TCD Statisitics Seminar Series
Clustering of Big Data with Mixed Features

November 11, 2020

Joshua Tobin & Mimi Zhang
Department of Statistics, Trinity College Dublin

Outline

Presentation will have three sections.

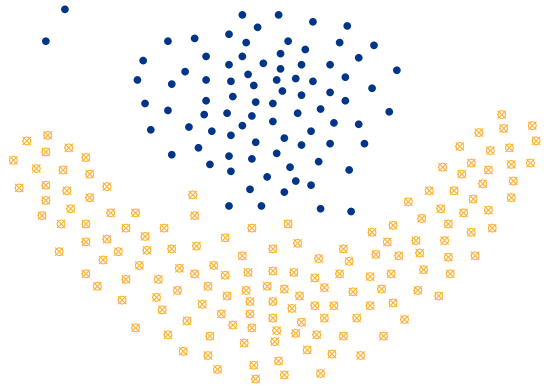
1. Introduction and motivation of the paper (Introduction)
2. Description of the CPF method (CPF)
   - Introduce distance metric for mixed data
   - Component-wise peak-finding (CPF)
   - Automatic center selection method
3. Brief experimental study detailing interesting features of CPF (Experimental Study)

Introduction - Motivation

Unsupervised Learning is a central problem in data science

Vital in its own right but also with applications in other learning domains & methods

Important that capability of clustering algorithms are not outstripped by methods in application domains



Introduction

Applications of unsupervised methods in other learning domains include:

- Cluster-based resampling algorithms
- Weakly-supervised learning settings
- Even within classic supervised learning algorithms, like CART

Crucial that clustering methods keep up with developments in these fields.

Introduction - Motivation

So what would we want a clustering method to be?

1. The type of clusters we want to detect:
   - Applications contain clusters of varying shape, size & density
   - Should not need to know the number of clusters in advance

2. These cluster should be detected by an algorithm which:
   - Returns the same results consistently
   - Can be adapted to different settings
   - Runs in a reasonable time (subquadratic at least)

Introduction - Motivation

So what would we want a clustering method to be?

1. The type of clusters we want to detect:
   - Applications contain clusters of varying shape, size & density
   - Should not need to know the number of clusters in advance

2. These cluster should be detected by an algorithm which:
   - Returns the same results consistently
   - Can be adapted to different settings
   - Runs in a reasonable time (subquadratic at least)

Introduction - Motivation

So what would we want a clustering method to be?

1. The type of clusters we want to detect:
   - Applications contain clusters of varying shape, size & density
   - Should not need to know the number of clusters in advance
2. These cluster should be detected by an algorithm which:
   - Returns the same results consistently
   - Can be adapted to different settings
   - Runs in a reasonable time (subquadratic at least)

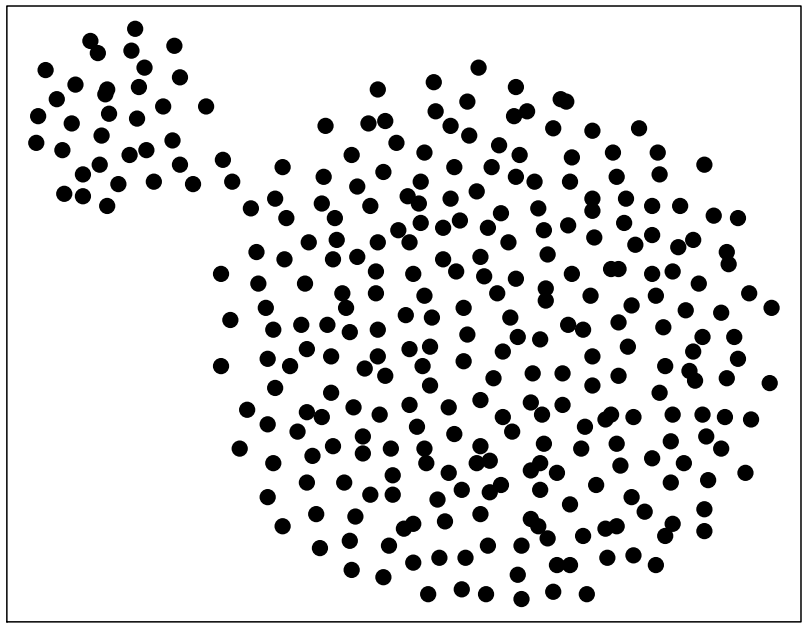
Introduction - Related Work

The Density Peak-Finding (DPC) (Rodriguez and Laio, 2014) algorithm is a potential solution to some of these issues.

“ Cluster centers...

..are surrounded by neighbors with lower local density..

.. and they are at a relatively large distance from any points with a higher local density ”



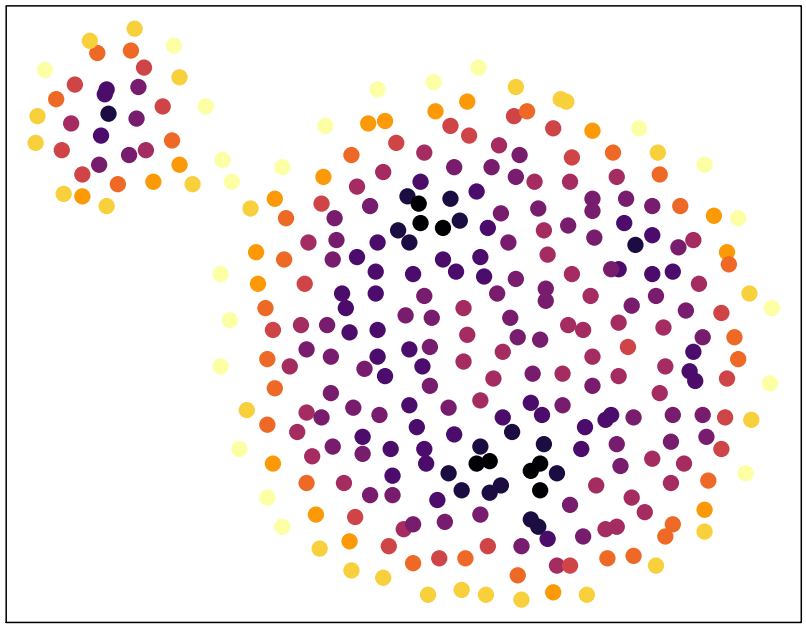
Introduction - Related Work

The Density Peak-Finding (DPC) (Rodriguez and Laio, 2014) algorithm is a potential solution to some of these issues.

“ Cluster centers...

..are surrounded by neighbors with lower local density..

.. and they are at a relatively large distance from any points with a higher local density ”



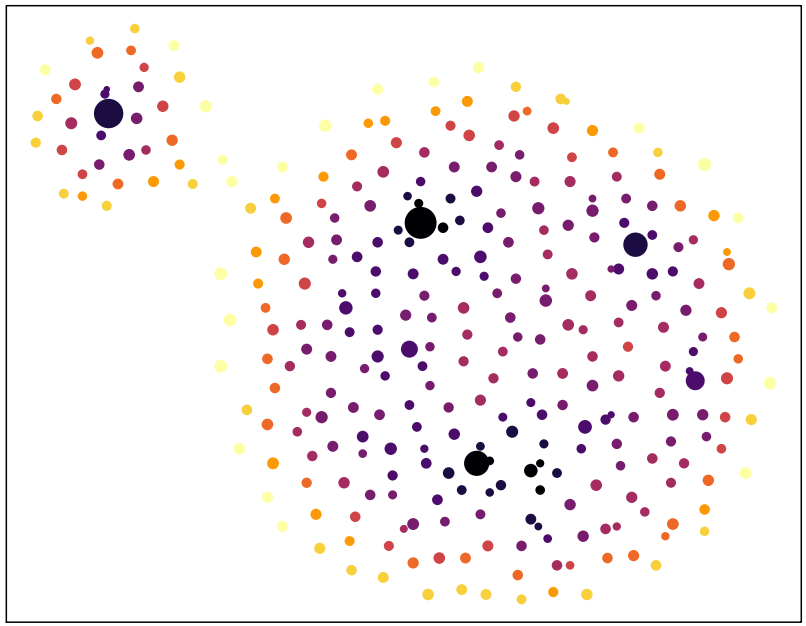
Introduction - Related Work

The Density Peak-Finding (DPC) (Rodriguez and Laio, 2014) algorithm is a potential solution to some of these issues.

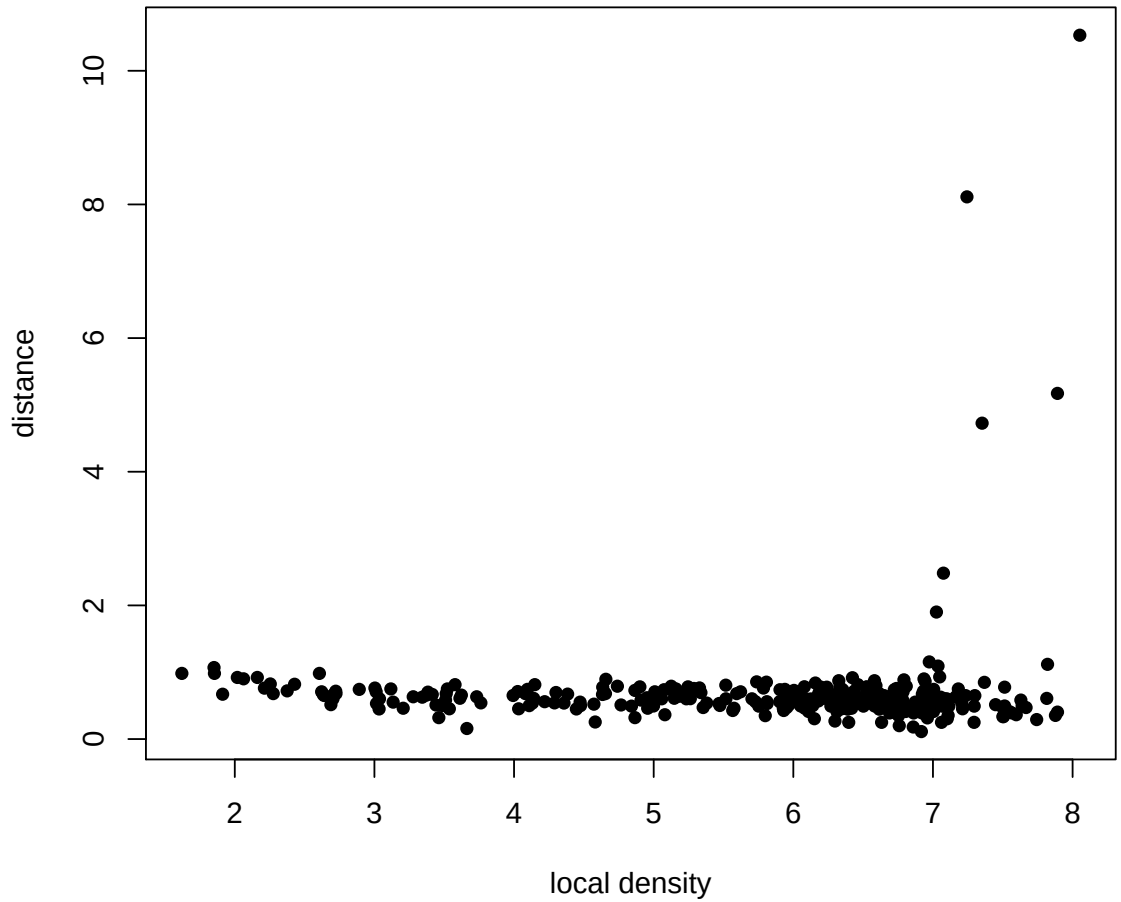
“ Cluster centers...

..are surrounded by neighbors with lower local density..

.. and they are at a relatively large distance from any points with a higher local density ”

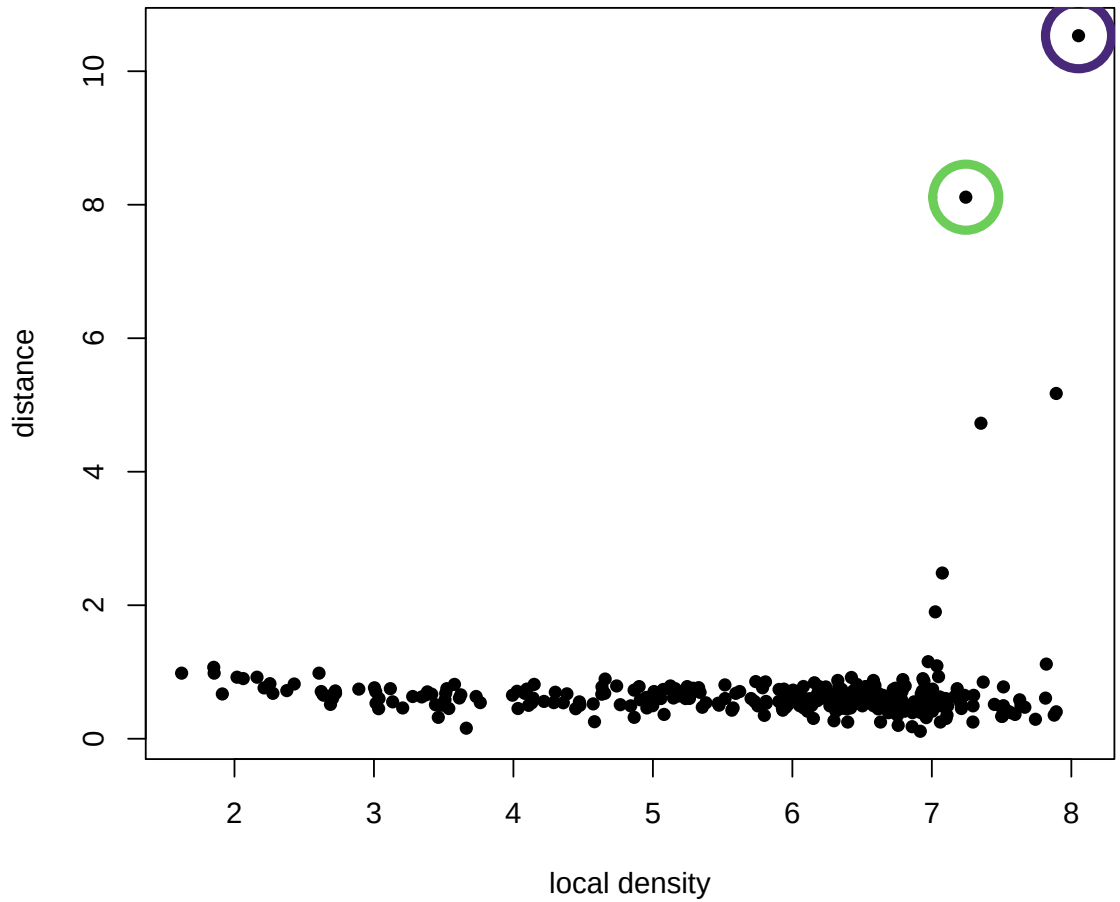


Introduction - DPC Method



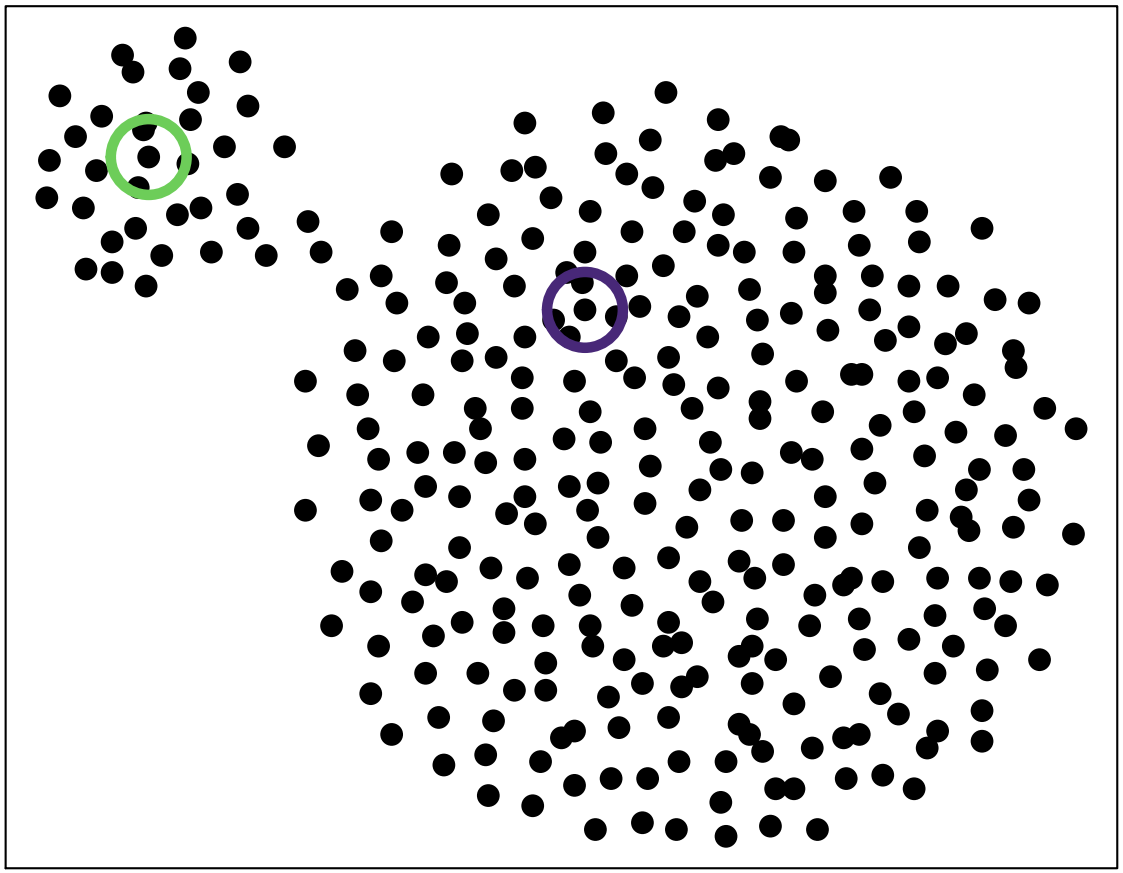
1. Plot a decision graph of local density against distance to nearest neighbor of higher density.

Introduction - DPC Method



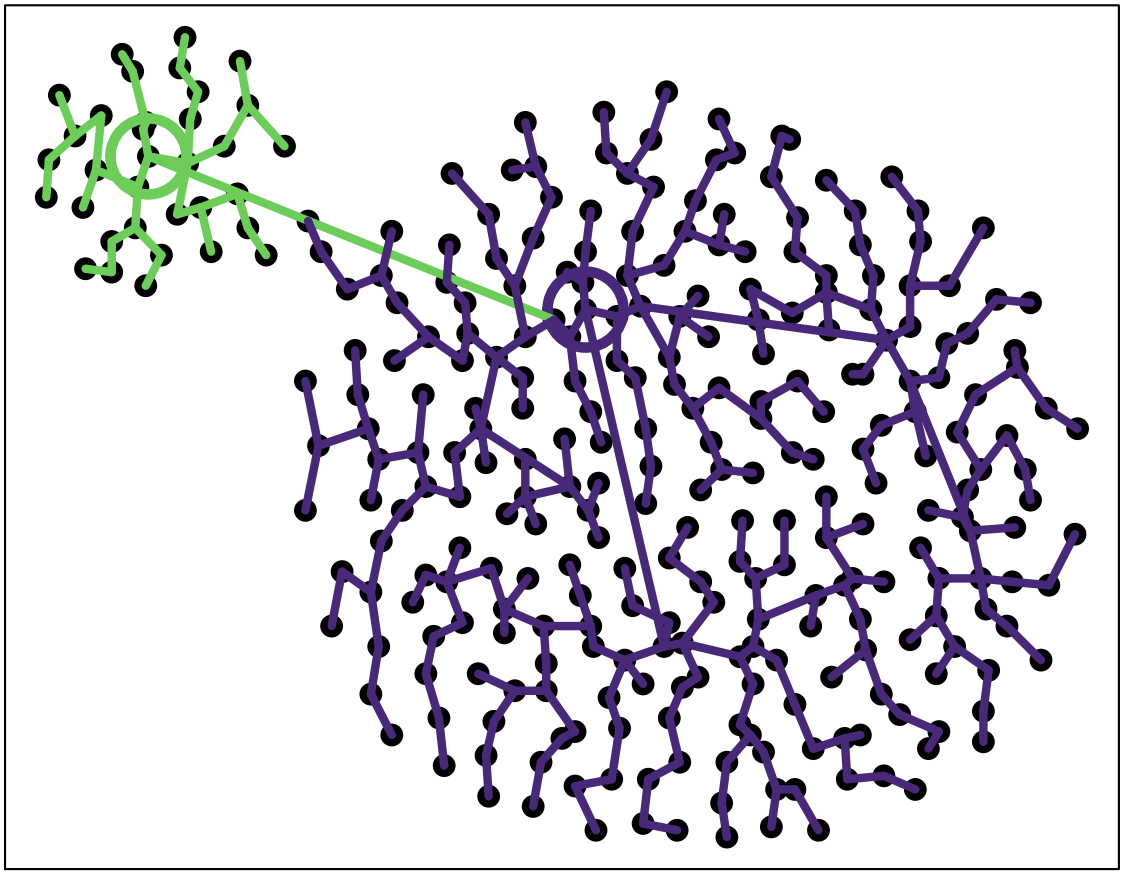
2. Select extreme points as cluster centers.

Introduction - DPC Method



3. Proposed cluster centers on original data.

Introduction - DPC Method



4. Assign each instance to the same cluster as its nearest neighbor of higher local density.

Related Work

DPC has been adapted & extended in three main ways:

1. For mixed attribute data
   - Ding et al. (2017) use an exponential dissimilarity function to incorporate categorical attributes
2. With local estimates of the density
   - Yaohui et al. (2017) applies kernel functions on nearest neighbors
3. To detect clusters automatically
   - Liu et al. (2019) describe ways to compute thresholds for the decision graph
   - Yaohui et al. (2017) iteratively merge candidate clusters together

CPF - Our Method

Motivated by DPC, we develop a method for clustering big data with mixed attributes.

1. Define a new distance metric which balanced contributions from numerical and categorical attributes.
2. Utilize the concept of 'components' from graph theory to detect clusters of varying density.
3. Introduce a new automatic center selection method.

With fast k-nearest neighbor method, the complexity of our method is $O(n \log n)$.

CPF - Distance Metric

We seek a distance metric with balanced contributions from numerical & categorical attributes.

- Numerical attributes are standardised by subtracting the mean and scaling by standard deviation.
- Ordinal attrbibutes are first encoded as integers and then standardised in the same way.
- Categorical/Nominal attributes are one-hot encoded using dummy variables. For a feature with q categories, q dummy variables are required.

2.17	F	Truck
2.36	E	Truck
0.28	A	Boat
1.04	C	Truck
2.38	B	Truck
1.94	D	Car
2.20	F	Van
7.23	C	Van
9.25	B	Boat
8.43	A	Car
5.98	B	Car
6.66	A	Boat
3.30	A	Boat
8.61	B	Car

CPF - Distance Metric

We seek a distance metric with balanced contributions from numerical & categorical attributes.

- Numerical attributes are standardised by subtracting the mean and scaling by standard deviation.
- Ordinal attrbibutes are first encoded as integers and then standardised in the same way.
- Categorical/Nominal attributes are one-hot encoded using dummy variables. For a feature with q categories, q dummy variables are required.

-0.75	F	Truck
-0.68	E	Truck
-1.37	A	Boat
-1.12	C	Truck
-0.68	B	Truck
-0.82	D	Car
-0.74	F	Van
0.93	C	Van
1.61	B	Boat
1.33	A	Car
0.52	B	Car
0.74	A	Boat
-0.37	A	Boat
1.39	B	Car

CPF - Distance Metric

We seek a distance metric with balanced contributions from numerical & categorical attributes.

- Numerical attributes are standardised by subtracting the mean and scaling by standard deviation.
- Ordinal attrbibutes are first encoded as integers and then standardised in the same way.
- Categorical/Nominal attributes are one-hot encoded using dummy variables. For a feature with q categories, q dummy variables are required.

-0.75	1.85	Truck
-0.68	1.27	Truck
-1.37	-1.03	Boat
-1.12	0.12	Truck
-0.68	-0.45	Truck
-0.82	0.70	Car
-0.74	1.85	Van
0.93	0.12	Van
1.61	-0.45	Boat
1.33	-1.03	Car
0.52	-0.45	Car
0.74	-1.03	Boat
-0.37	-1.03	Boat
1.39	-0.45	Car

CPF - Distance Metric

We seek a distance metric with balanced contributions from numerical & categorical attributes.

- Numerical attributes are standardised by subtracting the mean and scaling by standard deviation.
- Ordinal attrbibutes are first encoded as integers and then standardised in the same way.
- Categorical/Nominal attributes are one-hot encoded using dummy variables. For a feature with q categories, q dummy variables are required.

Num	Ord	Truck	Car	Van	Boat
-0.75	1.85	1	0	0	0
-0.68	1.27	1	0	0	0
-1.37	-1.03	0	0	0	1
-1.12	0.12	1	0	0	0
-0.68	-0.45	1	0	0	0
-0.82	0.70	0	1	0	0
-0.74	1.85	0	0	1	0
0.93	0.12	0	0	1	0
1.61	-0.45	0	0	0	1
1.33	-1.03	0	1	0	0
0.52	-0.45	0	1	0	1
0.74	-1.03	0	0	0	1
-0.37	-1.03	0	0	0	1
1.39	-0.45	0	1	0	0

CPF - Distance Metric

The distance between the two points $\boldsymbol{x}_i$ and $\boldsymbol{x}$ is defined as follows:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_{l=1}^{p_1} \rho_l \| \sqrt{\boldsymbol{w}^l} \circ (\boldsymbol{b}_i^l - \boldsymbol{b}_j^l) \|_2^2 + \| \boldsymbol{z}_i - \boldsymbol{z}_j \|_2^2,$$

where $\| \cdot \|_2$ is the Euclidean norm.

- For numerical and ordinal variables, Euclidean norm is used as in DPC.
- For binarised categorical attributes, weights are used in conjunction with the Euclidean norm.

CPF - Distance Metric

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_{l=1}^{p_1} \rho_l \| \sqrt{\boldsymbol{w}^l} \circ (\boldsymbol{b}_i^l - \boldsymbol{b}_j^l) \|_2^2 + \| \boldsymbol{z}_i - \boldsymbol{z}_j \|_2^2,$$

The two weights are:

1. $\{\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{p_1}\}$ - Matrix of weights applied to each binary feature computed from category frequency. Used to weight importance of features.
2. $\rho_1, \ldots, \rho_{p_1}$ - Scalar weights applied to ensure expected contribution of each categorical feature matches standardised numerical features.

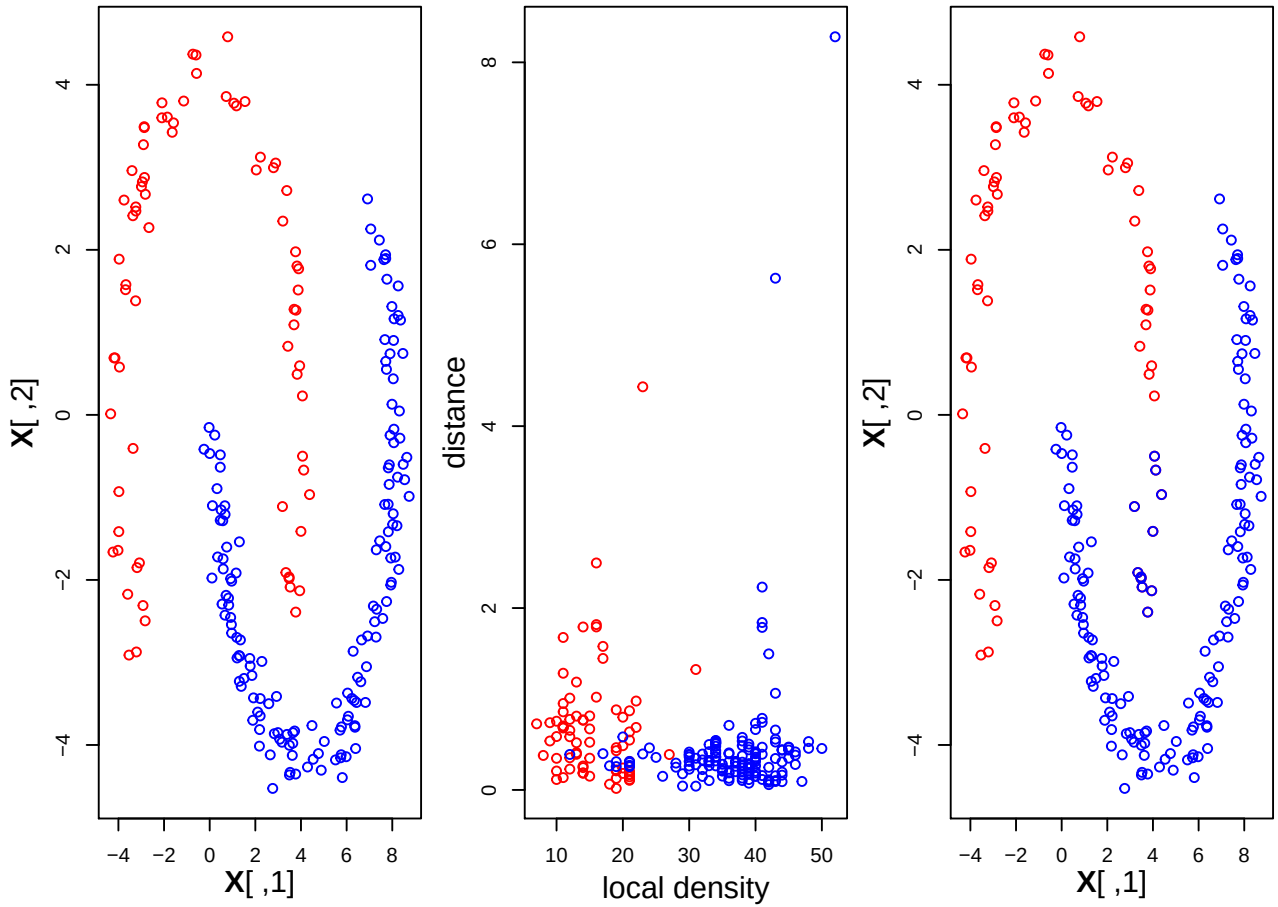
CPF - Distance Metric

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_{l=1}^{p_1} \rho_l \|\sqrt{\boldsymbol{w}^l} \circ (\boldsymbol{b}_i^l - \boldsymbol{b}_j^l)\|_2^2 + \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2,$$

The two weights are:

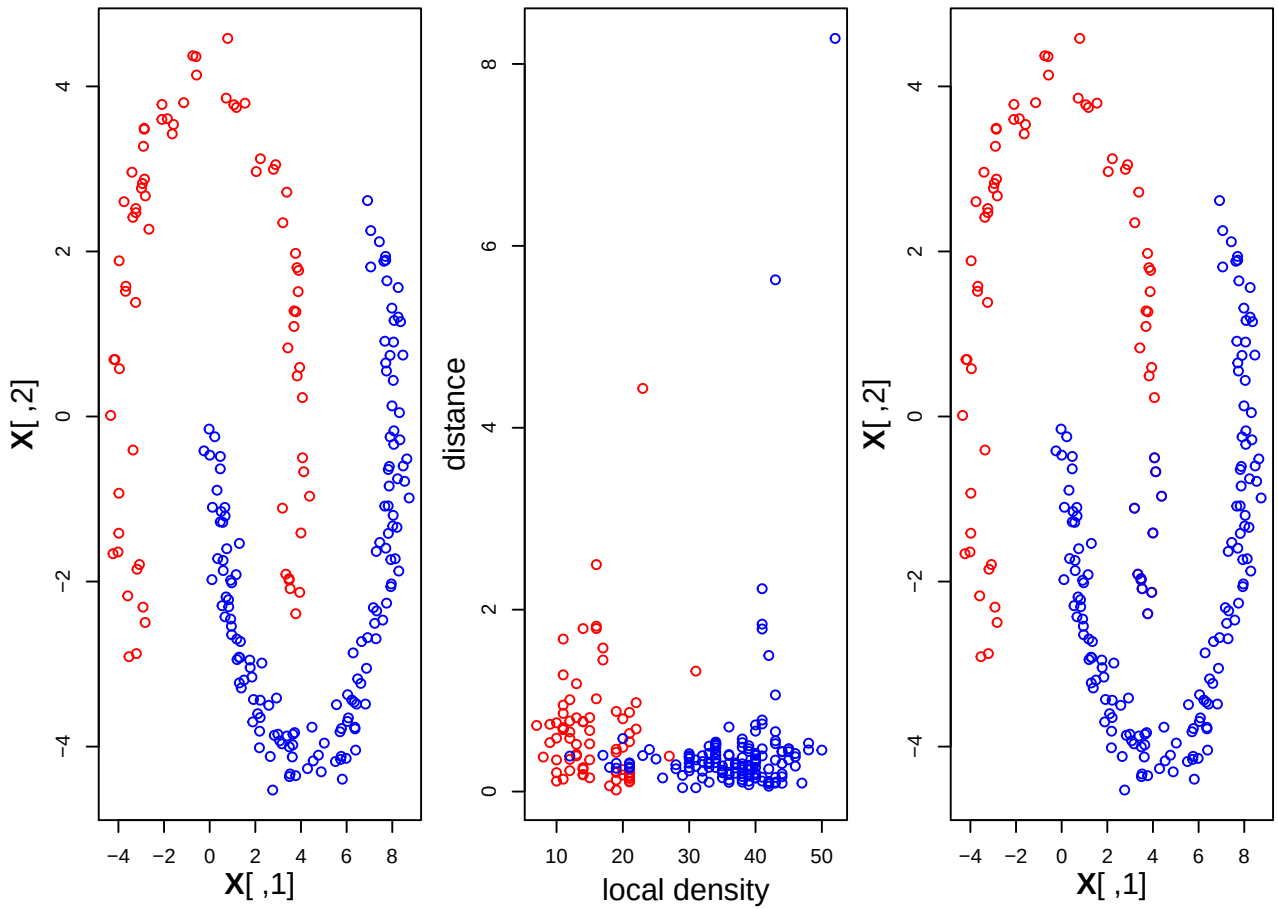
1. $\{\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{p_1}\}$ - Matrix of weights applied to each binary feature computed from category frequency. Used to weight importance of features.
2. $\rho_1, \ldots, \rho_{p_1}$ - Scalar weights applied to ensure expected contribution of each categorical feature matches standardised numerical features.

CPF - Motivations



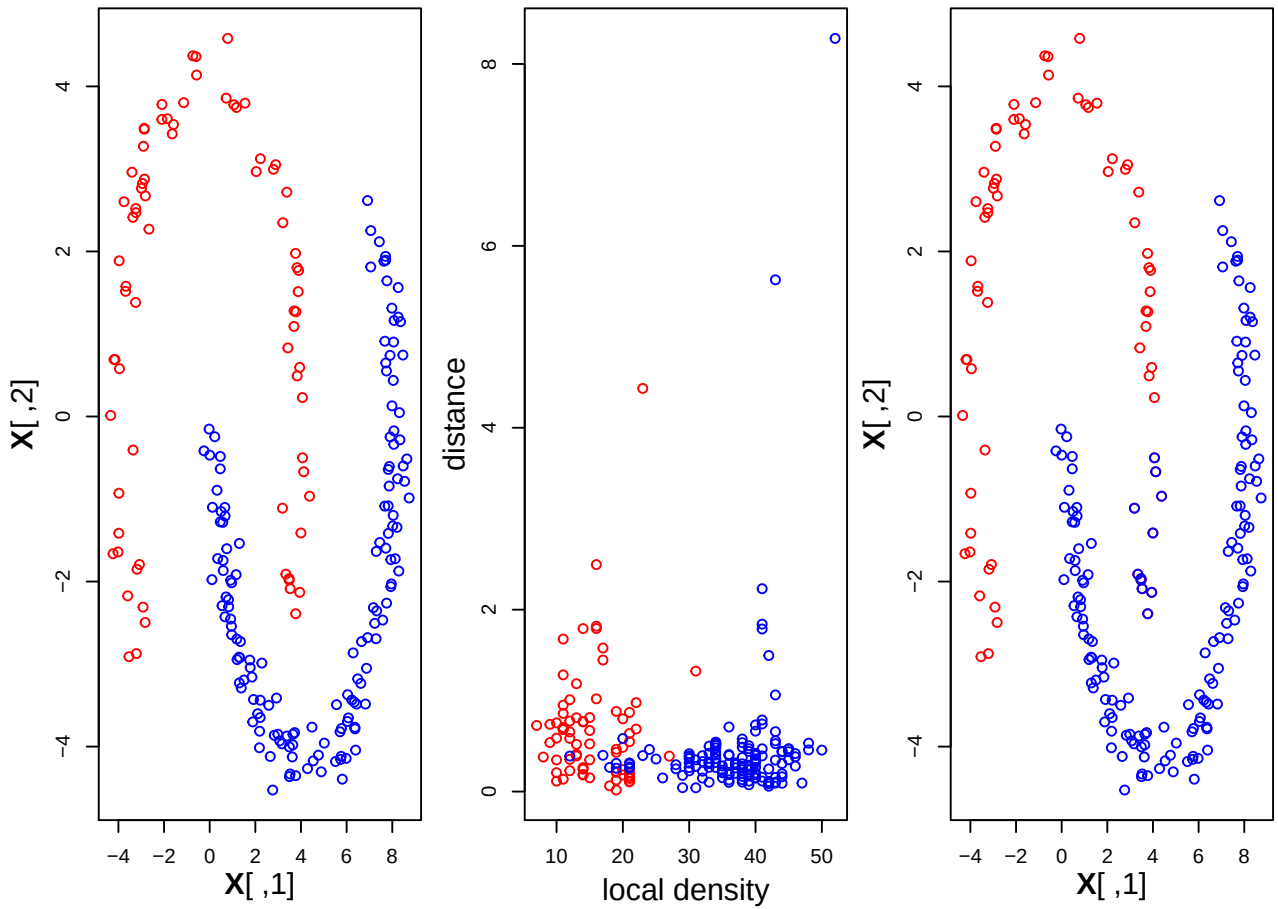
We apply DPC to simulated data with two numerical features to explain its drawbacks and the motivations for our method.

CPF - Motivations



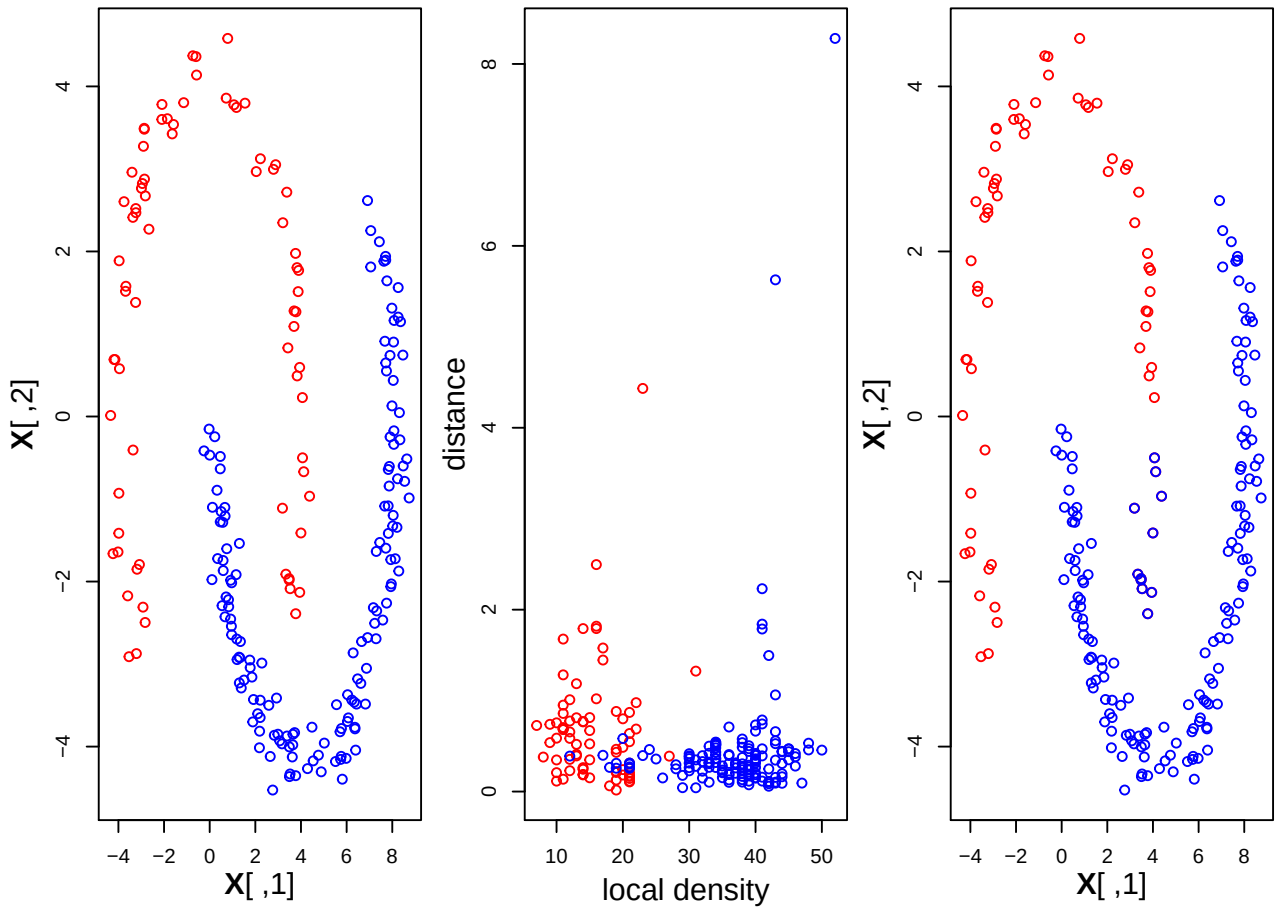
1. Difference in density between the clusters leads to incorrect centers suggested in the middle plot.

CPF - Motivations



2. Even selecting the top blue and red points leads to incorrect assignment of instances.

CPF - Motivations



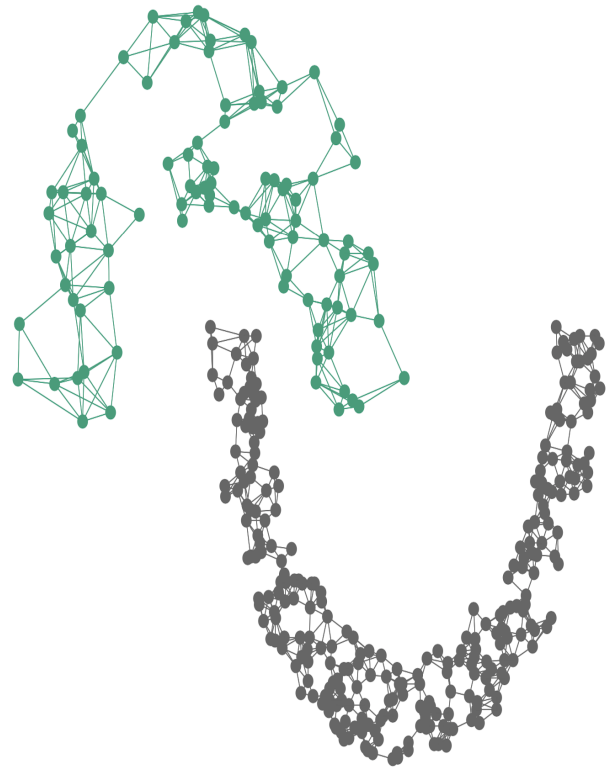
3. Correct number of cluster centers is not obvious from this representation.

CPF - Method

Issues (1) & (2) are caused by DPC not adequately capturing underlying structure of the data

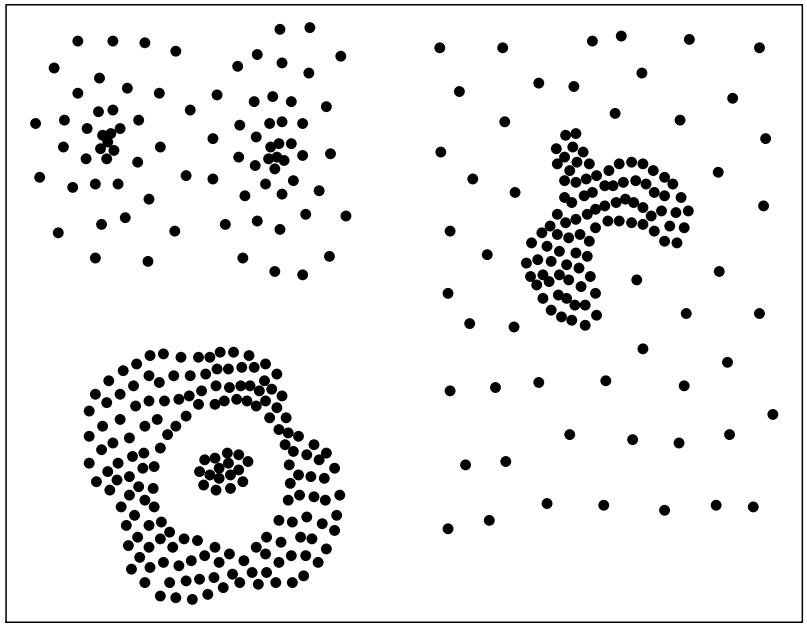
Shared Nearest Neighbors

- Idea is to build a graph where every instance in the dataset is a vertex and an edge between vertices exists if they are *both* in k-nn of each other
- Used as a standalone clustering method in Ertöz et al. (2003)
- Effective at detecting noise points and separates areas with different density



CPF - Method

Components can reveal underlying patterns of the data.

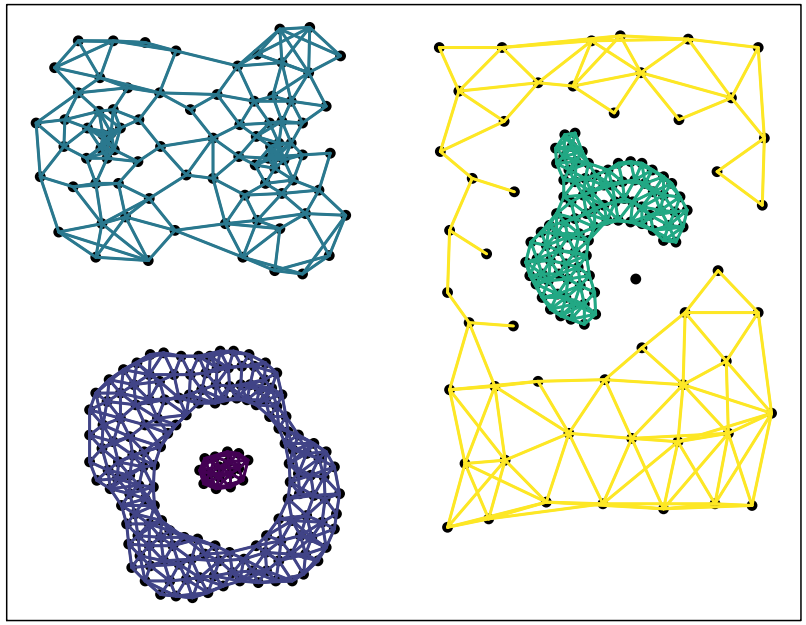


If two instances lie in different components, they are highly likely to belong to different clusters

We propose applying DPC on each individual component set, Component-wise Peak Finding (CPF)

CPF - Method

Components can reveal underlying patterns of the data.



If two instances lie in different components, they are highly likely to belong to different clusters

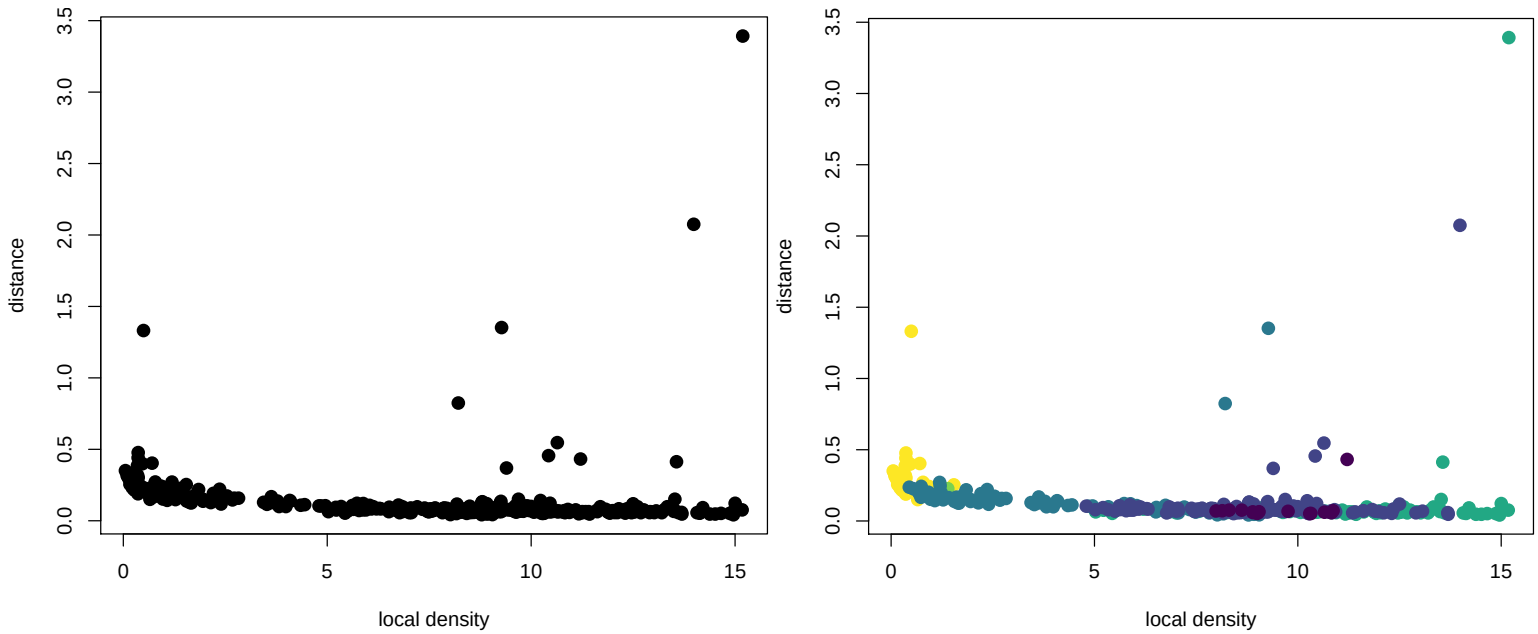
We propose applying DPC on each individual component set, Component-wise Peak Finding (CPF)

CPF - Method

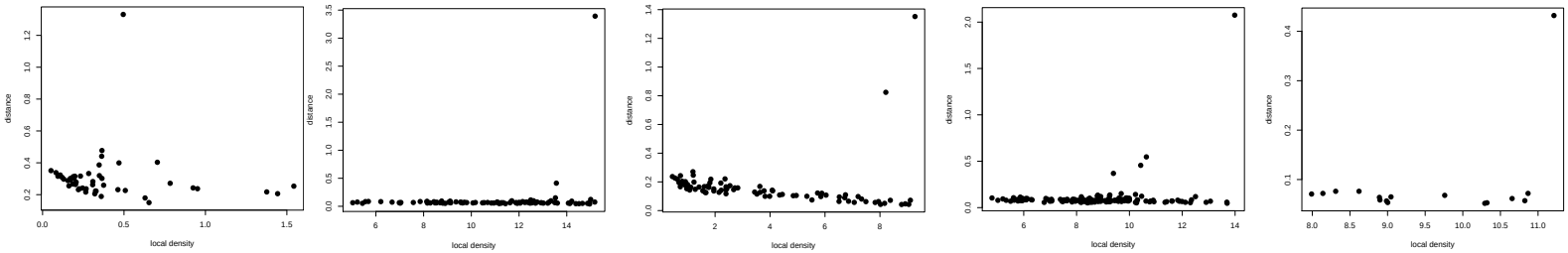
The Algorithm:

1. Create a shared nearest neighbor graph, with the parameter k.
2. Remove points with few incoming edges as outliers and create the component sets.
3. For each component set:
   - for every instance in the component set, find the K nearest neighbors
   - using these K nearest neighbors, calculate the local density as $\sum_{\boldsymbol{x} \in \mathcal{N}_K(\boldsymbol{x}_i)} \exp(-d(\boldsymbol{x}_i, \boldsymbol{x}))$
   - find the distance to the nearest neighbor of higher density for each instance
4. Select the cluster centers for each component set.
5. For a non-center point, assign it to the same cluster as its nearest neighbor of higher local density.

DPC Graphs



CPF Graphs



CPF - Center Selection

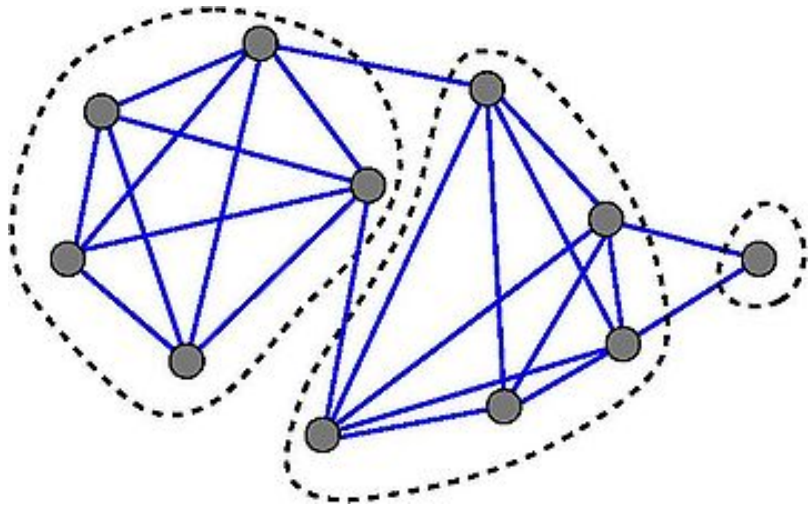
- Separating the data into components does improve discrimination of centers
- But analysing multiple decision graphs is not feasible in *any* application
- We develop an automatic center selection method
  - Computing the local density of each instance requires detecting the K nearest neighbors in the component set
  - The centers proposed by the decision graphs form partitions on the K-nn graph
  - Propose assessing the partitions using methods from graph community detection

CPF - Center Selection

- A cut on the graph $G(\mathbb{C})$ partitions component $\mathbb{C}$ into two non-empty subsets S and $\bar{S}$.

- The conductance of a cut $(S, \bar{S})$ of $\mathbb{C}$ is:

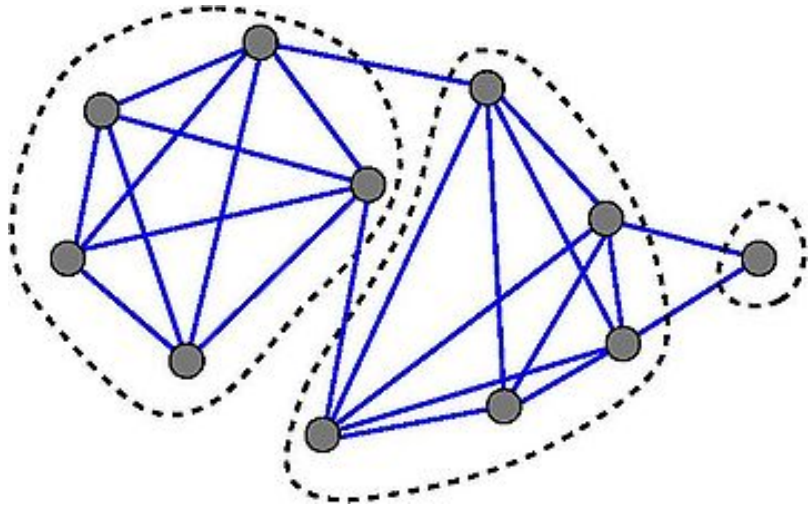
$$\Phi(S, \bar{S}; G(\mathbb{C})) = \frac{\sum_{\boldsymbol{x}_i \in S, \boldsymbol{x}_j \in \bar{S}} w(\{\boldsymbol{x}_i, \boldsymbol{x}_j\})}{\min\{w(S), w(\bar{S})\}}.$$



CPF - Center Selection

- A cut on the graph $G(\mathbb{C})$ partitions component $\mathbb{C}$ into two non-empty subsets S and $\bar{S}$.
- The conductance of a cut $(S, \bar{S})$ of $\mathbb{C}$ is:

$$\Phi(S, \bar{S}; G(\mathbb{C})) = \frac{\sum_{x_i \in S, x_j \in \bar{S}} w(\{x_i, x_j\})}{\min\{w(S), w(\bar{S})\}}.$$

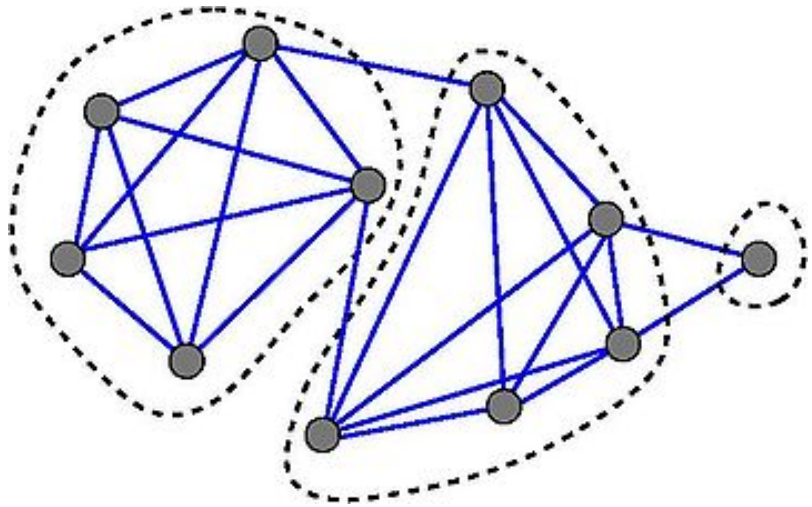


CPF - Center Selection

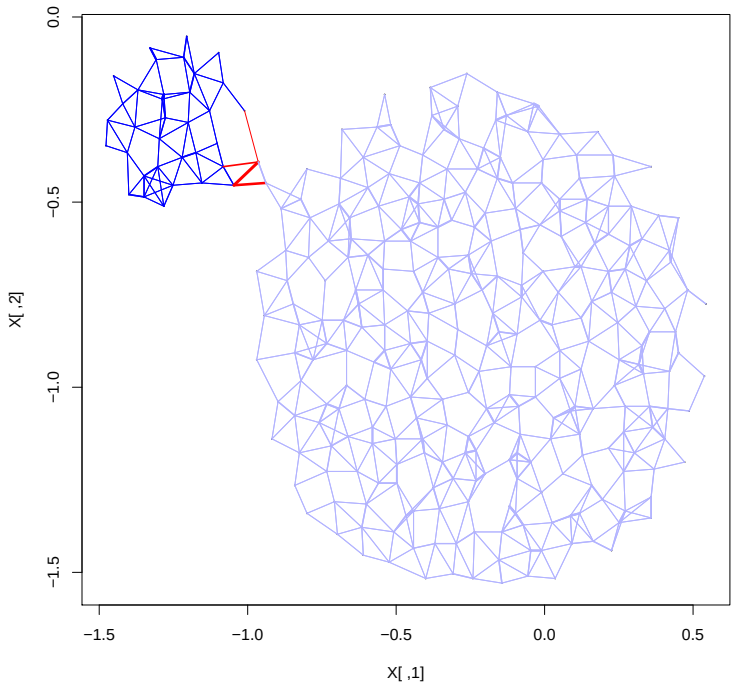
- A cut on the graph $G(\mathbb{C})$ partitions component $\mathbb{C}$ into two non-empty subsets S and $\bar{S}$.

- The conductance of a cut $(S, \bar{S})$ of $\mathbb{C}$ is:

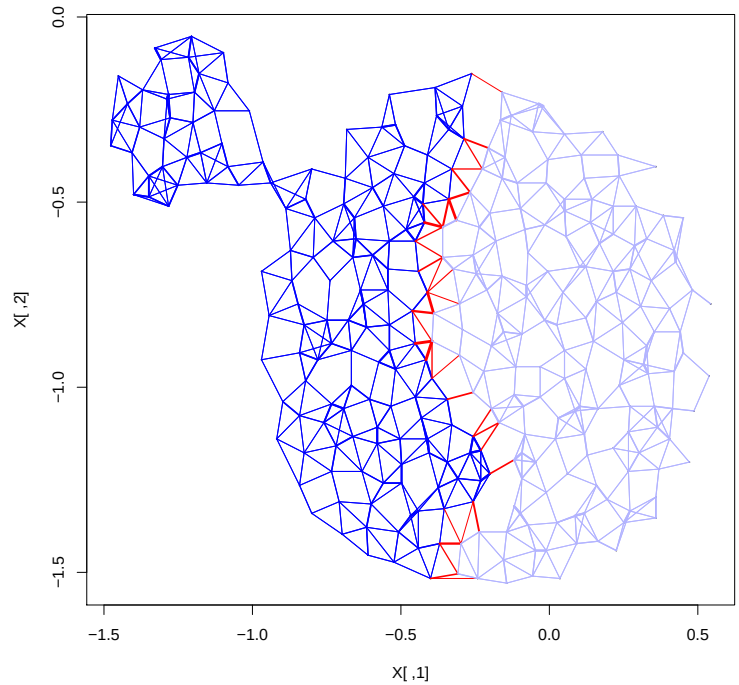
$$\Phi(S, \bar{S}; G(\mathbb{C})) = \frac{\sum_{\boldsymbol{x}_i \in S, \boldsymbol{x}_j \in \bar{S}} w(\{\boldsymbol{x}_i, \boldsymbol{x}_j\})}{\min\{w(S), w(\bar{S})\}}.$$



CPF - Center Selection



(a) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.02$



(b) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.04$

- Captures notion that clusters (or communities) should have stronger connections within than without.
- Shown that local minima of conductance values correspond to best clusterings Leskovec et al. (2008).

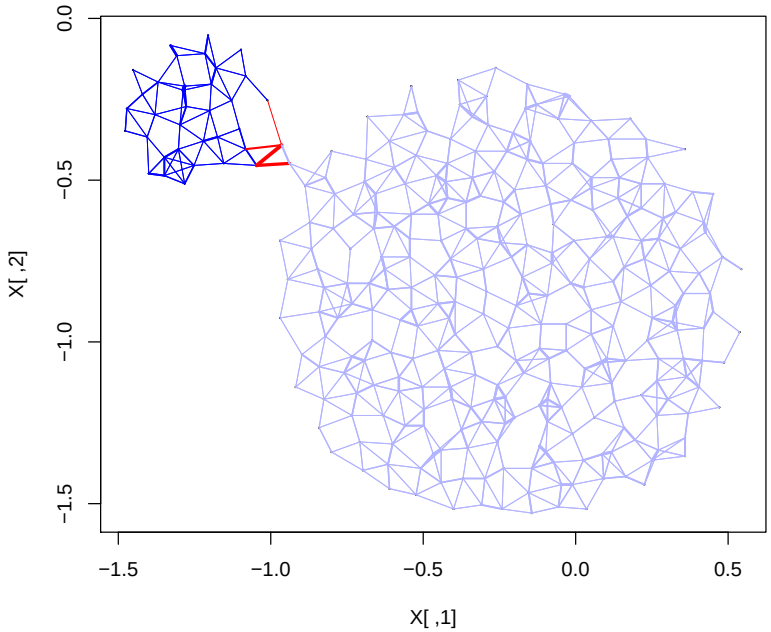
CPF - Center Selection

- The peak-finding method automatically suggests candidate cluster centers.
- These are found by sorting the product of local density and distance to nearest neighbor of higher density in decreasing order.
- Selecting instances as candidate centers defines a partition of the nearest neighbor graph for the component $G(\mathbb{C})$
- We use these partitions to answer two questions:
  1. Does the component set contain more than one cluster?
  2. If so, how many cluster centers should be selected?

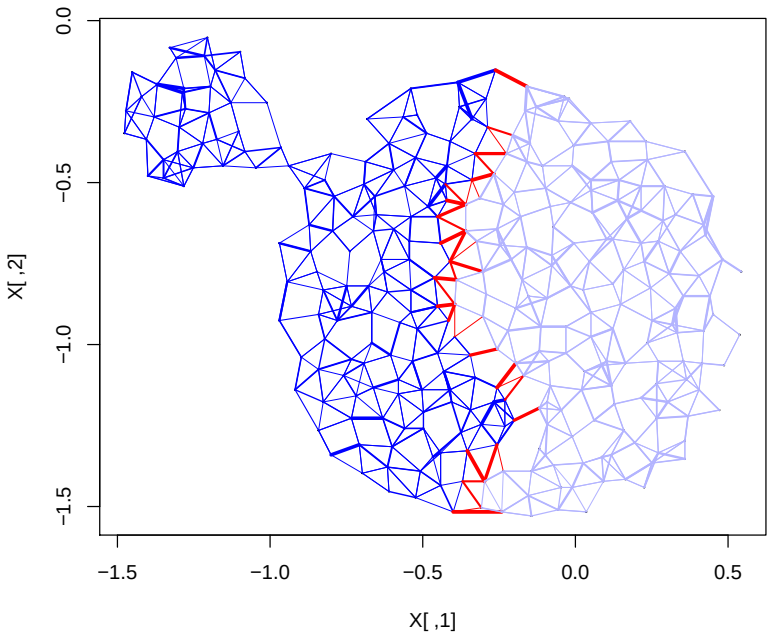
CPF - Center Selection

- We cannot assume that more than one cluster exists in each component.
- Heuristic: *'Internal connections are stronger & more prevalent than external connections'*
- So, taking the top two cluster centers proposed for a given component:
  1. Find the minimal value $\tilde{k}$ such that the clusters are connected.
  2. Calculate the conductance of the cut between the clusters.
  3. Set $\hat{k} = \tilde{k} + 1$. Create and calculate the conductance on the new graph.
  4. If the conductance increases, then component contains only one cluster. Otherwise, store the conductance value and continue.

CPF - Center Selection

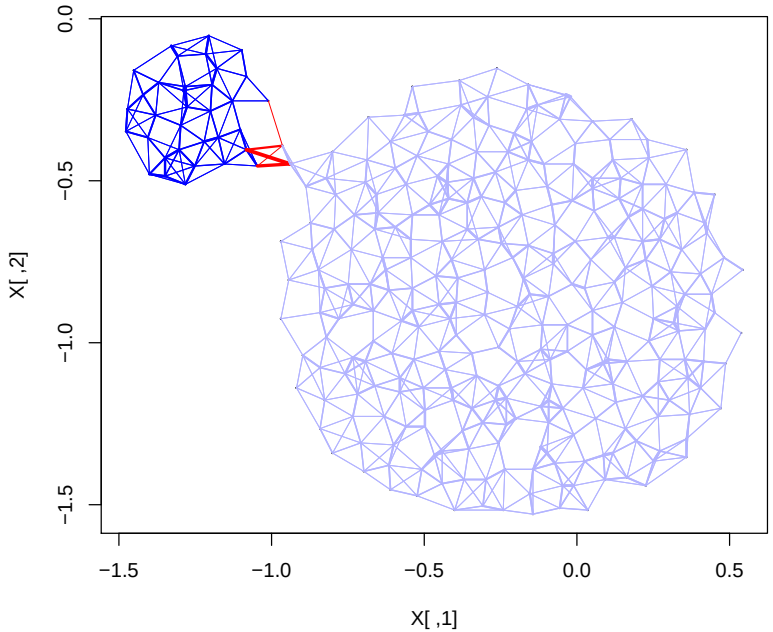


(a) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.02477$

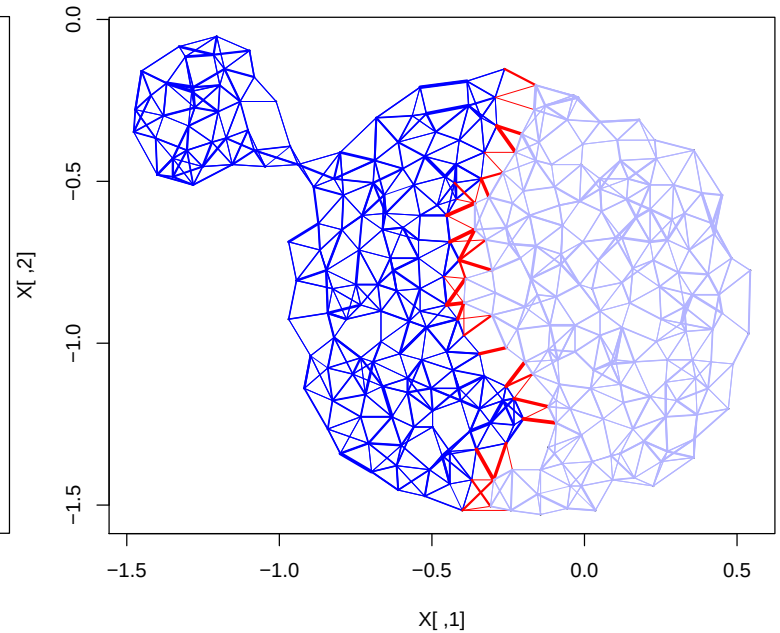


(b) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.04329$

CPF - Center Selection



(a) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.02328$



(b) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.04792$

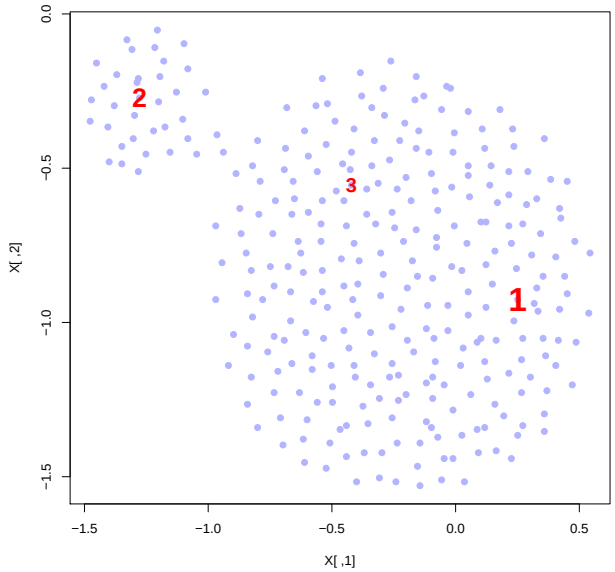
CPF - Center Selection

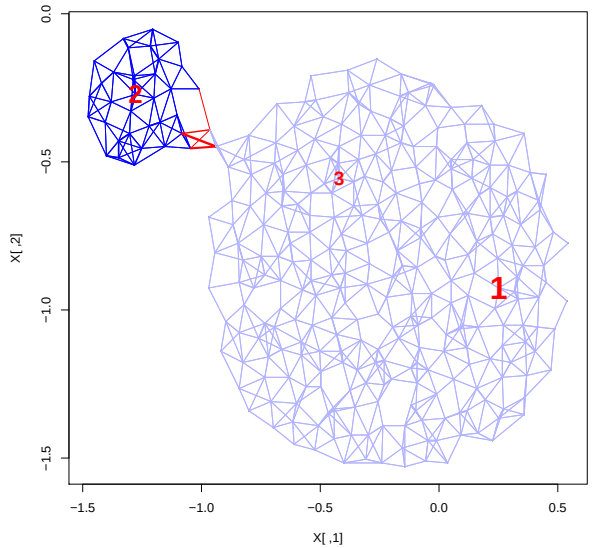
Once we determine more than one cluster exists in a component, we must determine how many?

2. For each center proposed by the clustering method:
   - Compute the conductance of each proposed cluster in the clustering.
   - Record the maximum conductance for each clustering, as Φ_j where j is the number of centers in the clustering.

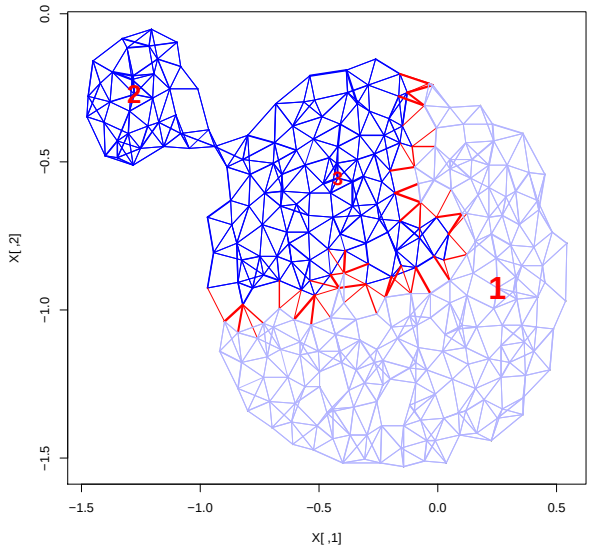
3. The final clustering has n_{clust} centers where $n_{clust} = \arg\min_j \Phi_j$.

This process is repeated for each component.





(a) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.0247$



(b) $\Phi(S, \bar{S}; G(\mathbb{C})) = 0.0701$

CPF - Complexity

- Computing the graph $G(\mathbb{X}, E)$ is $O(n \log(n))$ using fast k-nn methods (Zhang et al., 2013)
- Finding the nearest neighbor of higher density is also taxing
- If a point with higher density can be found in the K neighbors computed in the density step, no computation is needed.
- Else, a broad search must be undertaken
  - If the proportion that require a broad search is p, the complexity is $O(p|\mathbb{C}|^2)$; p is typically less than 1% for large components.
- So the complexity CPF is $O(n \log(n) + n_c^2)$, where n_c is the size of the largest component.

Experimental Study - Numerical

		k-means	Gen Clust++	AD k-means	CPF
Dermatology	ARI	0.701	0.761	0.001	**0.845**
	PS	0.858	**0.989**	0.310	0.916
	F1	0.241	0.003	0.159	**0.304**
	NMI	0.849	0.842	0.010	**0.873**
	CA	0.196	0.003	0.165	**0.304**
Page Blocks	ARI	0.101	0.002	0.013	**0.386**
	PS	0.905	0.450	0.898	**0.960**
	F1	0.010	**0.309**	0.201	0.001
	NMI	0.076	0.031	0.006	**0.273**
	CA	0.049	**0.420**	0.152	0.001
Wine Quality	ARI	0.034	0.000	0.001	**0.139**
	PS	0.477	**1.000**	0.449	0.559
	F1	0.155	0.064	**0.188**	0.139
	NMI	0.069	0.000	0.002	**0.076**
	CA	0.118	0.033	**0.146**	0.029

Experimental Study - Categorical

		k-modes	Gen Clust++	AD k-means	CPF
TicTacToe	ARI	0.011	0.007	0.028	**0.612**
	PS	0.653	0.040	0.655	**1.000**
	F1	0.148	0.002	**0.501**	0.015
	NMI	0.004	0.091	0.014	**0.596**
	CA	0.164	0.023	**0.491**	0.015
Breast Cancer	ARI	-0.001	0.001	0.140	**0.207**
	PS	0.776	0.166	0.775	**0.780**
	F1	0.442	0.013	**0.510**	0.083
	NMI	-0.003	0.026	0.055	**0.079**
	CA	0.350	0.051	**0.500**	0.083

Experimental Study - Mixed

		k-proto.	Gen Clust++	AD k-means	KAMILA	CPF
Credit	ARI	0.021	0.063	**0.409**	0.335	0.267
	PS	0.576	0.167	**0.807**	0.790	0.606
	F1	0.233	0.003	**0.255**	0.169	0.205
	NMI	0.013	0.156	**0.355**	0.329	0.049
	CA	0.224	0.018	**0.266**	0.210	0.032
KDD'99 RIvs.B	ARI	0.085	0.005	0.010	0.000	**0.151**
	PS	0.991	0.590	0.993	0.990	**1.000**
	F1	0.001	0.068	**0.224**	0.000	0.151
	NMI	0.069	0.008	**0.316**	0.000	0.010
	CA	0.000	0.187	0.163	**0.394**	0.028
KDD'99 GPvs.S	ARI	0.000	-0.003	-0.003	0.000	**0.118**
	PS	0.997	0.776	0.997	0.997	**0.997**
	F1	0.000	0.028	**0.680**	0.000	0.189
	NMI	0.000	0.001	0.001	0.000	**0.003**
	CA	0.000	0.113	**0.585**	0.003	0.102

Experimental Study - Large

		k-proto.	KAMILA	CPF
CovType	ARI	0.031	0.050	**0.153**
	PS	0.515	0.494	**0.670**
	F1	0.140	0.002	**0.341**
	NMI	0.078	0.132	**0.202**
	CA	0.249	0.063	**0.341**
KDD'99 DOSvs.NORM	ARI	-	0.000	**0.089**
	PS	-	0.181	**0.652**
	F1	-	0.000	**0.121**
	NMI	-	0.000	**0.020**
	CA	-	0.000	**0.121**

	k-proto.	KAMILA	CPF
Cov Type	52603.612	283.262	17782.344
KDD '99 DOS vs. NORM	-	416.293	137283.899

Experimental Study

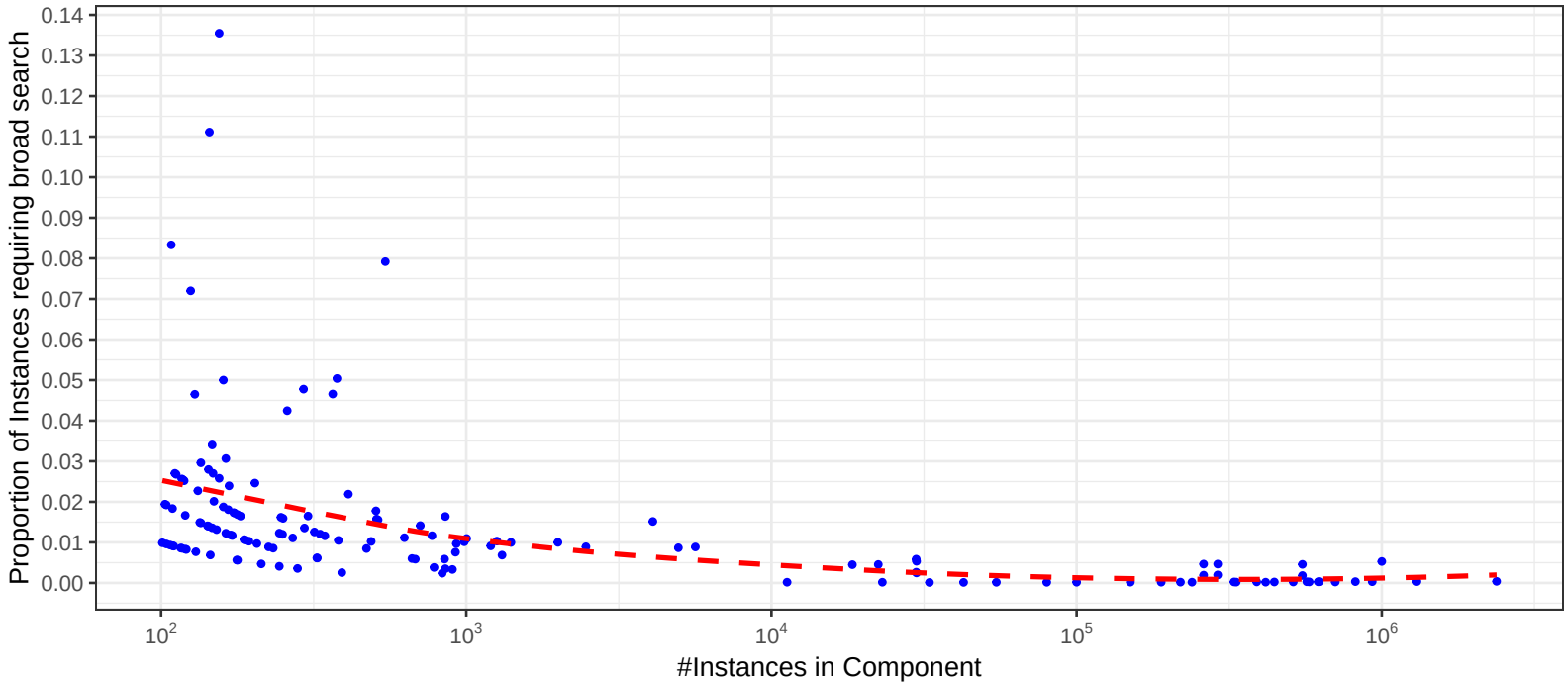
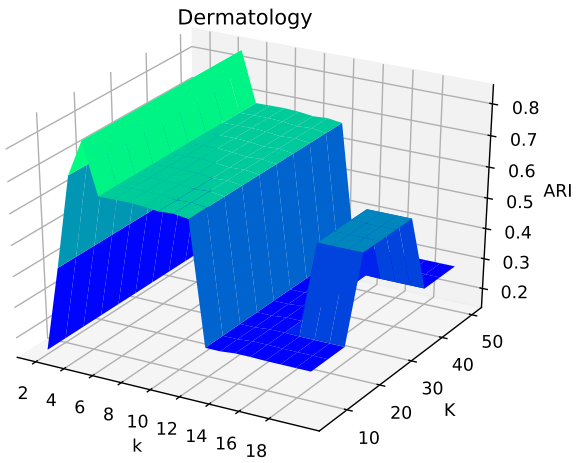


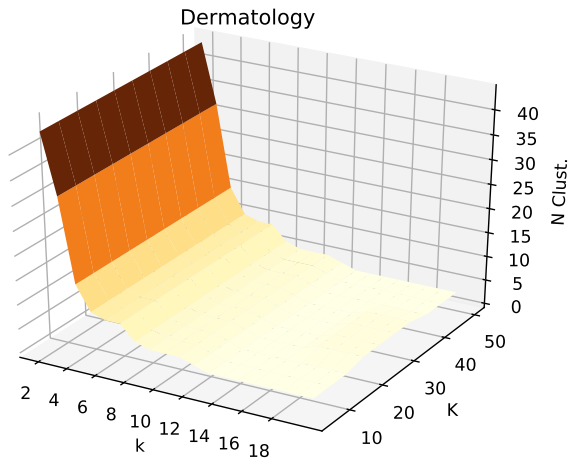
Figure: Number of instances in a component ($|\mathbb{C}|$) vs. Proportion of instances requiring a broad search (p).

- Experimental analysis supports complexity analysis, p is regularly less than 1% for large components

Experimental Study



(a) ARI of CPF vs. k and K.



(b) #Clust of CPF vs. k and K.

- k - Quality of clustering is high when k is in the range $[5, 10]$
- K - Observations indicate setting $K \approx \sqrt{n}$ is effective

Summary

- Introduce a new clustering algorithm, CPF, for large mixed attribute data
- Adapts & extends the peak-finding method using connected components
- Components give improved results while reducing complexity to $O(n \log n)$
- Experimental results indicate superior results compared to benchmark k-means type methods
- Driven by flexibility of methods, distance metrics & parameters allow detection of arbitrary clusters
- Code for the CPF method is available for download from https://pypi.org/project/CPFcluster/

Bibliography I

Ding, S., Du, M., Sun, T., Xu, X., and Xue, Y. (2017). An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood. *Knowledge-Based Systems*, 133:294–313.

Ertöz, L., Steinbach, M., and Kumar, V. (2003). Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, Proceedings, pages 47–58. Society for Industrial and Applied Mathematics.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2008). Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web*, pages 695–704.

Bibliography II

Liu, T., Li, H., and Zhao, X. (2019). Clustering by Search in Descending Order and Automatic Find of Density Peaks. *IEEE Access*, 7:133772–133780.

Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496.

Yaohui, L., Zhengming, M., and Fang, Y. (2017). Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. *Knowledge-Based Systems*, 133:208–220.

Zhang, Y.-M., Huang, K., Geng, G., and Liu, C.-L. (2013). Fast kNN Graph Construction with Locality Sensitive Hashing. In *Advanced Information Systems Engineering*, volume 7908, pages 660–674.