Spherical Randomized Gravitational Clustering

Jonatan Gomez[1] and Elizabeth Leon[2]

[1] ALIFE Research Group, Computer Systems, Universidad Nacional de Colombia
jgomezpe@unal.edu.co
[2] MIDAS Research Group, Computer Systems, Universidad Nacional de Colombia
eleonguz@unal.edu.co

Abstract. Circular data, i.e., data in the form of 'natural' directions or angles are very common in a number of different areas such as biological, meteorological, geological, and political sciences. Clustering circular data is not an easy task due to the circular geometry of the data space. Some clustering approaches, such as the spherical k-means, use the cosine distance instead of the euclidean distance in order to measure the difference between points. In this paper, we propose a variation of the randomized gravitational clustering algorithm in order to deal with circular data. Basically, we use the cosine distance, we modify the gravitational law in order to use the cosine distance and we use geodesics ('straight' lines in curved spaces) in order to move points according to the gravitational dynamic. Our initial experiments indicate that the spherical gravitational clustering algorithm is able to find clusters in noisy circular data.

1 Introduction

Circular data, i.e., data in the form of 'natural' directions or angles, are observations taken on compact Riemannian manifolds (curved spaces following a Riemannian geometry) [1]. Circular data are seen in different scientific areas: wave directions in oceanography, directions of animal movement in biology, wind directions in meteorology, rock fracture orientations in geology, periodic time in economy and conformational angles obtained from the $3D$ coordinates of the backbone chain of a protein in bio-informatics [1,2,3]. Modeling circular data (in particular using machine learning or statistical approaches) is not an easy task due to the curved geometry of the data space (Riemannian geometry). In particular, some approaches in the directional statistics field (the field of statistics that deals with circular data) consider the circular data as data drawn from a set of distributions of von Mises-Fisher (vMF) [4]. Some clustering approaches (approaches that try to find groups of similar points) in the data mining and machine learning fields replace the euclidean distance with the cosine distance (angular distance between points) in order to measure the difference between points in the Riemannian space. Such is the case of the spherical k-means [5].

Gomez et al. in [6,7] proposed a clustering algorithm (the randomized gravitational clustering algorithm, RGC), which is robust to noise and unsupervised in the number of cluster, based on concepts of field theory in physics . Basically,

the gravitational dynamics is determined by moving points (in an euclidean space) according to the gravitational field generated by other point randomly selected. In this paper, we propose a variation of such randomized gravitational clustering algorithm in order to deal with circular data. In this way, we replace the euclidean distance with the cosine distance, we modify the gravitational law in order to use the cosine distance and we use geodesics, the 'straight' lines in curved spaces, in order to move points according to the gravitational dynamic. This paper is divided in 5 Sections. Section 2 sumarizes the Rgc algorithm proposed by Gomez et al in [6,7]. Section 3 explains the changes introduced to the Rgc algorithm in order to deal with circular datasets using a cosine distance. Section 4 analyzes preliminar results obtained with Rgc. Finally, Section 5 outlines some conclusions.

2 Randomized Gravitational Clustering (RGC)

Gravitational clustering (**GC**) algorithms are considered agglomerative hierarchical algorithms based on concepts of field theory in physics [8,9]. The GC algorithm simulates the gravitational system obtained of considering data points as initial particles with mass equal one in a space exposed to gravitational fields. Gomez et al. in [6,7] proposed a GC algorithm which is robust to noise and unsupervised in the number of clusters (see Algorithm 1). Basically, points do not change in mass and are not removed during the simulation of the system dynamic. Two points are merged into virtual clusters using a union-disjoint set structure [10] (lines 1-2, 7 and 10, i.e., functions MAKE, FIND and UNION), when they are close enough (line 7). Gomez et al. estimated the greatest minimal separation ($\hat{d}$) between N uniformly separated points[3], (here N is the size of the data set), using the 2-dimensional hexagonal packing of circles approach [12] and used it as re-normalization factor to reduce the effect of the data set size in the system dynamic. Moreover, instead of considering all points to move a point, just another point is randomly selected and both points are moved (line 6) according to an oversimplified Universal Gravitational ($\overrightarrow{F_{x,y}} = G\overrightarrow{d_{x,y}} \left(\frac{\hat{d}}{\left\| \overrightarrow{d_{x,y}} \right\|} \right)^3$) and Second Newton's Motion Laws ($y_{t+1} = y_t + G\overrightarrow{d_{x,y}} \left(\frac{\hat{d}}{\left\| \overrightarrow{d_{x,y}} \right\|} \right)^3$), here $\overrightarrow{d_{x,y}}$ is the vector ('straight' line between points y and x). These equations are the vectorial representation of the Newton Gravitational and Second Laws[13,14]. The big crunch effect (one single big cluster at the end) is eliminated by introducing a cooling mechanism similar to the one used in simulated annealing (line

[3] The problem of determining the optimal arrangement of points in such a way that the greatest minimal separation between points is obtained, is an open problem in Geometry [11].

Algorithm 1 Randomized Gravitational Clustering

RGC(x, G, $\triangle(G)$, M, ε)

x: Data set, G: Gravity strength,$\triangle(G)$: Cooling factor, M: Iterations, ε: Fusion distance

1. **for** i=1 **to** N **do** *//Creates the Union-Disjoint cluster set*
2. MAKE(i) *//Each data points is initially a cluster*
3. **for** i=1 **to** M **do**
4. **for** j=1 **to** N **do**
5. k = random point index such that $k \neq j$
6. MOVE(x_j , x_k) *//Move both points using gravity motion equation.*
7. **if** $d_{x_j,x_k} \leq \varepsilon$ **then** UNION(j, k) *//Merges (virtually) to clusters*
8. G = (1-$\triangle(G)$)*G *//Reduces the gravity strength*
9. **for** i=1 **to** N **do** *//Canonical Union-Disjoint clusters set*
10. FIND(i)
11. **return** disjoint-sets

8). When the simulation is terminated, clusters are extracted if they have a minimum number of points (α). Finally, Gomez et al. determine an appropriated value of G by using an extended bisection search algorithm [10]: the number of clustered points q_M (points that were assigned to some cluster with two or more points), after some checking iterations of the RGC algorithm M, is used as indicator of the quality G, by comparing it against an expected value $Q \pm \tau$.

3 Spherical Randomized Gravitational Clustering (SGC)

Two elements should be considered to apply the RGC algorithm to circular data: (i) the system dynamic (gravitational field definition and movement of points in hyper-sphere surface) and (ii) the greatest minimal separation between 'uniformly' separated points in the surface of the hyper-sphere.

3.1 Spherical Gravity Law (Gravity Law in the $n-$sphere surface)

Although classic Newton Gravitational Law is defined for Euclidean spaces (spaces described by Euclidean geometry), it has been generalized to Curved spaces (spaces described by Riemannian geometry) such as the surface of an hyper sphere (in short $n-$sphere surface) [13,14]. In such curved spaces (the $n-$sphere surface), a 'straight' line becomes an arc (segment of circle) and is called **geodesic**. Since there is a one to one correspondence between the chord (the Euclidean straight line between points in a $n-$sphere surface) and the geodesic between them, and any Curved space behaves locally as an Euclidean space (it is a manifold), it is possible to approximate the motion of a point due to Gravitational Law and Newton motion law by using a cosine distance, the Euclidean straight line vector and projecting the obtained point to the $n-$sphere surface (renormalizing). In this way, the moving function of a point y due to the gravitational field of other point x (line 6 in Algorithm 1), is computed using Equation 1.

$$\overrightarrow{y_{t+1}} = normalize\left(\overrightarrow{y_t} + G\overrightarrow{x-y}\left(\frac{\widehat{d}}{cd_{x,y}}\right)^3\right) \quad (1)$$

where, $cd_{x,y}$ is the cosine distance between points x and y, $\overrightarrow{x-y}$ is the Euclidean straight line between points x and y, and $normalize\left(\overrightarrow{z}\right) = \frac{\overrightarrow{Z}}{\|\overrightarrow{z}\|}$ ($\|\overrightarrow{z}\|$ is the Euclidean norm of vector $\overrightarrow{z}$).

3.2 Greatest Minimal Separation between Points ($\hat{d}$)

We analize the behavior of $\hat{d}$ in the 2−sphere surface (circle in two dimensions) to find a better estimation of $\hat{d}$ when dealing with a n−sphere surface. Notice that, the maximum distance between closest points is the cosine distance defined by the angle $\frac{2\pi}{N}$. Similar behavior is observed when working in a 3−sphere surface, where, it is close to $\frac{2\pi}{\sqrt{N}}$. In this way, a rough approximation[4] of the angle defined by two closest points (of a data set of N points) in a n−sphere surface is provided by $\frac{2\pi}{\sqrt[n-1]{N}}$. Therefore, an estimation of $\hat{d}$ in circular data is given by Equation 2.

$$\widehat{d} = k * cd\left(\frac{2\pi}{\sqrt[n-1]{N}}\right) \quad (2)$$

Here N is the number of data points in the n-sphere surface, cd is the cosine distance and k is a correction factor due to high dimensionality of the data set (in our case we set it to 2).

4 Experiments

We use the SGC algorithm for finding clusters in different real and synthetic circular data sets. Due to the lack of space, we show (as proof of concept) the results obtained by SGC on two 3D synthetic data sets with six clusters, each cluster following a vMF distribution with different concentration parameter and number of samples. The first data set is free of noise while the second one contains 20% of noise. In this paper, we fixed $Q = \left\lfloor \frac{\sqrt{N}}{2} \right\rfloor$, $M = 1$ and $\tau = \sqrt{2Q}$ for estimating the value of G, since these values are good approximations to the ones proposed by Gomez et al in [7] and reduce the time complexity of this estimation algorithm to lineal $O(N)$ respect to the number of data points.

Figure 1 shows the evolution of the SRG on the clean synthetic data set after 25, 50 and 100 iterations. When noisy points are not part of the data set, while points are moved to their clusters centers (lower row Figure 1), clusters are formed quickly and points are not asigned to incorrect clusters (upper row Figure 1). The algorithm stops at iteration 119 when no more clusters can be formed due to the cooling factor. Clearly, the SGC algorithm is able to find cluster in clean circular data sets.

[4] By using the manifold property of the hyper-sphere surface, i.e. considering local vecinities of the hyper-sphere surface as hyper-cubes.

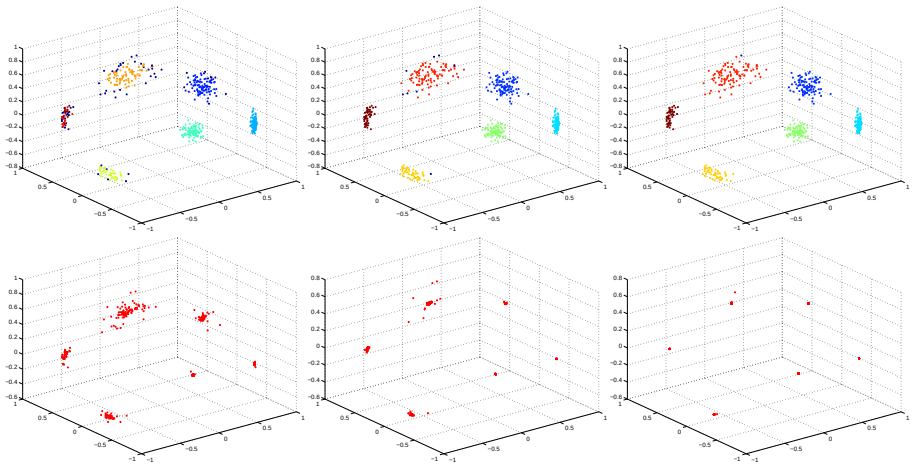

Fig. 1. Evolution of the SGC algorithm on the clean 3D circular data set: Set of clusters (upper row) obtained by SGC and position of the moving points (lower row) after 25 iterations (left), after 50 iterations (middle), and after 100 iterations (right).

Figure 2 shows the evolution of the SRG on the noisy synthetic data set after 25, 50 and 100 iterations. When noisy points are part of the data set, while noisy points are maintained at their original positions, 'good' points are moved to their clusters centers (lower row Figure 2), clusters are formed quickly (without including noisy points) and points are not asigned to incorrect clusters (upper row Figure 2). The algorithm stops at iteration 127 when no more clusters can be formed due to the cooling factor, so noisy points can be removed since they form cluster with size equal one 1. Clearly, the SGC algorithm is able to find cluster in noisy circular data sets.

5 Conclusions

Mining circular data (data in the form of 'natural' directions or angles) is a challenging task due to the curve geometry of the data space. However, it is possible to accomplish this task by considering the clustering process as the result of a dynamic system, in particular, the result of a gravitational dynamic system. In this direction, we were able to generalize the Newton gravitational law and Newton Second Motion law to curved space (like the surface of an hyper-sphere) in order to use the cosine distance instead of the euclidean distance for simulating such dynamic system in a curved space. Our results indicate that the Spherical Gravitational Clustering algorithm is able to find clusters in curved spaces and in the presence of noise. Our future work will concentrate in using the Sgc on analysis of protein structure and in higher dimensional spaces.

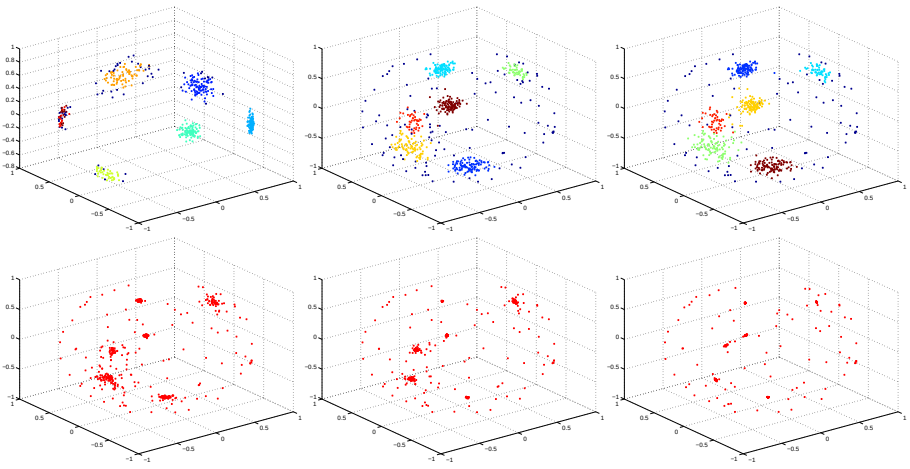

Fig. 2. Evolution of the SGC algorithm on the noisy 3D circular data set: Set of clusters (upper row) obtained by SGC and position of the moving points (lower row) after 25 iterations (left), after 50 iterations (middle), and after 100 iterations (right).

References

1. F. Wang, *Space and Space-Time Modeling of Directional Data.* PhD thesis, Duke University, 2013.
2. K. Mardia, C. Taylor, and G. Subramaniam, "Protein bioinformatics and mixtures of bivariate von mises distributions for angular data," *Biometrics*, vol. 63, no. 2, pp. 505–512, 2007.
3. I. S. Dhillon and S. Sra, "Modeling data using directional distributions," tech. rep., The University of Texas at Austin, 2003.
4. K. Mardia and P. Jupp, *Directional Statistics.* Jhon Wiley and Sons, 2000.
5. K. Hornik, I. Feinerer, M. Kober, and C. Buchta, "Spherical k-means clustering," *Journal of Statistical Software*, vol. 50, pp. 1–22, 9 2012.
6. J. Gomez, D. Dasgupta, and O. Nasraoui, "A new gravitational clustering algorithm," in *Proceedings of the Third SIAM International Conference on Data Mining 2003*, pp. 83–94, 2003.
7. J. Gomez, O. Nasraoui, and E. Leon, "Rain: Data clustering using randomized interactions between data points," in *Proceedings of the Third International Conference on Machine Learning and Applications (ICMLA 2004)*, pp. 250–255, 2004.
8. W. E. Wright, "Gravitational clustering," *Pattern Recognition*, no. 9, pp. 151–166, 1977.
9. S. Kundu, "Gravitational clustering: a new approach based on the spatial distribution of the points," *Pattern Recognition*, no. 32, pp. 1149–1160, 1999.
10. T. Cormer, C. Leiserson, and R. Rivest, *Introduction to Algorithms.* McGraw Hill.
11. H. T. Croft, K. J. Falconer, and R. K. Guy, *Unsolved Problems in Geometry.* New York: Springer-Verlag, 1991.
12. H. Steinhaus, *Mathematical Snapshots, 3rd ed.* New York: Dover, 1999.
13. M. C. Hazewinkel, *Theory of Gravitation.* Springer, 2001.
14. T. Padmanabhan, *Gravitation: foundations and Frontier.* Cambridge University Press, 2010.