A New Clustering Approach for Symbolic Data and Its Validation: Application to the Healthcare Data

Haytham Elghazel[1], Véronique Deslandres[1], Mohand-Said Hacid[2], Alain Dussauchoy[1], and Hamamache Kheddouci[1]

[1] PRISMa Laboratory, Claude Bernard University of Lyon I, 43 Bd du 11 novembre 1918, 69622 Villeurbanne cedex, France
{elghazel, deslandres, dussauchoy, hkheddou}@univ-lyon1.fr
[2] LIRIS Laboratory, Claude Bernard University of Lyon I, 43 Bd du 11 novembre 1918, 69622 Villeurbanne cedex, France
mshacid@liris.cnrs.fr

Abstract. Graph coloring is used to characterize some properties of graphs. A *b-coloring* of a graph *G* (using colors 1,2,…,k) is a coloring of the vertices of *G* such that (i) two neighbors have different colors (proper coloring) and (ii) for each color class there exists a dominating vertex which is adjacent to all other *k-1* color classes. In this paper, based on a *b-coloring* of a graph, we propose a new clustering technique. Additionally, we provide a cluster validation algorithm. This algorithm aims at finding the optimal number of clusters by evaluating the property of *color dominating vertex*. We adopt this clustering technique for discovering a new typology of hospital stays in the French healthcare system.

1 Introduction

Clustering is the process of dividing a set of given objects into groups, or clusters, such that all objects in the same group are similar to each other, while objects from different groups are dissimilar. Clustering plays an important role in data mining applications such as Web analysis, information retrieval and many other domains.

In French hospitals, the Diagnosis Related Groups (DRG) system was introduced in the eighties, by the Information Systems Medicalisation Program (PMSI). According to the PMSI process deployment, the PMSI data can also used for the assessment of performances for both public and private hospitals. This evaluation is based on the production of a Standard Discharge Summary (RSS) for each hospital stay. The RSS contains data related to the nature of treatments, medical exams and diagnosis as well as patient's information. The RSS is then identified to correspond to one patient's group called DRG, which is used for the classification of hospital stays. This operation is performed using a supervised approach according to a decision tree.

In spite of the successive improvements of DRG classification, this method remains inefficient for public and private hospitals. The major known problem of this method is that heterogeneous pathologies and examinations find themselves in the same DRG class. In this paper, we propose a new clustering approach based on graph *b-coloring* as an alternative for DRG classification: it builds a refined typology of

F. Esposito et al. (Eds.): ISMIS 2006, LNAI 4203, pp. 473 – 482, 2006.

hospital stays. A graph b-coloring consists to color the vertices of a graph with the largest number of color such that (i) two neighbors have different colors (proper coloring) and (ii) for each color class there exists a *dominating vertex* which is adjacent to all other color classes.

The paper is organized as follows: Section 2 provides summary of related work. Section 3 describes the clustering algorithm that uses as foundation a *graph b-coloring* method. Section 4 is devoted to the cluster validation algorithm. In Section 5, the application of the proposed method on *PMSI* data is illustrated. The experiments show that the algorithm is appropriate to find the optimal number of *hospital stay* groups. We conclude in Section 6 by anticipating on necessary extensions.

2 Related Work

Clustering of data is generally based on two approaches: *hierarchical and partitioning* [1]. The *hierarchical* clustering algorithms build a cluster hierarchy or, in other words, a tree of clusters, (*dendrogram*) whose leaves are the data points and whose internal nodes represent nested clusters of various sizes [2]. Hierarchical clustering methods can be further subdivided into *agglomerative* and *divisive*, depending on whether they start from the bottom or the top of the tree. The main weakness of this algorithm is a choice of the level that provides the best partition. It is often the data miner who decides about the number of clusters associated both the context and the goal of the analysis. Among these hierarchical methods, some authors have proposed to use *graph coloring techniques* for the classification purpose. In [3], the authors propose a divisive classification method based on the distance tables, where the iterative algorithm consists, at each step, in finding a partition by subdivising the class with the largest diameter into two classes in order to offer a new partition with minimal diameter. The subdivision is obtained by a *2-coloring* of the vertices of the maximum spanning tree built from the dissimilarity table.

The *partitioning* clustering algorithms give a single partition of the data by fixing some parameters (number of clusters, thresholds, etc.). The partitioning algorithm typically starts with an initial partition of data set and then uses an iterative control strategy to optimize an objective function. Each cluster is then represented by its centroid *(k-means algorithms*: does not work well with symbolic data like medical diagnoses) [4] or by one of its objects located near its center (*k-medoid algorithms*) [5]. In practice, the algorithm is typically running multiple times with different starting states to identify the optimal values of the fixed parameters. The best configuration obtained from all of the runs is used as the output clustering. In the framework of partitioning methods, Hansen and Delattre [6] reduced the partitioning problem of a data set into p classes with minimal diameter, to the *minimal coloring* problem of a *superior threshold graph*. The edges of this graph are the pairs of vertices distanced from more than a given threshold.

In this paper we propose a new coloring method namely *b-coloring*, well-adapted for clustering. It allows to build a fine partition of the data set (*classical* or *symbolic*) in classes where the number of classes is not pre-defined. This approach is interesting

in the sense that it identifies each cluster by at least one *dominant object* which guarantees the disparity between the classes of the partition. The optimal number of classes is then determined based on a new validation algorithm. This algorithm is designed to identify a partition of data that are compact and well separated by evaluating the property of *class dominant object*.

3 Symbolic Clustering Approach

In this section we describe our new graph clustering algorithm. It uses pairwise representation, where the objects (*hospital stays*) are mapped to the nodes of an undirected edge-weighted graph. The edge weights reflect the dissimilarity between the corresponding pair of nodes. The objective here is to generate a partition of a set of n objects $V = \{v_1,...,v_n\}$ in cohesive and well separated classes where their number is not given beforehand. The set V is described by a dissimilarity table $D=\{d_{j,j'} | v_j, v_{j'} \in V\}$.

Notations: we denote by $G=(V,D)$, the complete weighted graph depicting the objects set $V=\{v_1,...,v_n\}$ where each pair (v_i,v_j) is weighted by the dissimilarity d_{ij}. Each object v_i is represented by a mixture of m symbolic as well as classical attributes denoted x_i^k ($k \in \{1...m\}$). Therefore, the object v_i corresponds to the vector $(x_i^1, x_i^2,..., x_i^m)$. *Classical attributes* are defined by either quantitative or qualitative values when *symbolic* attributes can be either a set of values or intervals. The clustering problem is now formulated as a graph *b-coloring* problem.

3.1 A Graph b-Coloring Method

Let $G=(V,E)$ be an undirected, connected and simple graph without loops with a vertex set V and an edge set E. The *b-coloring* of G is the vertex coloring such that:

- For each pair of adjacent vertices $(x,y) \in G$, the color of x and y are different (*proper coloring*),
- In each color class, there exists at least one vertex having neighbors in all other color classes. Such a vertex is called a *dominating vertex*. A color which has a dominating vertex is called a *dominating color*.

We call the *b-chromatic number* $\varphi(G)$ of a graph G the largest number k such that G has a *b-coloring* with k colors.

In contrast to the *minimal coloring* of the vertices of graph G, the *b-coloring* consists to a *maximal coloring* under property and dominance constraints.

The degree of vertex is the number of its neighbors and the maximum degree denoted by $\Delta(G)$ of a graph G is the largest degree over all vertices of G. Irving and Manlove [7] have shown that:

$$\chi(G) \leq \varphi(G) \leq \Delta(G)+1 \ . \tag{1}$$

where *chromatic number* $\chi(G)$ is the minimum number of colors used to have a proper coloring of G,

Irving and Manlove have also shown that finding the *b-chromatic* number $\varphi(G)$ for any graph is a *NP*-hard problem. Several authors investigated this parameter for particular classes of graphs like trees [7] and power graphs [8]

3.2 Construction of the Threshold Graph

Our clustering approach based on a *b-coloring* technique requires to construct a *superior threshold graph*, which is a partial graph of the initial graph $G=(V,D)$. Let $G_{>s}=(V,D_{>s})$ be the superior threshold graph associated with threshold value s chosen among the dissimilarity table D. In other words, $G_{>s}$ is given by $V=\{v_1,...,v_n\}$ as vertex set and $\{(v_i,v_j)|\ D(v_i,v_j)=d_{ij} > s\}$as edge set. Figure 1 gives the superior threshold graph $G_{>0.2}$ associated with the following dissimilarity table:

Table 1. One dissimilarity table

v_i	A	B	C	D	E	F	G
A	0						
B	0.20	0					
C	0.20	0.30	0				
D	0.10	0.20	0.25	0			
E	0.20	0.20	0.10	0.40	0		
F	0.20	0.20	0.20	0.25	0.20	0	
G	0.25	0.20	0.20	0.10	0.20	0.65	0

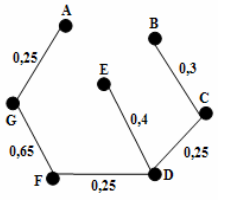

Fig. 1. The superior threshold graph $G_{>0.2}$

3.3 Algorithm

We present here our new clustering algorithm which is based on a graph *b-coloring* technique. The idea consists in applying the *b-coloring technique*-which we adapt to the clustering problem, on the superior threshold graph $G_{>s}$. One partition associated to the selected threshold value s is then returned. Doing so, the dissimilarity between objects within the same class is lower to s.

Notations: Let $G=(V,D_{>s})$ be the threshold graph, such that:

- $V=\{v_1,...,v_n\}$: the vertex set and $D_{>s}$ the edge set.
- Δ: the maximum degree of G.
- $c(v_i)$: the color (integer value) of the vertex v_i.
- $N(v_i)$: the neighborhood of vertex v_i.
- $N_c(v_i)$: the neighborhood colors of vertex v_i.
- $dist(v_i,c)$: the distance between the vertex v_i and the color c defined as the minimum distance from v_i to all vertices with color c:

$$dist(v_i,c)=\min\left\{\begin{matrix} d_{i,j}=D(v_i,v_j) \mid \\ 1\le i\ne j\le n \text{ and } c(v_j)=c \end{matrix}\right\} \quad (2)$$

- L is the color set used in the graph (a set of integer values).
- D_m is a set of colors which have dominating vertices.
- ND_m is a set of colors which have not dominating vertices.

A Graph b-Coloring Algorithm

The *b-coloring* algorithm uses a two-step procedure. Before starting the *b-coloring* algorithm, each vertex must be colored. The following procedure gives an initial configuration using the maximum number of colors available for a *b-coloring* (Eq. 2) (i.e. Δ+1). While vertices are not colored yet, the algorithm starts from the vertex of maximum degree Δ (let v be such a vertex). Then the algorithm puts: $c(v)=1$. The *Procedure 1* then colors remaining vertices. Let:

- T be the set of colored vertices which is sorted on descending order of vertex degrees. Initially T contains only the vertex v and it is updated as long as *Procedure 1* runs.
- *Update*($N_c(v_i)$) be the method which updates the neighborhood colors of the vertex v_i when the color of at least one of its neighbors is changed.
- *Add(o,S)* be the method which adds o (e.g.: color, vertex) into the set S.
- *Remove(o,S)* be the method which remove the object o from the set S.
- *Sort(S)* be the method which sorts the elements of S by decreasing order of vertex degree.

```
Procedure 1 Init_b-coloring()
 begin
    L := {1,2,.., Δ+1};
    repeat
      select v_i from T; M := N_c(v_i) ∪ {c(v_i)}; q := 0;
      for each vertex v_j ∈ N(v_i) such that c(v_j)=∅ do
        q := min{k| k>q, k∉M and k∉ N_c(v_j)};
        if q ≤ Δ+1 then c(v_j):= q;
        else c(v_j) := min{k|k∉ N_c(v_j)};
        Add(v_j,T);
        for each vertex v_k ∈ N(v_j) do Update(N_c(v_k)); enddo.
        sort(T);
      enddo.
      if N_c(v_i)=L\{c(v_i)}then Add(c(v_i),D_m); endif.
      Remove(v_i,T);
    until(T=∅)
 end.
```

Lemma 1. *Procedure 1* executes in $O(n^2\Delta)$.

Proof. *Procedure 1* is applied once on every colored vertex (at most n). First, the algorithm colors every non-colored neighbor (at most Δ). For each neighbor, the change of color is propagated to its own neighbors and the elements of set T (at most n: $O(n)$) are sorted by decreasing order of degree. Therefore *Procedure 1* executes in at most $O(n^2\Delta)$.

Procedure 1 performed on the previous threshold graph $G_{>0.2}$ gives the following colored graph:

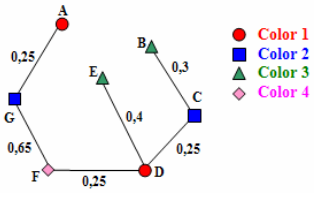

Fig. 2. The initial colored graph $G_{>0.2}$

After applying *Procedure 1*, some colors remain without any *dominating vertex*. The following *Procedure 2* finds a *b-coloring* of graph *G* where all the colors belonging to *L* are *dominating colors* (i.e. D_m=*L*). The idea is the following: each non-dominating color *q* (i.e. $q \in ND_m$) can be changed. In fact, after removing *q* from the graph *G*, for each vertex v_i colored with *q* (i.e. $c(v_i)=q$), a new color is assigned to v_i which is different from those of its neighborhoods. As our objective is to find a partition such that the sum of vertex dissimilarities within each class is minimized, the color whose distance with v_i is minimal will be selected if there is a choice between many colors for v_i. Before starting again with another color $q' \in ND_m$, we verify if colors of ND_m have a dominating vertex (in such a case, these colors are added to the D_m set).

```
Procedure 2 find_b-coloring()
begin
  repeat
    q := max{k| k∈ ND_m}; L := L\{q}; ND_m:= L\D_m;
    for each vertex v_i such that c(v_i)=q do
      K := {k| k∈L and k∉ N_c(v_i)};
      c(v_j):= {c| dist(v_i,c)=min_{k∈K}(dist(v_i,k))};
    enddo.
I   for each vertex v_j such that c(v_j) ∈ ND_m do
      Update(N_c(v_j));
      if N_c(v_j)=L\{c(v_j)}then Add(c(v_j),D_m); endif.
    enddo.
  until(ND_m =∅)
end.
```

Once we modify the color of all vertices v_i having color *q*, we must verify, using the instructions I, if each non-dominating color became a new dominating color.

Lemma 2. We see easily that the coloring generated by *Procedure* 2 is *proper*.

Proof. If the color $c(v_i)$ is changed, it is selected to be different from the neighbor colors of v_i.

Lemma 3. After running the *Procedure 2*, there exists at least one dominating vertex for each obtained color class.

Proof. The *Procedure 2* converges when the set of colors which have not dominating vertices is an empty set.

Lemma 4. *Procedure 2* generates the *b-coloring* of any graph *G* in $O(n\Delta^2)$.

Proof. *Procedure 2* is applied for every color q without dominating vertices (at most Δ). A color q is removed, and every vertex v_i such that $c(v_i)=q$ (at most n) is recolored. Then, for every vertex v_j (at most n), we update its neighborhood colors $N_c(v_j)$ (at most Δ) and thus we verify if its color is a new dominating color. Therefore *Procedure 2* uses at most $((n+n*\Delta) * \Delta)$ instructions. Its complexity is $O(n\Delta^2)$.

Let us consider the previous example. Starting with $L=\{1,2,3,4\}$ *and* $D_m=\{1\}$, the *b-coloring* of the graph $G_{>0.2}$ given by *Procedure 2* is depicted in Figure 3.

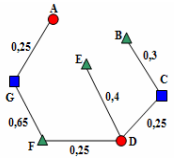

Fig. 3. The b-coloring of the graph $G_{>0.2}$

Thus, the graph $G_{>0.2}$ is colored with three colors, with one *dominating vertex* for each color. Therefore, the partition returned by the algorithm associated to the threshold 0.2 is composed of the three following classes: $C_1=\{\mathbf{D},A\}$, $C_2=\{\mathbf{C},G\}$ and $C_3=\{\mathbf{F},B,E\}$ as shown in Figure 4. The bold characters represent dominating vertices of the obtained classes. This means that these vertices are joined to at least one vertex in each other color class.

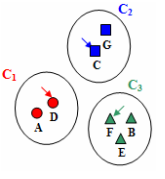

Fig. 4. A partition associated to the threshold 0.2

4 Cluster Validation Algorithm

The clustering algorithm is an iterative algorithm which that executes multiple runs increasing the value of dissimilarity threshold s. Next, the major problem is the validation of clusters resulting from all the runs to select the best configuration (partition) which is considered as the optimal output clustering.

Different cluster validation indices have been proposed in the literature [9], such as *Davies-Bouldin index* and *Dunn's index*. Therefore, the correct value of dissimilarity threshold corresponds to an optimal index value. However, there are several ties or near-ties corresponding to the best results. In order to make the automatic determination of the optimal data partition more robust and eliminate such ties, a cluster validation algorithm was implemented by evaluating the property of dominance in each cluster from all the best considered threshold partitions.

We denote by $I_V(P_i)$ the validity index value associated with the partition P_i.

Algorithm

```
begin
V=(v1,v2,..,vn);
for each Partition Pi selected as one best partition do
 for each cluster Ci=(v1i,..,vηii)∈ Pi where v1i is a dominant do
  for each object vji∈ Ci do
    Q(v1i -> vji):= Q(v1i -> vji)+IV(Pi); // The Rule quality
    Add(v1i -> vji) to the rules set R if it does not exists;
enddo. enddo. enddo.
Sort R by decreasing order of rule quality Q;
for each rule  (x -> y) ∈ R taken consecutively such that V≠∅ do
  Add y to the cluster C which contains x;
  Remove x and y from V if they exist;
  Remove from R all the rules such that x or y are results;
  For each rule Ri such that y is dominant as (y -> t) do
     Q(y -> t):= Q(x -> t)/Q(y -> t);
  enddo.
  Sort R by decreasing order of rule quality Q;
enddo.
end.
```

The cluster validation algorithm was run on the previous data set *(A,B,C,D,E,F,G)*. For each threshold selected in the dissimilarity Table 1, the value of *Dunn's index* of the returned partition is calculated (*cf.* Table 2). The Dunn's index is designed to offer a compromise between the *intercluster separation* and the *intracluster distances*. It identifies sets of clusters that are compact and well separated. Let $s_a(C_k)$ denote the *average distance* within the cluster C_k and $d_a(C_k, C_l)$ the *between-cluster separation*. The larger values of Dunn's indicate the better clustering. For a partition P into p clusters $(C_1,C_2,..,C_p)$, the Dunn's index is defined as follows:

$$Dunn = \min_i \left\{ \min_{j \neq i} \left\{ \frac{d_a(C_i, C_j)}{\max_k s_a(C_k)} \right\} \right\} \quad \text{where} \quad \begin{cases} d_a(C_i, C_j) = \dfrac{1}{\eta_i \times \eta_j} \displaystyle\sum_{u=1}^{\eta_i} \sum_{q=1}^{\eta_j} D(v_u, v_q) \\ s_a(C_k) = \dfrac{1}{\eta_k \times (\eta_k - 1)} \displaystyle\sum_{u=1}^{\eta_k} \sum_{q=1}^{\eta_k} D(v_u, v_q) \end{cases} \tag{3}$$

where $1 \leq i, j, k \leq p$, D indicates the dissimilarity measures, and η_k the number of objects in cluster C_k.

Table 2. Evaluation of each threshold partition

Threshold	Partition	Dunn's index value
0.1	**{A} {B}** {**C**,E}{**G**,D}{**F**}	1.75
0.2	**{C**,G} {**F**,B,E} {**D**,A}	1.00
0.25	**{F**,A,B,D}{**C**,E,G}	1.37
0.3	**{F**,A,B,C,D}{**E**,G}	1.19
0.4	**{F**,A,B,C,D,E} **{G}**	1.25

From Table 2, one can note that the optimal partition which maximizes the Dunn's index value is associated with the threshold 0.1. However, if there are several ties or near-ties corresponding to the maximal values of Dunn's index, we propose to apply

the validation algorithm on the first closest partitions. Assume that the first four best results for the Dunn's index (1.75, 1.37, 1.25 and 1.19) are too closed, so it is not easy to really decide which is the optimal partition, the cluster validation algorithm gives the partition composed of the two clusters: {**F**,A,B,D} and {**C**,E,G}.

5 Experiments

The clustering algorithm and the cluster validation algorithm described in Sections 3 and 4, respectively, were implemented and evaluated. The clustering algorithm was applied to a real data set. This data set consists of 500 instances of *PMSI hospital stays* with 10 features. There are 5 quantitative, 2 qualitative and 3 symbolic attributes. The instances are pre-classified on 21 DRG.

Since the objective was to perform unsupervised classification, hence any class information was eliminated from the data. The results are compared with that of Agglomerative Hierarchical Classification (AHC) [2] and the predefined DRG classification. The measures of the overall average distance within cluster S_a and the overall average between-cluster separations D_a are used for this purpose. For a partition P into p clusters $(C_1,C_2,..,C_p)$ the values of S_a and D_a are given by:

$$S_a = \frac{1}{p}\sum_{i=1}^{p} s_a(C_i) \quad \text{and} \quad D_a = \frac{1}{p\times(p-1)}\sum_{i=1}^{p}\sum_{j=1}^{p} d_a(C_i, C_j) \tag{4}$$

Table 3 shows the first ten best clustering results according to the larger values of Dunn's index.

Table 3. The first ten best Dunn's index values of the PMSI data

Threshold	Number of clusters of partition	Dunn's index value
1.540665	28	0.568483
1.360186	45	0.568007
1.539183	27	0.567937
1.469705	32	0.567813
1.363826	45	0.566491
1.581157	27	0.562751
1.583693	26	0.562511
1.590260	25	0.561748
1.591159	24	0.561563
1.377407	43	0.560905

Table 4. Evaluation of AHC, DRG and our typology of PMSI hospital stays

Classification Method	Number of clusters	*Dunn's*	S_a	D_a	D_a / S_a
Our optimal partition	27	0.5679	1.1767	1.8729	1.5916
Optimal AHC partition	23	0.4026	1.2490	1.8516	1.4824
DRG	21	0.2843	1.2932	1.7989	1.3910

The application of the cluster validation algorithm to these ten partitions provides the partition composed of 27 clusters and associated with the threshold 1.539183 as an optimal partition. According to the *Dunn's index*, S_a and D_a measures, the results presented in the table 4 show that our clustering approach gives better results than the *AHC* method and also the *DRG* classification.

6 Conclusion

In this paper, a new clustering algorithm is proposed. It is based on a graph *b-coloring* technique. We implemented, performed experiments, and compared our approach to the Agglomerative Hierarchical Classification and the DRG classification, and illustrated its efficiency on the *PMSI* data. A real advantage of this method is that it offers a real representation of clusters by a dominant object which is used to evaluate the quality of cluster. So, an optimization algorithm has been developed based on a cluster dominance property. The optimal partition is therefore automatically identified.

The present work can be extended in many directions: (1) we are leading additional experiments on a larger *PMSI* data set by considering other validity indices, (2) improvements have to be made on this stage in order to formally prove the performance of the method; (3) the validation stage of the *hospital stays typology* will be completed with the participation of several specialists from the medical domain.

References

[1] Jain, A.K., M.N. Murty, and P.J. Flynn. Data Clustering: A Review. In *ACM Computing Surveys,* volume. 31, pages 264-323, 1999.

[2] Guha, S., R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD Conference*, Seattle, WA, pages 73-84, 1998.

[3] Guénoche, A., P. Hansen, and B. Jaumard. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. *Journal of Classification*, volume. 8, pages 5-30, 1991.

[4] Hartigan, J. and M. Wong. Algorithm AS136: A k-means clustering algorithm. *Journal of Applied Statistics*, volume 28, pages 100-108, 1979.

[5] NG, R. and J. HAN. Efficient and effective clustering methods for spatial data mining. *In Proceedings of the 20th Conference on VLDB*, Santiago, Chile, pages 144-155, 1994.

[6] Hansen, P. and M. Delattre. Complete-link cluster Analysis by graph coloring. *Journal of the American Statistical Association*, volume 73, pages397-403, 1978.

[7] Irving, W. and D. F. Manlove. The b-chromatic number of a graph. *Discrete Applied Mathematics,* volume 91, pages 127-141, 1999.

[8] Effantin, B. and H. Kheddouci. The b-chromatic number of some power graphs. *Discrete Mathematics and Theoretical Computer Science,* 6(1):45-54, 2003.

[9] Bezdek, J.C. and N.R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):301-315, 1998.