Flexible Grid-Based Clustering

Marc-Ismaël Akodjènou-Jeannin, Kavé Salamatian, and Patrick Gallinari

LIP6 - Université Paris 6 Pierre et Marie Curie
104 avenue du Président Kennedy, Paris, France
{Marc-Ismael.Akodjenou,Kave.Salamatian,Patrick.Gallinari}@lip6.fr

Abstract. Grid-based clustering is particularly appropriate to deal with massive datasets. The principle is to first summarize the dataset with a grid representation, and then to merge grid cells in order to obtain clusters. All previous methods use grids with hyper-rectangular cells. In this paper we propose a flexible grid built from arbitrary shaped polyhedra for the data summary. For the clustering step, a graph is then extracted from this representation. Its edges are weighted by combining density and spatial informations. The clusters are identified as the main connected components of this graph. We present experiments indicating that our grid often leads to better results than an adaptive rectangular grid method.

1 Introduction

With the ever-increasing amount of storage and processing capacities, huge datasets are now common in many areas : earth science, astronomy, or computer networks, just to name a few. The mining of such datasets, and especially the clustering task, calls for robust and efficient techniques. Grid-based clustering methods have been the subject of many recent studies [1,2,3].

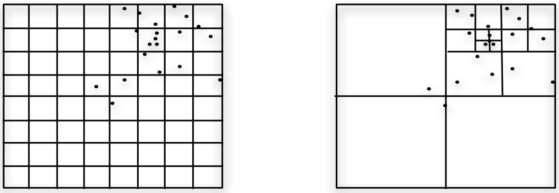


Fig. 1. Summaries of datasets. Left a regular rectangular grid. Right an adaptive hypercubic grid.

Grid-based clustering consists in clustering the space surrounding the datapoints instead of the datapoints themselves [4]. The basic idea is to cover the data space with a grid in order to construct a spatial summary of the data. Each non-empty cell of the grid is weighted by the number of original datapoints it contains (see Figure 1). The clustering is performed by aggregating adjacent dense cells to form clusters. Grid-based methods are similar to density-based clustering, but

J.N. Kok et al. (Eds.): PKDD 2007, LNAI 4702, pp. 350–357, 2007.

with local densities and neighborhood relations taking place between cells, and no longer between individual points.

In this paper, we propose a new type of grid to build the dataset summary. The cells of the grid are general polyhedra, and are not axis-aligned hypercubes or hyper-rectangles like in all existing methods. The neighborhood relation between cells is richer; hence the aggregation process (which is the base operation for clustering) is more efficient. The clustering step is performed by extracting a graph from the spatial summary, and identifying clusters as its main connected components. The edges of the graph are weighted by a similarity metric which uses both spatial and density information from the summary.

The remainder of the paper is structured as follows : Section 2 presents related work and motivations. Section 3 describes the construction of the flexible grid. Section 4 describes the clustering step and the similarity metric. Section 5 discusses complexity and sensitivity to dimensionality. Section 6 contains results of experiments and a comparison with a hypercubic adaptive grid method.

2 Related Work and Motivation

Many grid-based clustering approaches [1,3] rely on the traditional regular, hypercubic grid (Figure 1, left). The main drawback of these approaches is that the grid construction requires to cover all the data space with the same precision independently of the data density. Thus a very high resolution could be needed to obtain a satisfying spatial summary. Another class of methods [5,2] uses multi-resolution grids with size-varying hypercubic or hyper-rectangular cells (Figure 1, right). The basic idea is to cover with more precision regions with many points. Usually the clustering step follows the hierarchy of the data structure. Both sets of methods are parametrized by the resolution of the grid. The clustering step usually relies on a density threshold discarding low-density cells. The complexity of these methods is linear in the number of data points $O(N)$. The complexity of the clustering step depends only on the number of (non-empty) cells M.

The aggregation of neighbor cells is the basis for the clustering process. Since the ultimate goal is to find patterns in the original data, one wants to minimize the impact of the particular geometry of the grid on the efficiency of the aggregation process. Classical grids (be they regular or multi-resolution) have their cell borders aligned with the axes of the space; this directional bias has a strong influence on the resulting data summaries.

In this work, we propose a multi-resolution grid whose cells have randomly oriented borders. It is close to the Crack STIT tessellation model of stochastic geometry [6]. The resulting spatial summary has no particular orientation and does not suffer from the rigid geometry of hyper-rectangular tilings. The cells are general polyhedra, allowing a spatially more flexible aggregation process. For the clustering step, we extract a weighted graph from the spatial summary. We propose a similarity metric to weight edges; it takes into account both spatial and density similarities of cells. The clusters are identified as the main connected components of the graph. The complexity of the clustering step is $O(M)$. The parameters of the whole method are the size of the summary M, the number of clusters K and the minimum number of points $MinPts$ per cluster.

3 Flexible Grid

3.1 Hyperplanes and Polyhedra

We recall here simple facts about hyperplanes and polyhedra. A hyperplane in a d dimensional space $H = \{z \mid \langle u \cdot z \rangle = t\}$ is defined by its *orientation vector* $u \in \mathbb{S}^{d-1}$ and *its offset* $t \in \mathbb{R}$ ($\langle \cdot \rangle$ denotes scalar product). For a given u, a hyperplane with offset $t = \langle z_0 \cdot u \rangle$ passes through the point z_0. A uniform random hyperplane can be obtained by taking a random d-dimensional gaussian random vector, and normalizing its norm to 1. A polyhedra $P \subset \mathbb{R}^d$ admits two representations : the H-representation (set of delimiting hyperplanes) and the V-representation (convex hull of vertices). The H-representation describes P as the intersection of halfspaces defined by a set of hyperplanes $(\cap H_i^{\sigma_i})$, where $\sigma = (\sigma_1 \dots \sigma_m)$ is a binary codeword locating the point in halfspaces defined by the hyperplanes.

3.2 Construction

The principle of the construction of the multi-resolution flexible grid is simple (see Algorithm 1). It begins with the hypercube containing the data. At each step, the cell containing the largest number of points is splitted into two subcells by a random hyperplane. This process is iterated until a given number of non-empty cells M (fixed by the user) is reached. The hyperplanes are chosen with a uniform random orientation. The splitting process has a natural binary tree structure, as depicted in Figure 2. The algorithm iteratively encodes the data points into binary codewords. These binary codewords correspond to the H-representation of the cells. At the end of the algorithm, the dataset has been summarized to a set of weighted polyhedra. Each datapoint belongs to a particular cell.

The flexible grid is a particular realization of a stochastic process. It is built iteratively during the cell refinement process and automatically adapts its resolution to the local data density. Finer parts of the gird are revealed in regions

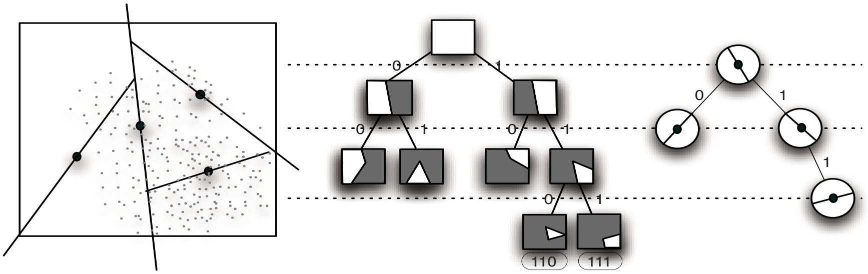


Fig. 2. Data domain, cell tree and hyperplane tree

Algorithm 1. Construction of flexible grid

```
Inputs
$X = \{x_1, \ldots, x_N\}$ dataset of $N$ points in $\mathbb{R}^d$
$D$ : hyper-rectangle containing $X$
$M$ : desired size (number of occupied cells) of the summary

Outputs
$\mathcal{S} = \{(S_1, p_1), \ldots, (S_M, p_M)\}$ set of $M$ polyhedra along with the proportion of points they contain
$NR$ =neighborhood relation between the cells

Begin
$\mathcal{S}_0 \leftarrow \{(D, 1, [])\}$ initial region containing all the points
$T \leftarrow$ empty hyperplane binary tree
$NR \leftarrow$empty list

While $| \mathcal{S}_0 | < M$
    $(C, p, w) \leftarrow$ cell of $\mathcal{S}_0$ with the maximum $p$ and with codeword $w$
    $H_{split} \leftarrow$ random hyperplane passing through the center of $C$
    $T \leftarrow$ Add hyperplane $H_{split}$ to hyperplane tree at node of binary index $w$
    $\{(C_1, p_1, w_1), (C_2, p_2, w_2)\} \leftarrow$ subcells created by splitting $C$ with hyperplane $H_{split}$
    Replace $(C, p, w)$ in $\mathcal{S}_0$ by non-empty elements of $\{(C_1, p_1, w_1), (C_2, p_2, w_2)\}$
    $NR \leftarrow$ Update neighborhood relations of the new cells replacing $C$
End

Extract $\mathcal{S}$ from $\mathcal{S}_0$
End
```

where there are many datapoints. The resulting summary has small, high-density cells in dense regions and big, low-density cells in sparsely populated regions.

4 Clustering

4.1 Graph Clustering

Graph clustering has been the subject of numerous studies (see [7]). The idea is to modelize the clustering problem by a weighted graph; the original clustering problem reduces to find clusters of vertices of the graph. In this paper, we extract the graph from the spatial summary (Figure 3). The graph representation is well suited to our problem since it allows to describe in a compact form the polyhedra (the vertices), their neighborood relation (the edges) and their similarities (edge weights). An edge links two vertices if they correspond to neighbor cells. Two cells are neighbors if they have a $(d-1)$-dimensional intersection. Edges are weighted with a similarity metric described below. We iteratively remove edges of the graph until we have K connected components, of at least $MinPts$ points each. At each step the edge with the minimum weight is chosen for removal.

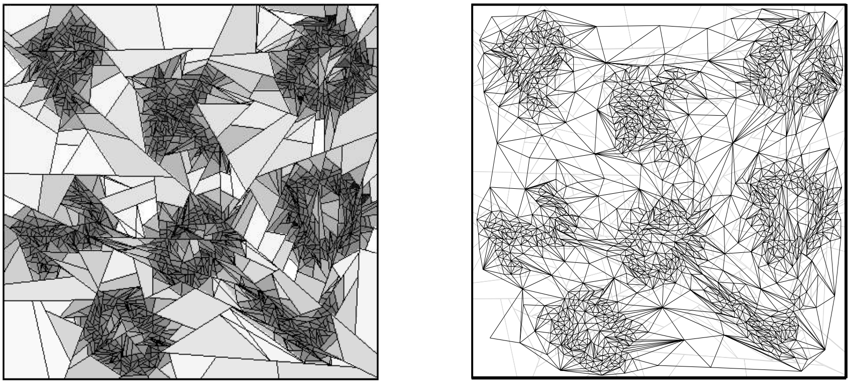


Fig. 3. (a) Flexible data summary (b) Structure of extracted graph

4.2 Similarity Metric

In the majority of previous works, the grouping of two cells is determined (implicitly or explicitly) by the closeness of the densities of the two cells. This stems from the intuitive assumption that cells at the frontier of a cluster will see a large density variation. This is not robust since even in dense regions, important density variations may appear. We propose a more robust cell grouping criterion incorporating spatial closeness between cells. Because of the multi-resolution, the distance between cell centers already conveys much information about the density of the data. The spatial information allows a smoothing of the density variations, thereby allowing better clustering results.

Given two cells with centers c_i, $c_j \in \mathbb{R}^d$, and cell density values D_i and $D_j \in \mathbb{R}$, we set the similarity between cells i and j to be $f_{sim} = f_{dens} \cdot f_{spat}$, with $f_{dens}(i,j) = exp\left(-\frac{\|D_i - D_j\|^2}{2 \cdot \sigma_{dens}^2}\right)$ and $f_{spat}(i,j) = exp\left(-\frac{\|c_i - c_j\|^2}{2 \cdot \sigma_{spat}^2}\right)$ with σ_{dens} being the mean euclidean difference between their densities, and σ_{spat} the average euclidean distance between centers of two neighbors cells of the grid. The exponentiation is the most natural way to express the similarities. The density D_i is the ratio (p_i/V_i) where p_i and V_i are respectively the proportion of points and the volume of the cell.

5 Dimensionality and Complexity

5.1 Dimensionality

Grid-based methods are well suited for small dimensional spaces. For high dimensional data, the number of grid cells and of neighbor cells increases exponentially and the methods cannot be used as such when the number of dimensions iq too high [8]. the exponential number of grid cells, and the high number of neighbor cells are highlighted as the main issues. Compared to regular rectangular grids, the multi-resolution grid and our graph clustering technique partly circumvents this phenomenon. In our case the main limitation comes from the complex structure of polyhedra : computing the volume of the polyhedra and testing which polyhedra are in its neighborhood rapidly becomes prohibitive. We propose here to approximate these two steps: instead of computing the whole neighborhood of cell, we compute the distance between all cell centers. The neighborhood of a specific cell is then defined as the set of cells whose center is among the closest centers according to the distance matrix. The volume of a cell of the grid is then approximated by the volume of a ball, the diameter of which is set to the distance to the nearest cell center. These approximations are reasonable with regard to the multi-resolution nature of the summaries and to our edge-removal graph clustering technique. This approximation does not degrade the performance of the method as will be seen in Section 6.

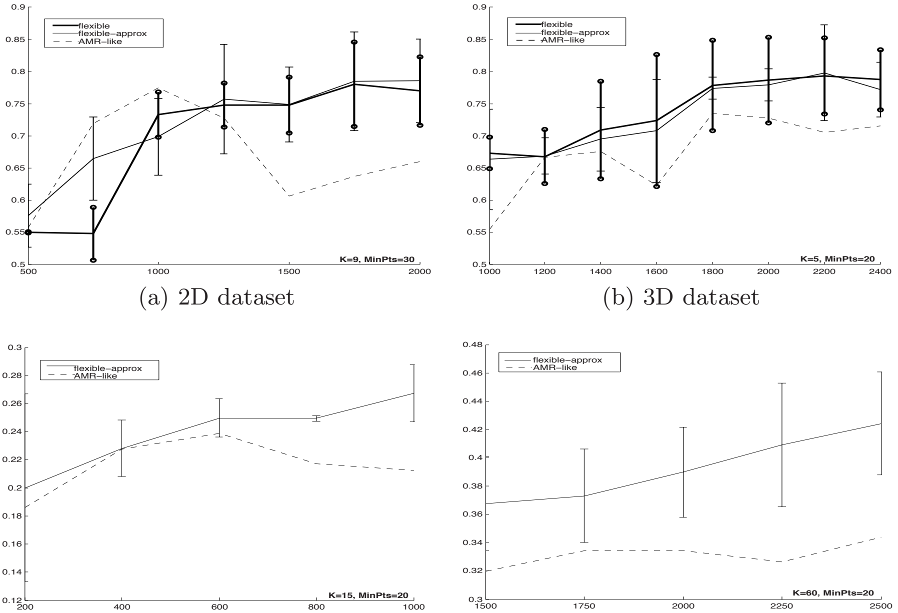
5.2 Complexity

The construction step has linear complexity in the input data size $O(N)$ (with an analysis similar to [2]). All the other steps depend only on M. Neighborhood check has complexity $O(m \cdot LP(m,d))$, $LP(m,d)$ being the complexity of a linear program with m constraints in a d-dimensional space, m depending on the polyhedra. For the clustering step, each search for connected components has a complexity linear in the size of the graph: $O(V+E)$, V and E being respectively the number of vertices and edges of the graph. It is $O(M)$ for our problem since the number of edges can be bounded by $(n_{max} \cdot M)/2$ with n_{max} the maximum number of neighbors of a node.

6 Experiments

6.1 Experimental Setting

We implemented in C++ the construction of our flexible grid, flexible grid approximation. We also implemented the AMR-like (Adaptive Mesh Refinement) grid (Figure 1 right), which is an adaptive, axis-aligned, hypercubic grid described in [9,2]. Experiments were performed with four datasets : a first complex 2D dataset of 3000 points from [10], a 3D dataset of 8000 points with five non-convex “banana” shapes, the Pageblocks database of 5400 points ($d = 10$), and a subset of 7800 points Letter Recognition database ($d = 16$) from the UCI Machine Learning Repository. For the 2D and 3D datasets, we compared the axis-aligned case (’AMR-like’), the flexible case (’flexible’) and the approximation of the flexible case described in Section 5 (’flexible-approx’). For the higher dimensional datasets we compared the flexible approximation and the AMR-like summaries. For flexible approximations, we took respectively 3,4,11 and 17 neighbors per cell for the 2D, 3D, 10D and 16D datasets (following the simple idea that a polyhedron in d dimensions has at least $(d+1)$ faces). We measured the raw performance with respect to the full original dataset with the Normalized Mutual Information criterion ([11]). Error bars show standard deviation of experiments for the flexible and flexible-approximation cases. Clustering parameters are indicated in the lower right corner of the figures.



(a) 2D dataset

(b) 3D dataset

(c) 10D dataset

(d) 16D dataset

Fig. 4. Clustering quality for growing summary sizes

6.2 Discussion

The results show that the axis-aligned summary type has a rather unpredictable behavior. The clustering performance does not always grow with the summary size : it may remain approximately constant (16D dataset) or even degrade (2D and 10D datasets). The flexible grid yields better results most of the time. The clustering performance globally grows with the resolution. Note that the flexible approximation has practically the same performance than the full flexible summary. With this approximation, the complexity of the algorithm is greatly reduced so that it could be used reasonably for dimensions up to 50.

7 Conclusion and Future Work

We have proposed a new type of grid for data summaries in the context of grid-based clustering methods. The grid is locally-adaptive and has a flexible geometry. We also proposed an approximation of this method adapted to high dimensional spaces. We have presented results indicating that the proposed grid often yields more accurate clustering results than its axis-aligned counterpart. In future work, we will incorporate the flexible grid into classical variations and improvements for grid-based methods (e.g subspace clustering [12]).

References

1. Peter, W., Chiochetti, J., Giardina, C.: New unsupervised clustering algorithm for large datasets. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, New York (2003)
2. Liao, W.K., Liu, Y., Choudhary, A.: A grid-based clustering algorithm using adaptive mesh refinement. In: 7th Workshop on Mining Scientific and Engineering Datasets of SIAM International Conference on Data Mining (2004)
3. Yu, Z., Wong, H.S.: Gca: A real-time grid-based clustering algorithm for large dataset. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR) (2006)
4. Schikuta, E.: Grid-clustering: An efficient hierarchical clustering method for very large data sets. In: 13th International Conference on Pattern Recognition (ICPR'96) (1996)
5. Schikuta, E., Erhart, M.: The bang-clustering system: Grid-based data analysis. In: Liu, X., Cohen, P.R., Berthold, M.R. (eds.) Advances in Intelligent Data Analysis. Reasoning about Data. LNCS, vol. 1280, Springer, Heidelberg (1997)
6. Nagel, W., Weiss, V.: Crack stit tessellations: characterization of stationary random tessellations stable with respect to iteration. Advances In Applied Probability 37, 859–883 (2005)
7. Brandes, U., Gaertler, M., Wagner, D.: Experiments on graph clustering algorithms. In: ESA, pp. 568–579 (2003)
8. Hinneburg, A., Keim, D.: Optimal grid-clustering towards breaking the curse of dimensionality in high-dimensional clustering. In: Proceedings of the 25th International Conference on Very Large Databases (VLDB) (1999)
9. Wang, W., Yang, J., Muntz, R.: Sting: a statistical information grid approach to spatial data mining. In: Twenty-Third International Conference on Very Large Databases (1997)
10. Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithm. In: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04), pp. 576–584. IEEE Computer Society Press, Los Alamitos (2004)
11. Strehl, A., Gosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research (JMLR) 3 (2002)
12. Agrawal, R., Gehrke, J., Gunopoulos, J., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM International Conference on Management of Data (SIGMOD '98), pp. 94–105. ACM Press, New York (1998)