Discovering Clusters in Spatial Data Using Swarm Intelligence

Gianluigi Folino, Agostino Forestiero, and Giandomenico Spezzano

Institute for High Performance Computing and Networking(ICAR) – CNR
Via P. Bucci 41C I-87030 – Rende (CS), Italy
{folino, forestiero, spezzano}@icar.cnr.it

Abstract. This paper presents a novel algorithm that uses techniques adapted from models originating from biological collective organisms to discover clusters of arbitrary shape, size and density in spatial data. The algorithm combines a smart exploratory strategy based on the movements of a flock of birds with a shared nearest-neighbor clustering algorithm to discover clusters in parallel. In the algorithm, birds are used as agents with an exploring behavior foraging for clusters. Moreover, this strategy can be used as a data reduction technique to perform approximate clustering efficiently. We have applied this algorithm on synthetic and real world data sets and we have measured, through computer simulation, the impact of the flocking search strategy on performance.

1 Introduction

Data mining deals with the problem of extracting interesting associations, classifiers, clusters, and other patterns from data by paying careful attention to the available computing, storage, communication, and human resources. Clustering is a data mining task concerning the process of grouping similar objects according to their distance, connectivity, or their relative density in space. In particular, spatial data mining is the discovery of interesting relationships and characteristics that may exist implicitly in spatial data. Spatial clustering has been an active area of research into data mining, with many effective and scalable clustering methods developed. These methods can be classified into partitioning methods, hierarchical methods, density-based methods, grid-based methods. Han, Kamber and Tung's paper [1] is a good introduction to this subject.

Recently, other algorithms based on biological models [2,3] have been introduced to solve the clustering problem. These algorithms are characterized by the interaction of a large number of simple agents that sense and change their environment locally. Furthermore, they exhibit complex, emergent behavior that is robust with respect to the failure of individual agents. Ants colonies, flocks of birds, termites, swarms of bees etc. are agent-based insect models that exhibit a collective intelligent behavior (*swarm intelligence*) [4] which may be used to define new algorithms of clustering.

In this paper, we present the parallel spatial clustering algorithm SPARROW-SNN (***SPA****tial Cluste****R****ing Algo****R****ithm thr****O****ugh S****W****arm Intelligence and* ***S****hared* ***N****earest-****N****eighbor Similarity*) which is based on an adaptive

W. Banzhaf et al. (Eds.): ECAL 2003, LNAI 2801, pp. 598–605, 2003.

flocking algorithm proposed by Macgill and S. Openshaw [5] as a form of effective search strategy to perform an exploratory geographical analysis. The algorithm takes advantage of the parallel search mechanism a flock implies, by which if a member of a flock finds an area of interest, the mechanics of the flock will draw other members to scan that area in more detail. SPARROW-SNN combines the flocking algorithm with a shared nearest neighbor cluster algorithm to discover clusters of arbitrary density, shape and size in spatial data. SPARROW-SNN uses the stochastic and exploratory principles of a flock of birds to detect clusters in parallel according to the shared nearest neighbor-based principles of the SNN [6] clustering algorithm and a parallel iterative procedure to merge the clusters discovered. Moreover, we have applied this strategy as a data reduction technique to perform approximate clustering efficiently [7]. We have built a SWARM [8] simulation of SPARROW-SNN to investigate the interaction of the parameters that characterize the algorithm. The first experiments show encouraging results and a better performance of SPARROW-SNN in comparison with the linear randomized search. The remainder of this paper is organized as follows. Section 2 briefly presents the heuristics of the SNN clustering. Section 3 introduces the classical flocking algorithm and presents the SPARROW-SNN algorithm. Section 4 discusses the obtained results and Section 5 draws some conclusions.

2 The SNN Clustering Algorithm

SNN is a clustering algorithm developed by Ertöz, Steinbach and Kumar [6] to discover clusters with differing sizes, shapes and densities in noisy, high dimensional data. The algorithm extends the nearest-neighbor non-hierarchical clustering technique developed by Jarvis-Patrick [9] redefining the similarity between pairs of points in terms of how many nearest neighbors the two points share. Using this new definition of similarity, the algorithm eliminates noise and outliers, identifies representative points also called *core points*, and then builds clusters around the representative points. These clusters do not contain all the points, but rather represent relatively uniform groups of points. The SNN algorithm starts performing the Jarvis-Patrick scheme. In the Jarvis-Patrick algorithm a set of objects is partitioned into clusters on the basis of the number of shared nearest-neighbors. The standard implementation is constituted by two phases. The first is a pre-processing stage which identifies the K nearest-neighbors of each object in the data set. In the subsequent clustering stage a shared nearest neighbor graph is constructed from the pre-processed data as follows. A link is created between two objects i and j if:

- i is one of the K nearest-neighbors of j;
- j is one of the K nearest-neighbors of i;
- i and j have at least K_{min} of their K-nearest-neighbors in common;

where K and K_{min} are used-defined parameters. Each link has an associate weight defined as:

$$weight(i,j) = \sum (k+1\text{-}m)(k+1\text{-}n),\ where\ \ i_m = j_n \tag{1}$$

In the equation above, k is the nearest neighbor list size, m and n are the positions of a shared nearest neighbor in i and j's lists. At this point, clusters can be obtained by removing all edges with weights less than a user specified threshold and taking all the connected components as clusters. A major drawback of the Jarvis-Patrick is that, the threshold needs to be set high enough since two distinct sets of points can be merged into the same cluster even if there is only one link across them. On the other hand, if a high threshold is applied, then a natural cluster will be split into many small clusters due to the variations in the similarity in the cluster. SNN addresses these problems adding to the Jarvis-Patrick algorithm the following steps:

- for every data point in the weighted graph, calculate the total sum of weights associated with the links coming out of the point. This value is called *connectivity*;
- identify core points by choosing the point that have a value of connectivity greater than a predefined threshold (*core_threshold*);
- identify noise points by choosing the points that have a value of connectivity lower than a user specified threshold (*noise_threshold*) and remove them;
- remove all links between points with weight smaller than a threshold);
- form clusters with the connected components of points. Every point in a cluster is either a core point or is connected to a core point.

The number of clusters is not given to the algorithm as a parameter. Also note that not all the points are clustered.

3 SPARROW-SNN: A Flocking Algorithm for Spatial Clustering

In this section, we present a multi-agent clustering algorithm, called SPARROW-SNN, which combines the stochastic search of an adaptive flocking with the SNN heuristics for discovering clusters in spatial data. This approach has a number of nice properties. It has the advantages of being easily implementable on parallel computers and is robust compared to the failure of individual agents. It can also be applied to perform *approximate clustering* efficiently since the points that are, to each iteration, visited and analyzed by the agents represent a significant (in *ergodic* sense) subset of the entire data set. The subset reduces the size of the data set while keeping the loss of accuracy as small as possible. We propose to use flocking *sampling* as a data reduction technique to speed up the operations of cluster and outlier detection on large data sets collections. Our approach iteratively improves the accuracy of the clustering because, at each generation, new data points are discovered and added to each cluster with about the same increase per cent.

SPARROW-SNN uses a modified version with an *exploring* behavior of standard Reynolds' flock of birds model [10] to describe the movement rules of the

agents. The behavior requires each agent to search the clusters in parallel and to signal the presence or the lack of significant patterns in the data to the other flock members, by changing its color. The entire flock then moves towards the agents (*attractors*) that have discovered interesting regions to help them, avoiding the uninteresting areas that are instead marked as obstacles. Clusters are discovered using the heuristics principles of the SNN clustering algorithm.

As first step, SPARROW-SNN computes the nearest-neighbor list for each data point using a *threshold similarity* that reduces the number of data elements to take in consideration. The introduction of the threshold similarity produces variable-length nearest-neighbor lists and therefore now i and j must have at least P_{min} of the shorter nearest-neighbor list in common; where P_{min} is a user-defined percentage. After the nearest-neighbor list is computed, SPARROW-SNN starts a fixed number of agents that will occupy a randomly generated position. The agents have an attribute that defines their color. Initially the color is the same for all. From its initial position, each agent moves around the spatial data testing the neighborhood of each location in order to verify if the point can be identified as a core point. All agents execute the same set of rules for a fixed number of times (*MaxGenerations*). When an agent falls on a data point A not yet visited, it computes the connectivity, *conn[A]*, of the point, i.e. computes the total number of strong links the points has according to the rules of the SNN algorithm. Points having a connectivity smaller than a fixed threshold (*noise_threshold*) are classified as noise and are considered to be removed from the clustering. Each agent is colored on the basis of the connectivity computed in the visited data point. The colors assigned to the agents are: **red** (*conn* > *core_threshold*), revealing core points, **green** (*noise_threshold* < *conn* <= *core_threshold*), for border points, **yellow** (*0* < *conn* < *noise_threshold*), for noise points, and **white** (*conn* = *0*), indicating an obstacle (uninteresting region).

After the coloration step, the green and yellow agents, compute their movement observing the positions of all other agents that are at some fixed distance (*dist_max*) from them, and applying the rules of Reynolds' with the following modifications:

- *Alignment* and *cohesion* do not consider yellow agents, since they move in a not very attractive zone.
- *Cohesion* is the resultant of the heading towards the average position of the green flockmates (*centroid*), of the attraction towards red agents, and of the repulsion by white agents.
- A *separation* distance is maintained from all the agents, whatever their color is.

Agents will move towards the computed destination with a speed depending from their color: green agents will move slowly than yellow agents since they will explore denser zones of clusters. Green and yellow agents have a variable speed, with a common minimum and maximum for all agents. An agent will speed up to leave an empty or uninteresting region whereas it will slow down to investigate an interesting region more carefully. The variable speed introduces an adaptive

behavior in the algorithm. In fact, agents adapt their movement and change their behavior (speed) on the basis of their previous experience represented by the red and white agents. Red and white agents will stop signaling to the others the interesting and desert regions.

Note that for any agent which has become red or white, a new agent will be generated in order to maintain a constant number of agents exploring the data. In the first case, the new agent will be generated in a close random point, since the zone is considered interesting, while in the latter it will be generated in a random point over all the space.

In any case, each red agent (placed on a representative point) will run the merge procedure so that it will include, in the final cluster, the representative point discovered together with the points that share with it a significant (greater than P_{min}) number of neighbors and are not noise points. The merging phase considers two different cases: when we have never visited any of these points in the neighborhood and when we have points belonging to different clusters. In the first case, the points will be assigned the same temporary label and will constitute a new cluster; in the second case, all the points will be merged into the same cluster, i.e. they will get the label corresponding to the smallest one. So clusters will be built incrementally.

During simulations a *cage effect*, was observed; in fact, some agents could remain trapped inside regions surrounded by red or white agents and would have no way to go out, wasting useful resources for the exploration. So, a limit on their life was imposed to avoid this effect; hence, when their age exceeded a determined value (*maxLife*) they were killed and were regenerated in a new randomly chosen position of the space.

4 Experimental Results

For the experiments we used two synthetic data sets and one real life data set from a spatial database. The structure of these data sets is shown in figure 1. The first data set, called GEORGE, consists of 8000 points, characterized by a large number of noise points. The second data set, called DS1, contains 8000 points and presents different densities in the clusters. The third data set, called North-East, contains 123593 points representing postal addresses of three metropolitan areas (New York, Boston and Philadelphia) in the North East States.

We implemented our algorithm using SWARM, a multi-agent software platform for the simulation of complex adaptive systems. We first illustrate the loss of accuracy of our SPARROW-SNN algorithm in comparison with SNN algorithm when SPARROW-SNN is used as a technique for approximate clustering. To this purpose, we implemented a version of SNN and we computed the number of clusters and the number of points for cluster for the three datasets.Table 1 presents a comparison of these results with respect to the ones obtained from SPARROW-SNN when a population of 50 agents have visited respectively 7%, 12% and 22% of the entire data set.

Fig. 1. The three data sets used in our experiments.

Table 1. Number of clusters and number of points for cluster for GEORGE, DS1 and North-East data sets (percentage in comparison to the total points for cluster found by SNN) when SPARROW-SNN analyzes 7%, 12% and 22% points.

Clustering using the GEORGE data set	Per. of data points for cluster found by SPARROW-SNN		
	7%	12%	22%
G	55.8%	80.0%	88.3%
E	53.1%	69.8%	88.6%
O	62.8%	82.1%	86.2%
R	49.0%	67.5%	83.2%
G	47.9%	72.0%	78.8%
E	64.2%	77.7%	86.8%

Clustering using the DS1 data set	Per. of data points for cluster found by SPARROW-SNN		
	7%	12%	22%
1	41.35%	58.31%	70.37%
2	30.08%	53.58%	60.72%
3	29.28%	40.99%	53.02%
4	20.9%	30.5%	51.41%
5	53.38%	65.36%	76.56%
6	56.89%	69.87%	73.63%
7	33.89%	43.5%	61.58%

Clustering using the North-East data set	Per. of data points for cluster found by SPARROW-SNN		
	7%	12%	22%
Philadelfia	42.5%	65.2%	79.4%
New York	38.7%	52.3%	67.6%
Boston	46.5%	68.6%	82.3%

Note that with only 7% of points we can have a clear vision of the found clusters and with a few more points we can obtain the nearly totality of the points. This trend is well marked in GEORGE and in North-East data sets. For DS1 data set the results are not so clear because the 3 and 4 clusters have very few points, so they are very hard to discover. For the real data set we only reported the results for the three main clusters representing the towns of Boston, New York and Philadelphia. We can explain the good results through the

adaptive search strategy of SPARROW-SNN that requires the individual agents to first explore the data searching for representative points whose position is not known *a priori*. Then, after the representative points are located, all the flock members are steered to move towards the representative points, that represent the interesting regions, in order to help them, avoiding the uninteresting areas that are instead marked as obstacles and adaptively changing their speed.

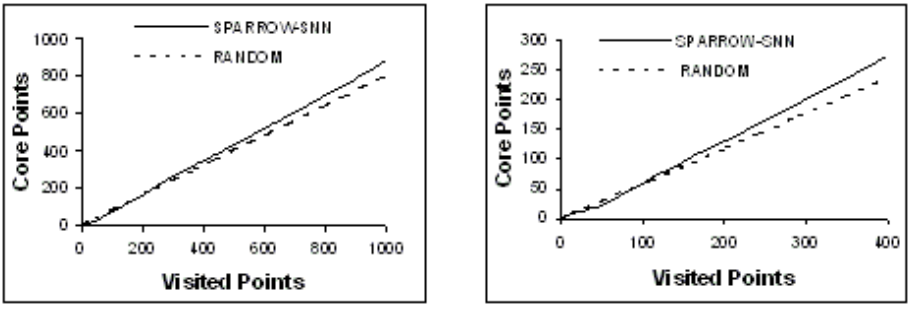

Fig. 2. Number of representative points found with SPARROW-SNN vs. total number of visited points for GEORGE and North-East datasets.

To verify the effectiveness of the search strategy we have compared SPARROW-SNN with the random-walk search strategy. Figure 2 shows, for the data sets GEORGE and North-East, the number of core points found with SPARROW-SNN and those found with the random search vs. the total number of visited points. This figure reveals that the number of core points discovered at the beginning (110 visited points for George data set and 200 for North-East data set) from the random strategy is slightly higher than the number discovered by SPARROW-SNN.

Subsequently, our strategy presents a superior behavior on the random search strategy because of the adaptive behavior of the algorithm that allows the agents to learn on their previous experience. A similar behavior has been observed for the DS1 dataset.

5 Conclusions

In this paper, we have described the parallel clustering algorithm SPARROW-SNN, which is based on the use of swarm intelligence techniques. The algorithm combines a smart exploratory strategy based on a flock of birds with a shared nearest neighbor clustering algorithm to discover clusters of arbitrary shape, size and density in spatial data. The algorithm has been implemented in SWARM and evaluated using two synthetic data sets and one real word data set. Measures of accuracy of the results show that SPARROW-SNN can be efficiently applied as a data reduction strategy to perform approximate clustering. Moreover, the

adaptive search strategy of SPARROW-SNN is more efficient than that of the random-walk search strategy.

Acknowledgements. This work was supported by the CNR/MIUR project-Legge 449/97-DM 30/10/2000.

References

1. Han J., Kamber M., Tung A.K.H., Spatial Clustering Methods in Data Mining: A Survey, H. Miller and J. Han (eds.), Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2001.
2. Lumer E. D., Faieta B., Diversity and Adaptation in Populations of Clustering Ants, Proc. of the third Int. Conf. on Simulation of Adaptive Behavior: From Animals to Animats (SAB94), D. Cliff, P. Husbands, J.A. Meyer, S.W. Wilson (Eds), MIT-Press, pp. 501–508, 1994.
3. N. Monmarché, M. Slimane, and G. Venturini, "On improving clustering in numerical databases with artificial ants", in Advances in Artificial Life: 5th European Conference, ECAL 99, LNCS 1674, Springer, Berlin, pp. 626–635, 1999.
4. Bonabeau E., Dorigo M., Theraulaz G., Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, 1999.
5. James Macgill, Using Flocks to Drive a Geographical Analysis Engine, Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life, MIT Press, Reed College, Portland, Oregon, pp. 1–6, 2000.
6. Ertöz L., Steinbach M., and Kumar V., A New Shared Nearest Neighbor Clustering Algorithm and its Applications, Workshop on Clustering High Dimensional Data and its Applications, 2nd SIAM International Conference on Data Mining Arlington, VA, 2002.
7. Kollios G., Gunopoulos D., Koudas N., Berchtold S.: "Efficient Biased Sampling for Approximate Clustering and Outlier Detection in Large Datasets". IEEE Trans. on Knowledge and Data Engineering (TKDE), to appear, 2003.
8. Minar, N., Burkhart, R., Langton, C. and Askenazi, M., http://www.santafe.edu/projects/swarm, 1996.
9. 11. R. A. Jarvis and E. A. Patrick, Clustering using a similarity measure based on shared nearest neighbors, IEEE Transactions on Computers, C-22(11), 1973.
10. Reynolds C. W., Flocks, Herds, and Schools: A Distributed Behavioral Model, Computer Graphics vol. 21, n. 4, , (SIGGRAPH 87), pp. 25–34, 1987. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)