Searching for the most plausible partition: an evidential reasoning approach to clustering

Orakanya Kanjanatarakul & Thierry Denoeux

Sorbonne Universités, Université de Technologie de Compiègne,
CNRS, Heudiasyc UMR 7253, France

{okanjana,tdenoeux}@utc.fr

Introduction

- Clustering can be seen as the search for a "good" partition of a set of n objects described either by attributes, or by a dissimilarity matrix.
- Usual approaches are based either on a geometric criterion, as in the k-means algorithm, or on a finite mixture model whose parameters are estimated using, e.g., the EM algorithm.
- Here, we propose a different view of partitional clustering, in which dissimilarities are seen as pieces of evidence and represented as belief functions on the set of all partitions of the dataset under study.
- Finding the most plausible partition is a linear programming problem, which can be solved exactly for small n.
- A heuristic algorithm (the Ek-NN algorithm) can find a local optimum for large datasets, without specifying the number of classes.

Formalization

Let $\mathcal{O}$ denote a set of n objects and let $\mathcal{R}$ be the set of equivalence relations on $\mathcal{O}$ (this set is in one-to-one correspondence with the set of partitions). We assume the existence of a true equivalence relation R_0. Dissimilarities between objects are considered as items of evidence about R_0, which can be represented by mass function m_{ij} with three focal sets: the set $\mathcal{R}_{ij}$ of equivalence relations containing objects i and j, its complement $\neg\mathcal{R}_{ij}$, and $\mathcal{R}$, and corresponding masses

$$m_{ij}(\mathcal{R}_{ij}) = \alpha_{ij} \quad (1a)$$
$$m_{ij}(\neg\mathcal{R}_{ij}) = \beta_{ij} \quad (1b)$$
$$m_{ij}(\mathcal{R}) = 1 - \alpha_{ij} - \beta_{ij}. \quad (1c)$$

After combining these $n(n-1)/2$ mass functions by Dempster's rule, we get a mass function m on $\mathcal{R}$ with contour function pl defined by the following equation,

$$\ln pl(R) = C + \sum_{i<j} R_{ij} \ln \frac{1-\beta_{ij}}{1-\alpha_{ij}}, \quad (2)$$

where C is a constant. The most plausible partition can thus be found exactly, for small n (until, say, $n \leq 100$) using a binary linear programming solver.

Hopfield model

To make the above approach feasible for large n, we need a heuristic optimization method. We show that a local maximum of $\ln pl(R)$ defined by (2) can be found by a Hopfield neural network model [3] with n neurons, in which each neuron can be in one of c states, where c is the desired number of clusters. The weight v_{ij} of the connection between neurons i and j is the coefficient of R_{ij} in (2). Starting from random initial states, the state of each neuron i is updated at asynchronous times, by finding k such that $\sum_{j \neq i} v_{ij} s_{jk}$ is maximum, where $s_{jk} = 1$ if neuron j is in state k, and $s_{jk} = 0$ otherwise. This algorithm is shown to converge to a global network state that corresponds to a local maximum of (2).

EK-NNclus algorithm

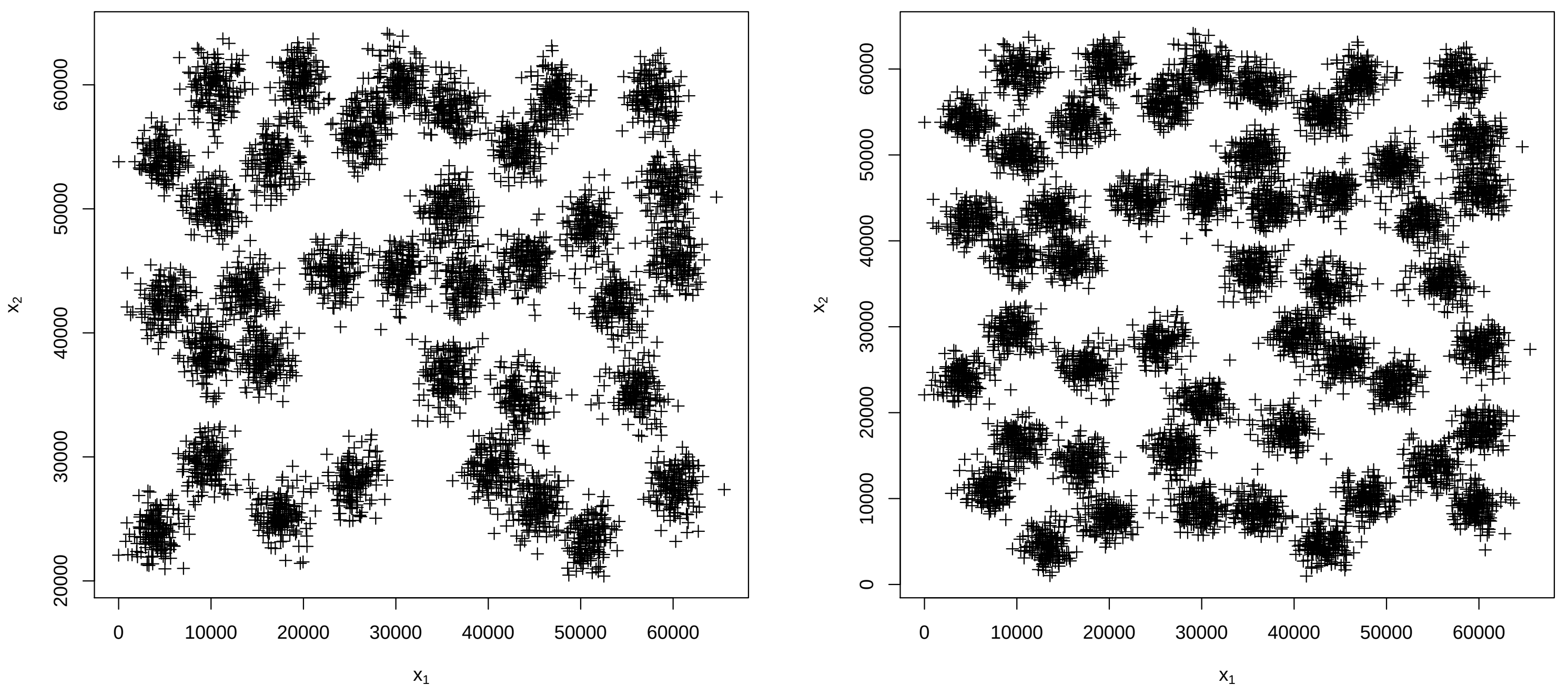
- Fast implementation: $\beta_{ij} = 0$, $\alpha_{ij} = 0$ except for the K nearest neighbors of object o_i.
- Unsupervised version of the evidential K-NN classifier [1].

```
Require: Number of states $c$, distance matrix $D = (d_{ij})$, number of neighbors $K$
  Randomly initialize variables $s_{ik}$ for $i = 1, \ldots, n$; $k = 1, \ldots, c$.
  Compute $\alpha_{ij} = \varphi(d_{ij})$ if $j \in N_K(i)$ and $\alpha_{ij} = 0$ otherwise, and $v_{ij} = -\ln(1-\alpha_{ij})$, for $i = 1, \ldots, n$; $j = 1, \ldots, n$
  $change$ ← true
  while $change$ do
    Select a random permutation $\sigma$ of $\{1, \ldots, n\}$
    $change$ ← false
    for $i = 1$ to $n$ do
      for $k = 1$ to $c$ do
        $u_{\sigma(i)k} \leftarrow \sum_{j \in N_K(\sigma(i))} v_{\sigma(i)j} s_{jk}$
      end for
      $k^* \leftarrow \arg\max_k u_{\sigma(i)k}$
      if $s_{\sigma(i)k^*} = 0$ then
        Set $s_{\sigma(i)k^*} \leftarrow 1$ and $s_{\sigma(i)k} \leftarrow 0$ for all $k \neq k^*$
        $change$ ← true
      end if
    end for
    Update $c$, renumber the clusters and change variables $s_{ik}$ accordingly
  end while
```

Experiments

Settings: $\varphi(d_{ij}) = \exp(-\gamma d_{ij}^2)$, where d_{ij} is the Euclidean distance between objects i and j. Parameter γ was fixed to the inverse of the q-quantile of the set $\Delta = \{d_{ij}^2, i \in \{1, \ldots, n\}, j \in N_K(i)\}$.

A-datasets Two-dimensional datasets with 20, 35 and 50 clusters. Parameter q of the EK-NNclus algorithm was fixed to $q = 0.9$. The number of neighbors was fixed to $K = 150$ for dataset A1, and $K = 200$ for datasets A2 and A3 (i.e., consistently with the rule of thumb that K should be of the order of two to three times $\sqrt{n}$). Two initialization methods were used: $c_0 = n$ initial clusters, and $c_0 = 1000$ random initial clusters. The EK-NNclus algorithm was run 10 times.



Dataset	Result	EK-NNclus ($c_0 = n$)	EK-NNclus ($c_0 = 1000$)	pdfCluster	model-based	model-based (constrained)
A1	c	20 (0)	20 (0)	17	24	24
$n = 3000$	time	32.9 (3.14)	9.8 (0.2)	84.5	31.8	7.88
A2	c	35 (0)	34 (1)	26	39	39
$n = 5250$	time	193 (9.81)	23.8 (0.6)	298	138	36.2
A3	c	49 (1)	49 (2.5)	34	50	51
$n = 7500$	time	358 (8.23)	35.1 (1.09)	629	412	99.4

DIM-datasets High-dimensional data sets $n = 1024$ and 16 Gaussian clusters. Parameters q and K of the EK-NNclus algorithm were fixed to $q = 0.9$ and $K = 50$. The algorithm was initialized with $c_0 = n$ clusters and was run 10 times. The c-means algorithm was run 100 times with $c = 16$ clusters and the result with the best value of the objective function was kept. As the `pdfCluster` procedure cannot be used in high dimensions, we performed a PCA of the data and used the first two principal components, with parameter `n.grid` set to 1000. For the model-based method `Mclust`, the constrained model (spherical cluster shape and equal volume) was assumed and the number of clusters was varied from 3 to 20.

Dataset	Result	EK-NNclus	c-means	pdfCluster	model-based (constrained)
dim256	c	16 (0)	16 (fixed)	5	16
	ARI	1.0 (0)	0.94	0.23	1
	time	1.4 (0.058)	2.76	11.30	116
dim512	c	16 (0)	16(fixed)	9	16
	ARI	1 (0)	0.94	0.5	1
	time	1.4 (0.11)	13.27	10.9	467
dim1024	c	16 (0)	16 (fixed)	8	18
	ARI	1 (0)	0.94	0.28	0.998
	time	1.4 (0.14)	36.38	11.13	23

Conclusions

The EK-NNclus algorithm generally performs better than density-based and model-based clustering procedures, especially when it comes to determining the number of clusters. It is also faster than the nonparametric density-based approach, and it performs much better with high-dimensional data. As the EK-NNclus algorithm is based on distances, it can be applied to any proximity data, and it can be kernelized to handle data with complex cluster shapes. These research directions are currently being investigated.

References

[1] T. Denœux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. on Systems, Man and Cybernetics*, 25(05):804–813, 1995.

[2] T. Denœux, O. Kanjanatarakul, and S. Sriboonchitta. EK-NNclus: a clustering procedure based on the evidential k-nearest neighbor rule. *Knowledge-based Systems (under revision)*, 2015.

[3] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 1982.

Acknowledgements

This research was supported by the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" by the National Agency for Research (reference ANR-11-IDEX-0004-02).