-Means	COP-COBWEB (Wagstaff and Cardie, 2000) COP-KMeans (Wagstaff et al, 2001) Seed-KMeans (Basu et al, 2002) Constrained-KMeans (Basu et al, 2002) ICOP-KMeans (Tan et al, 2010) Sequenced Assignment COP-KMeans (Rutayisire et al, 2011) MLC-KMeans (Huang et al, 2008) SCREEN (Tang et al, 2007) GA Dispersion & Impurity (Demiriz et al, 1999) CVQE (Davidson and Ravi, 2005) LCVQE (Pelleg and Baras, 2007) PCK-Means (Basu et al, 2004b) Lagrangian Relaxation (Ganji et al, 2016) Tabu Search (Hiep et al, 2016) Fuzzy CMeans (Grira et al, 2006) Non-Negative Matrix Factorisation (Li et al, 2007) Mathematical Program (Ng, 2000) Minimal Capacity Constraints (Bradley et al, 2000) Balanced Clustering (Banerjee and Ghosh, 2006) Minimal Size (Demiriz et al, 2008) Minimal Size & Balanced Clustering (Ge et al, 2007)
Metric Learning	Euclidean (Klein et al, 2002) Mahanalobis (Bar-Hillel et al, 2003, 2005; Xing et al, 2002) Kullback-Leibler Divergence (Cohn et al, 2003) String-Edit Distance (Bilenko and Mooney, 2003) LRML (Hoi et al, 2008, 2010) Partially Observed Constraints (Yi et al, 2012)
k-Means & Metric Learning	MPCK-Means (Bilenko et al, 2004) HMRF-KMeans (Basu et al, 2004b) Semi-Supervised Kernel k-Means (Kulis et al, 2005, 2009) CLWC (Cheng et al, 2008)
Spectral Graph Theory	Adjacency Matrix Modification (Kamvar et al, 2003) Out-Of-Sample Adjacency Matrix Modification (Alzate and Suykens, 2009) CSP (Wang and Davidson, 2010a; Wang et al, 2014) Constraint Satisfaction Lower Bound (Wang et al, 2010) Inconsistent Constraints (Rangapuram and Hein, 2012) Logical Constraint Combinations (Zhi et al, 2013) Distance Modification (Anand and Reddy, 2011) Constraint Propagation Binary Class (Lu and Carreira-Perpiñán, 2008) Constraint Propagation Multi-Class (Lu and Ip, 2010; Chen and Feng, 2012; Ding et al, 2013) Kernel Matrix Learning (Zhang and Ando, 2006; Hoi et al, 2007; Li and Ding, 2008; Li and Liu, 2009) Guaranteed Quality Clustering (Cucuringu et al, 2016)
Ensemble Clustering	SCEV (Iqbal et al, 2012) Consensus Function (Al-Razgan and Domeniconi, 2009; Xiao et al, 2016; Dimitriadou et al, 2002)
Collaborative Clustering	SAMARAH (Forestier et al, 2010a) Penta-Training (Domeniconi and Al-Razgan, 2008)
Declarative Approaches	SAT (Davidson et al, 2010) CP (Dao et al, 2013, 2016, 2017; Guns et al, 2016) ILP Column Generation (Merle et al, 1999; Aloise et al, 2012; Babaki et al, 2014) Restricted Cluster Candidates (Mueller and Kramer, 2010; Ouali et al, 2016)
Miscellaneous	Constrained EM (Shental et al, 2013) Evolutionary Algorithm (Handl and Knowles, 2006) Random Forest (Zhu et al, 2016)

Interesting Algorithms with references

1. Cop-Kmeans (Constrained KMeans) [1]
2. PCKMeans (Pairwise Constrained KMeans) & Others [2]
3. PreIdentify and Kmeans

Implemented Algorithms

1. Cop-Kmeans (Constrained KMeans)
2. PreIdentify and Kmeans

Cop-KMeans

Table 1. Constrained K-means Algorithm

COP-KMEANS(data set D, must-link constraints $Con_= \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$)

1. Let $C_1 \ldots C_k$ be the initial cluster centers.

2. For each point d_i in D, assign it to the closest cluster C_j **such that** VIOLATE-CONSTRAINTS(d_i, C_j, $Con_=$, $Con_{\neq}$) **is false. If no such cluster exists, fail (return {})**.

3. For each cluster C_i, update its center by averaging all of the points d_j that have been assigned to it.

4. Iterate between (2) and (3) until convergence.

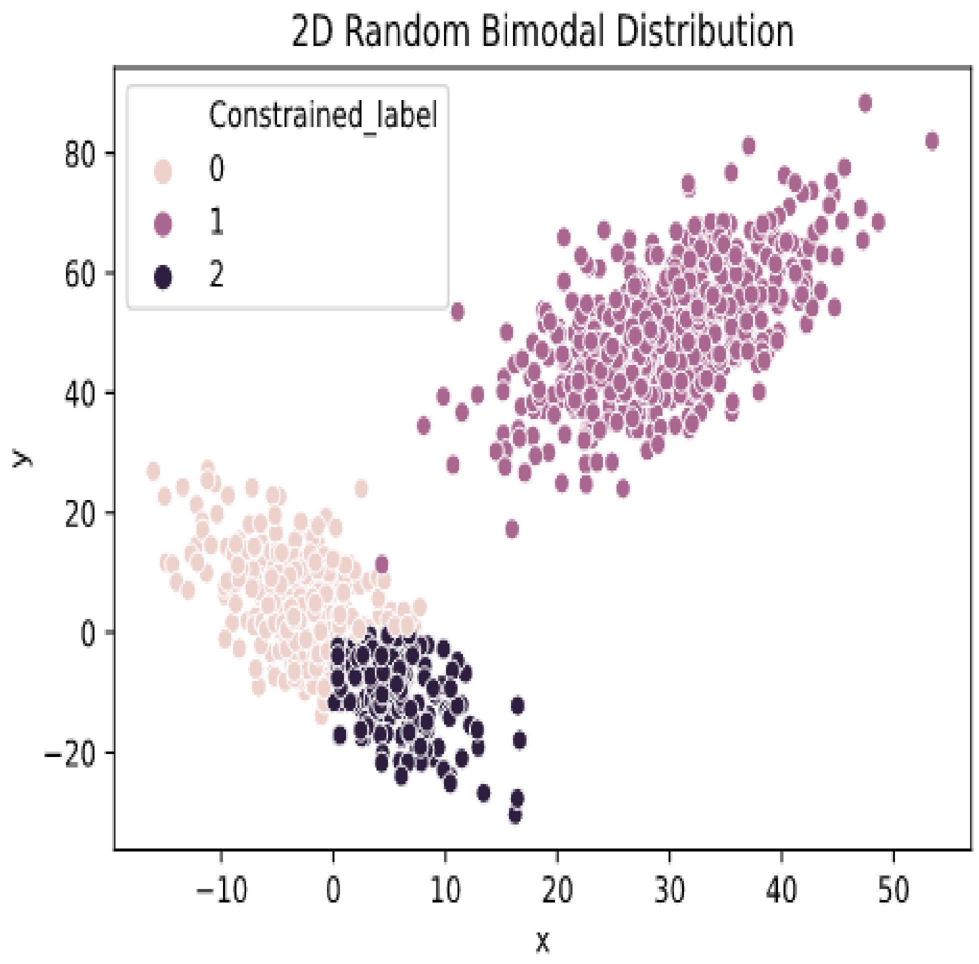
5. Return $\{C_1 \ldots C_k\}$.

VIOLATE-CONSTRAINTS(data point d, cluster C, must-link constraints $Con_= \subseteq D \times D$, cannot-link constraints $Con_{\neq} \subseteq D \times D$)

1. For each $(d, d_=) \in Con_=$: If $d_= \notin C$, return true.

2. For each $(d, d_{\neq}) \in Con_{\neq}$: If $d_{\neq} \in C$, return true.

3. Otherwise, return false.



Cop-KMeans - Implemented Algorithms(contd.)

Steps Involved :

1. Generate all Must-Link and Cannot-Link constraints from Ground Truth of the data
2. ML - datapoints in one cluster
3. CL - datapoints from different cluster
4. When the whole GT is known, every combination can be achieved
5. Transitive closure should be oberved when making constraints
6. In reality, we might know less than 5% of the GT
7. So we take these available info from GT(1%) and make them into constraints
8. Run the algorithm as mentioned where constraints are not violated along with finding the closest cluster

Data & Constraints Information

Data :

- 1000 records
- 2 dimensions
- 3 Labels

ML & CL - 499500

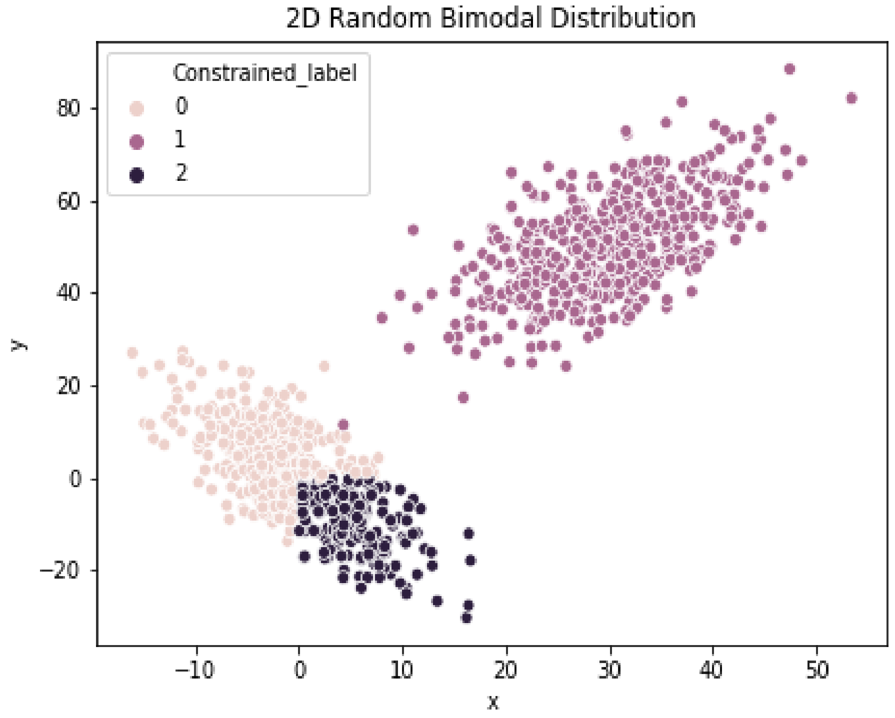
- 1 % - 4995
- 0.1 % - 499.5
- 0.01 % - 49.95

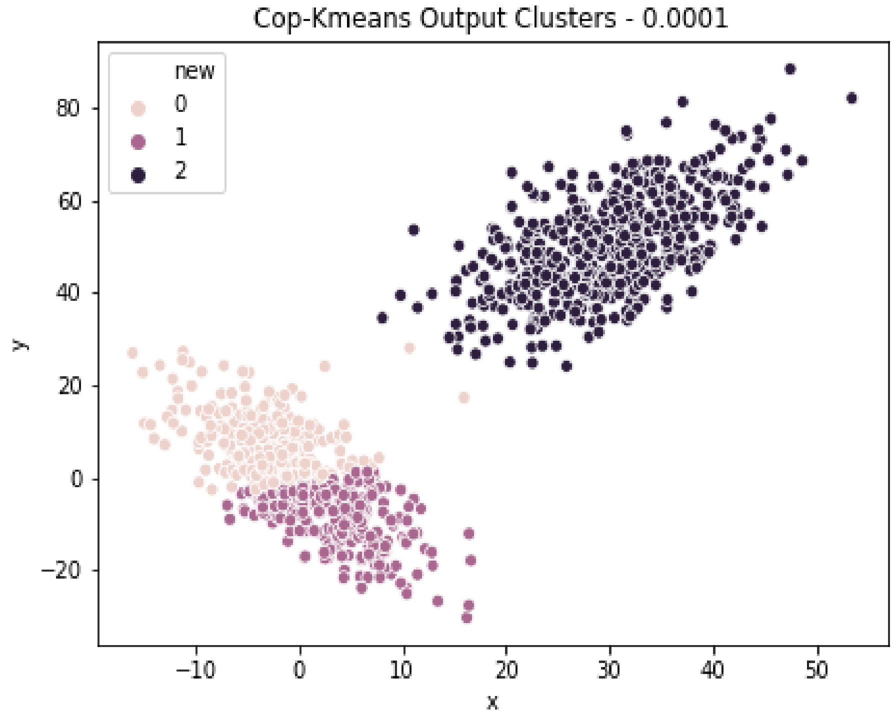
PreIdentify-KMeans - Implemented Algorithms(contd.)

Steps Involved :

1. Pre-label all the data points with the desired cluster tag
2. Find the centroid of that specific cluster
3. Find the remaning centroids on all of the remaining data based on distance
4. Converge when the centroids do not move - as usual

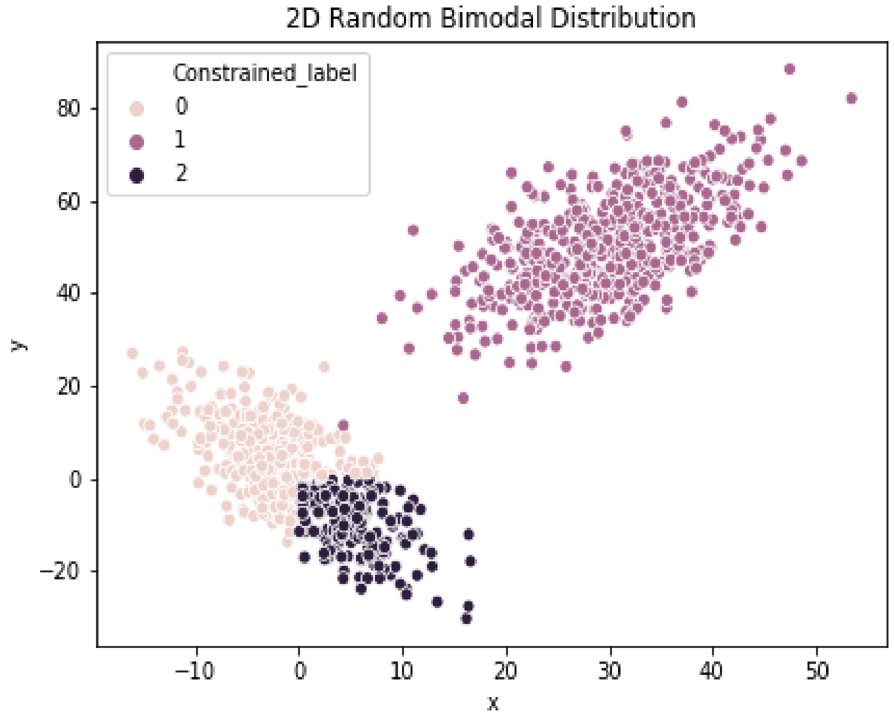
Results : Cop-Kmeans - 0.01 % constraints

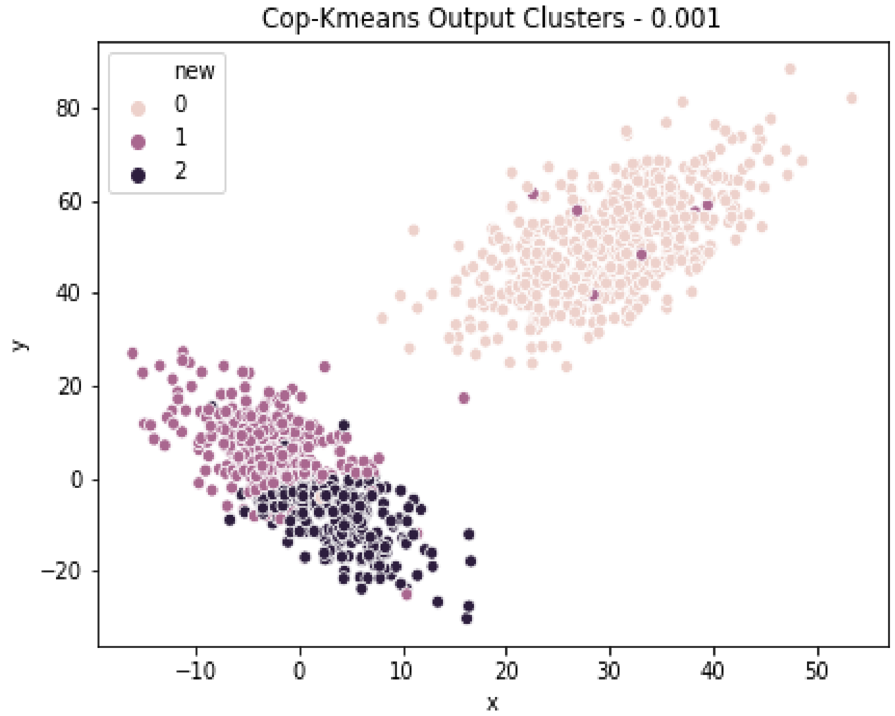




- NMI ~ 0.84

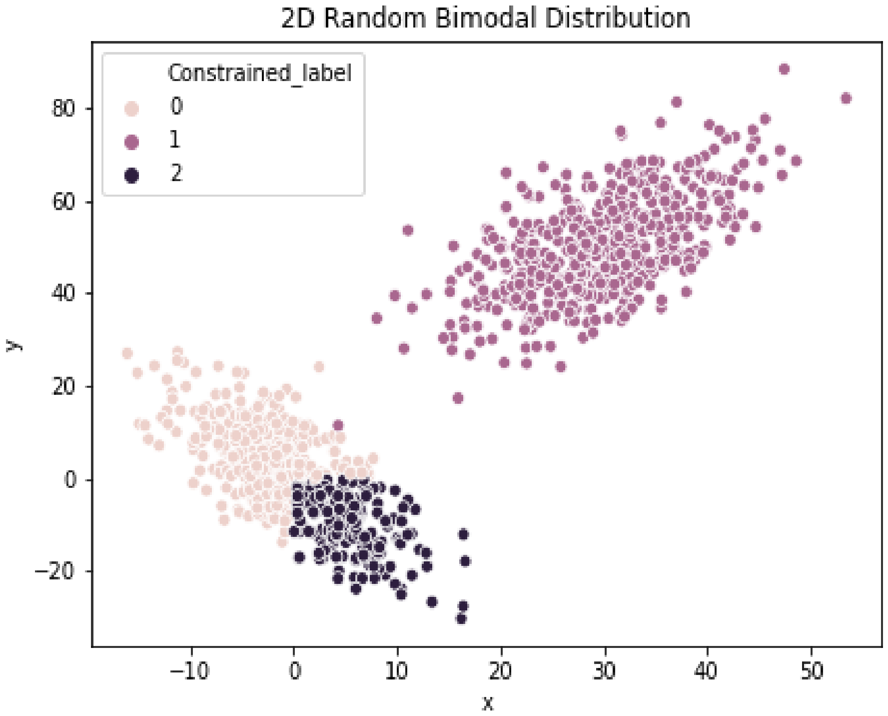
Results : Cop-Kmeans - 0.1 % constraints

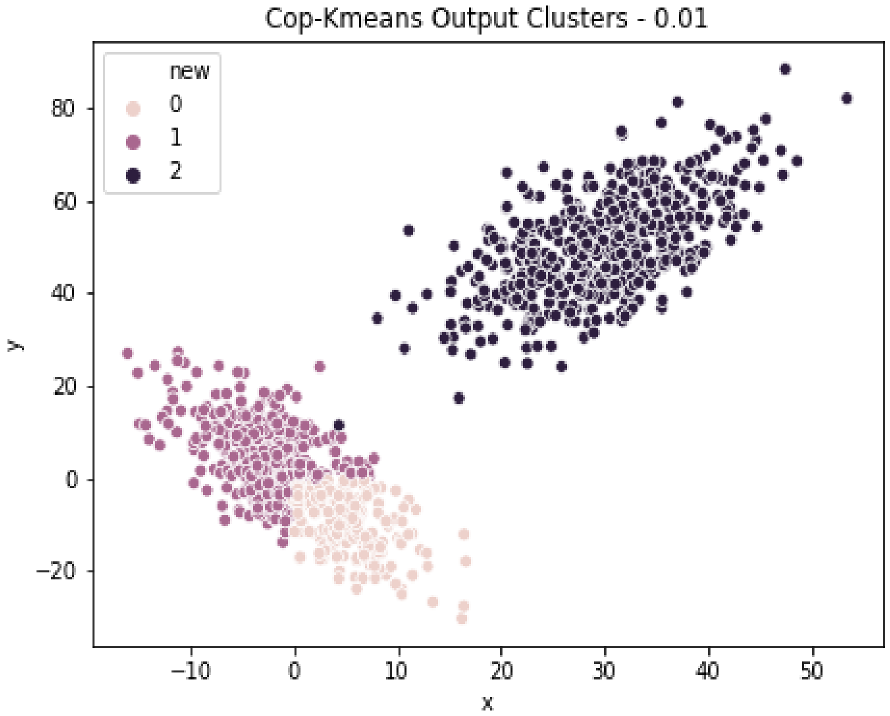




- NMI ~ 0.8

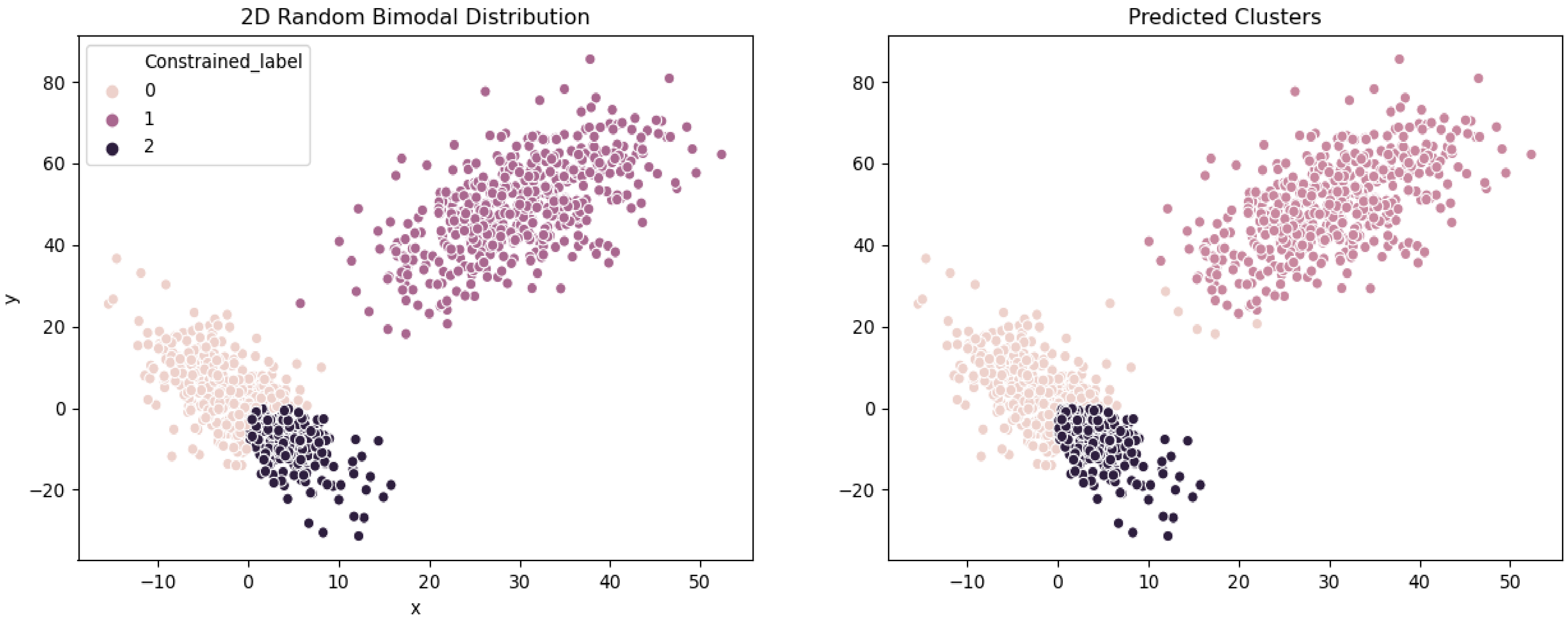
Results : Cop-Kmeans - 1 % constraints





- NMI ~ 1.0

Results : PreIdentify-Kmeans



- NMI = 0.782

Limitations

Cop-Kmeans

1. Cop-KMeans is time consuming
2. Sometimes, does not converge after processing for a long time - due to impractical constraints - need for correct domain knowledge

PreIdentify-Kmeans

1. Pre-Idenifying the data points with their labels is exactly not part of unsupervised/semi-supervised for that cluster
2. There is no clustering happening in the selected portion, just finding a centroid
3. Always need labels for the special cluster

References:

- https://github.com/Behrouz-Babaki/COP-Kmeans
- https://en.wikipedia.org/wiki/Mutual_information
- COP-Kmeans
- Constrined CLustering - PCKS - Size constraints

Topics:

- KL-Divergence
- Jensen-Shannon Divergence
- Jensens Inequality
- ELBO
- Projection Gradient

Thank you

https://github.com/krishnatejak2/customKmeans