PSEUDOCODE FOR THEORIZED GRAVITATIONAL CLUSTERING

Original Work: Jonathan M. Rodriguez

Unpublished

Dates back July 2014

Dataset:

Set of Points Separated by Euclidean Distances

Input Parameters:

ε, epsilon

F, Force

1. Determine the **Influence** ($\boldsymbol{I}_{P,\varepsilon}$), within the reach ε, of each point in the dataset.

Influence ($\boldsymbol{I}_{P,\varepsilon}$) is defined as

$$\boldsymbol{I}_{P,\varepsilon} = \sum \frac{1}{d_{i,\varepsilon}}$$

where

ε	- is the neighbourhood distance or reach of point P where the Influence is to be measured
$d_{i,\varepsilon}$	- is the distance between point P and the i_{th} neighbour point within reach, ε, of point P

2. Determine the **Force ($\boldsymbol{F}_P$)** between all pairs of points in the dataset.

Force ($\boldsymbol{F}_P$) is defined as

$$\boldsymbol{F}_P = \frac{(I_{P_1,\varepsilon})(I_{P_2,\varepsilon})}{D_{P1P2}}$$

where

D_{P1P2}	- is the distance between points P_1 and P_2

3. Cluster the points, i.e. a pair of points with sufficient ***F*** acting between them belong to the same cluster.

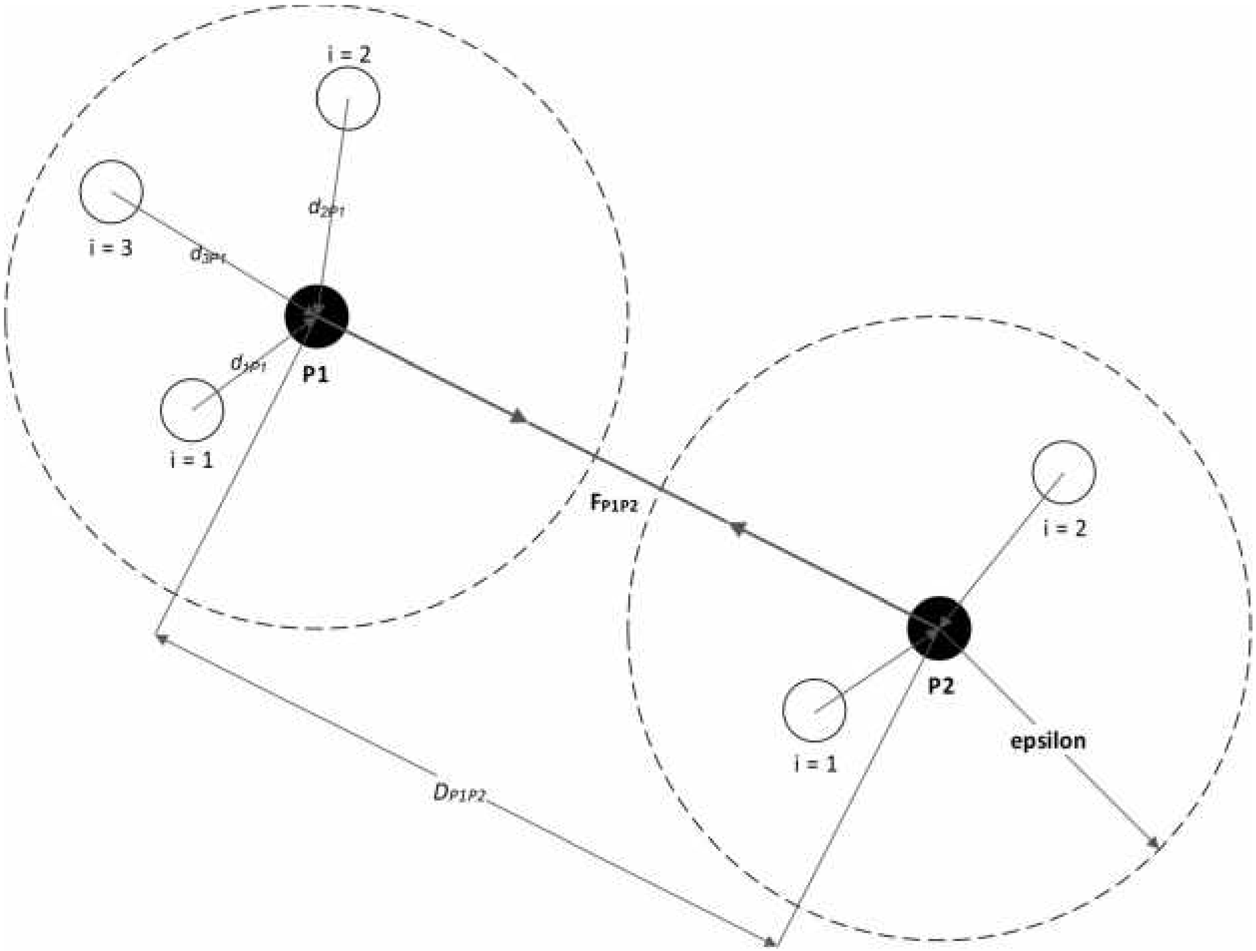

Figure 1: Illustration of the Proposed Algorithm*

*****Points with no neighbours within a specified ε or singletons produced with not enough F_P are considered as "noise"

The proposed algorithm primarily draws inspiration from the physical laws of interaction such as Coulomb's law and Newton's law, and from the clustering algorithms DBSCAN and other forms of density-based clustering or attraction-based clustering and Jarvis-Patrick Clustering. The issue of the physical inverse square law, applied in Coulomb's law and Newton's law, may not be exactly relevant to the proposed clustering method but the concept is not ruled out.

The algorithm, although may at some cases result to clustering similar to hierarchical clustering methods, is not hierarchical, either agglomerative or divisive. It does not produce a dendogram in the process of clustering per se.

Complexity Analysis

1. Determination of **Influence** ($I_{P,\varepsilon}$), Worst Case if ε encompasses all points

Visit "n" points	O(n)
Through all other "n – 1" points	O(n)
(Computation of I) Arithmetic Operation running in Constant Time	O(1)
Big Oh Rule of Products	O(n*n)
Complexity	***O(n^2)***

2. Determination of **Force (F_P),** Worst Case if redundancy is ignored

Visit "n" points	O(n)
Through all other "n – 1" points	O(n)
(Computation of F) Arithmetic Operation running in Constant Time	O(1)
Big Oh Rule of Products	O(n*n)
Complexity	***O(n^2)***

3. Clustering

"If" Statement	O(1)
Complexity	***O(1)***

The sequence of the clustering algorithm leads to:

Step 1 Complexity	O(n^2)
Step 2 Complexity	O(n^2)
Step 3 Complexity	O(1)
Big Oh Rule of Sums	O(n^2) + O(n^2) + O(1) = O(max(n^2, n^2, 1))
Complexity	**O(n^2)**