Semi-Supervised Incremental Clustering of Categorical Data

Dan Simovici*
Namita Singla**

*University of Massachusetts Boston
Department of Computer Science, Boston, MA 02125, USA
dsim@cs.umb.edu
**University of Massachusetts Boston
Department of Computer Science, Boston, MA 02125, USA
namita@cs.umb.edu

Résumé. Le clustering semi-supervisé combine l'apprentissage supervisé and non-supervisé pour produire meilleurs clusterings. Dans la phase initiale supervisée de l'algorithme, un échantillon d'apprentissage est produit par selection aléatoire. On suppose que les exemples de l'échantillon d'apprentissage sont étiquetés par un attribut de classe. Puis, un algorithme incrémentiel développé pour les données catégoriques est utilisé pour produire un ensemble de clusters pur (tels que les exemple de chaque cluster ont la même étiquette), qui servent de "seeding clusters" pour la deuxiéme phase non-supervisée de l'algorithme. Dans cette phase, l'algorithme incrémentiel est appliqué aux données non étiquetées. La qualité du clustering est évaluée par l'index de Gini moyen des clusters. Les expériences démontrent que des très bons clusterings peuvent être obtenus avec des petits échantillons d'apprentissage.

1 Introduction

Clustering is a process that aims to partition data into groups that consists of similar objects. Similarity among objects is measured using some metric defined on the set of objects or, whenever possible, using pre-existing classifications of objects. In general, clustering is an unsupervised activity. In other words, clustering takes place without any intervention of an exterior operator that assigns objects to classes. Assuming that the class of an object is determined by the other characteristics of the object, a good clustering algorithm should generate clusters that are as homogeneous as possible.

The core of the clustering algorithm is the incremental construction of a clustering partition of the set of objects such that that the total distance from this partition to the partitions determined by the attributes is minimal. A special challenge of clustering categorical data stems from the fact that no natural ordering exists on the domains of attributes of objects. This leaves only the Hamming distance as a dissimilarity measure, a poor choice for discriminating among multi-valued attributes of objects.

Semi-supervised clustering of categorical data entails two phases : the first phase consists of a supervised process that is applied to a training set obtained randomly sampling the data set. Clusters are formed using an incremental clustering algorithm

that is appropriate for categorical data. Then, these clusters are split into homogeneous clusters that form the seeding clusters for the second phase of the algorithm. In the second unsupervised phase, objects are incrementally added to the existing clusters without using any class label. Finally, clusterings are evaluated using the average Gini index.

Incremental clustering can be traced to (Hartigan 1975) and (Carpenter *et al.*, 1990). This was followed by a seminal paper by Fisher (Fisher 1987) who created COBWEB, an incremental clustering algorithm that involved restructurings of the clusters in addition to the incremental additions of objects. Incremental clustering related to dynamic aspects of databases were discussed in (Can 1993) and (Can *et al.*, 1995). It is also notable that incremental clustering has been used in a variety of applications (Langford *et al.*, 2001), (Lin *et al.*, 2004), (Charikar *et al.*, 2997), (Ester *et al.*,1998).

The other main paradigm applied here, semi-supervised clustering, has recently received lots of attention (Cheung and Yeung 2004), (Bilenko *et al.*, 2004), (Cohn *et al.*,2003), (Zhu *et al.*, 2002), mostly related to numerical data. Our focus here is on categorical data which requires a specific approach.

Incremental clustering insures that the main memory usage is minimal since there is no need to keep in memory the mutual distances between objects ; therefore, the algorithms are very scalable with respect to the size of the set of objects and the number of attributes. Semi-supervised clustering, acting as a wrapper for the underlying incremental clustering improves the quality of the clustering.

2 Partitions and Clusterings

Let S be a set. A *partition on* S is a non-empty collection of non-empty subsets of S indexed by a set I, $\pi = \{B_i \mid i \in I\}$ such that $\bigcup_{i\in I} B_i = S$ and $i \neq j$ implies $B_i \cap B_j = \emptyset$. The sets B_i are the *blocks of the partition* π. The set of partitions on S is denoted by $\mathsf{PART}(S)$.

For $\pi, \sigma \in \mathsf{PART}(S)$ we write $\pi \leq \sigma$ if every block B of π is included in a block of σ, or equivalently, if every block of σ is an exact union of blocks of π. This partial order generates a lattice structure on $\mathsf{PART}(S)$; this means that for every two partitions $\pi, \pi' \in \mathsf{PART}(S)$ there is a least partition π_1 such that $\pi \leq \pi_1$ and $\pi' \leq \pi_1$ and there is a largest partition π_2 such that $\pi_2 \leq \pi$ and $\pi_2 \leq \pi'$. The first partition is denoted by $\pi \vee \pi'$, while the second is denoted by $\pi \wedge \pi'$.

To introduce a metric on the set of partitions of a finite set we define the mapping $v : \mathsf{PART}(S) \longrightarrow \mathbb{R}$ by $v(\pi) = \sum_{i=1}^{n} |B_i|^2$, where $\pi = \{B_1, \ldots, B_n\}$.

The mapping v is a lower valuation on $\mathsf{PART}(S)$, that is,

$$v(\pi \vee \sigma) + v(\pi \wedge \sigma) \geq v(\pi) + v(\sigma) \tag{1}$$

for $\pi, \sigma \in \mathsf{PART}(S)$ (see Appendix 5 for a proof).

For every lower valuation v the mapping $d : (\mathsf{PART}(S))^2 \longrightarrow \mathbb{R}$ defined by $d(\pi, \sigma) = v(\pi) + v(\sigma) - 2v(\pi \wedge \sigma)$ is a metric on $\mathsf{PART}(S)$ (see (J.P. Barthélemy et B. Leclerc, 1995), (J.P. Barthélemy, 1978), (Monjardet, 1981). A special property of this metric

allows the formulation of an incremental clustering algorithm which is used as a part of the semi-supervised clustering.

An *object system* is a pair $\mathcal{S} = (S, H)$, where S is set called the set of objects of $\mathcal{S}$, $H = \{A_1, \ldots, A_m\}$ is a set of mappings defined on S. For each mapping A_i (referred to as an attribute of $\mathcal{S}$) there exists a nonempty set E_i called the domain of A_i such that $A_i : S \longrightarrow E_i$ for $1 \leq i \leq m$. The value of an attribute A_i on an object t is denoted by $t[A_i]$. This is consistent with the terminology used in relational databases, where a table can be regarded as an object system ; however, the notion of object system is more general because objects have an identity as members of the set S, instead of being regarded as just m-tuples of values. In this spirit, we shall refer to $t[A_i]$ as *projection of* t *on* A_i.

An attribute A of an object system $\mathcal{S} = (S, H)$ generates a partition π^A of the set of objects S, where two objects belong to the same block of π^A if they have the same projection on A. We denote by B_a^A the block of π^A that consists of all tuples of S whose A-component is a. Note that for relational databases, π^A is the partition of the set of rows of a table that is obtained by using the **group by** A option of **select** in standard SQL.

A *clustering* of an object system $\mathcal{S} = (S, H)$ is defined as a partition κ of S. The blocks of the partition κ are the *clusters* of κ.

3 A Semi-Supervised Incremental Clustering Algorithm

A semi-supervised clustering of an object system $\mathcal{S} = (S, H)$ begins with the assumption that an oracle provides the value of a special attribute K of objects referred to as the *class of the object* for a subset T of the object set S.

In the first phase of the algorithm an incremental clustering algorithm $\mathcal{A}$ is applied to the object set T which yields an initial clustering σ of this set. In general, these clusters are not pure relative to the class K, that is, we may find in the same class objects that have distinct values of the attribute K. Then, each of the clusters of T is split into pure clusters. The partition κ_0 of T obtained in the manner contains the *seeding clusters* for the clustering of the full set of objects.

The second, unsupervised phase of the algorithm starts with the partition κ_0 of the set T. Using the incremental clustering algorithm, objects from the set $S - T$ are added to existing clusters or form new clusters. The class attribute (if existent) plays no role in this phase. The final clustering extends the partition κ_0 of T to a clustering partition κ of the entire set of objects.

We begin by discussing our incremental clustering algorithm. For an object system $\mathcal{S} = (S, H)$ we seek a clustering $\kappa = \{C_1, \ldots, C_n\} \in \mathsf{PART}(S)$ such that the total distance from κ to the partitions of the attributes :

$$D(\kappa) = \sum_{i=1}^{n} d(\kappa, \pi^{A_i})$$

has a local minimum. The definition of d allows us to write :

$$D(\kappa) = \sum_{i=1}^{n} |C_i|^2 + \sum_{j=1}^{m_A} |B^A_{a_j}|^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{m_A} |C_i \cap B^A_{a_j}|^2,$$

Let t be a new object, $t \notin S$, and let let $Z = S \cup \{t\}$. To form a clustering of the set Z the object t may added to an existing cluster C_k, or a new cluster C_{n+1}, may be created that consists only of t.

If t is added to an existing cluster C_k, the new clustering is

$$\kappa_{(k)} = \{C_1, \ldots, C_{k-1}, C_k \cup \{t\}, C_{k+1}, \ldots, C_n\},$$

and the new attribute partition is

$$\pi^{A\prime} = \{B^A_{a_1}, \ldots, B^A_{t[A]} \cup \{t\}, \ldots, B^A_{a_{m_A}}\}$$

Now, we have :

$$\begin{aligned} & d(\kappa_{(k)}, \pi^{A\prime}) - d(\kappa, \pi^A) \\ &= (|C_k| + 1)^2 - |C_k|^2 + (|B^A_{t[A]}| + 1)^2 - |B^A_{t[A]}|^2 - 2(2|C_k \cap B^A_{t[A]}| + 1) \\ &= 2|C_k| + 1 + 2|B^A_{t[A]}| + 1 - 4|C_k \cap B^A_{t[A]}| - 2 \\ &= 2|C_k \oplus B^A_{t[A]}|, \end{aligned}$$

where $\oplus$ is the symmetric difference of sets given by $X \oplus Y = (X \cup Y) - (X \cap Y)$ for every sets X, Y.

When t is forming a new cluster we have the partitions

$$\begin{aligned} \kappa' &= \{C_1, \ldots, \ldots, C_n, \{t\}\} \\ \pi^{A\prime} &= \{B^A_{a_1}, \ldots, B^A_{t[A]} \cup \{t\}, \ldots, B^A_{a_{m_A}}\} \end{aligned}$$

which yield

$$d(\kappa', \pi^{A\prime}) - d(\kappa, \pi^A) = 2|B^A_{t[A]}|.$$

Consequently,

$$D(\kappa') - D(\kappa) = \begin{cases} \sum_A 2 \cdot |C_k \oplus B^A_{t[A]}| & \text{in Case 1} \\ \sum_A 2 \cdot |B^A_{t[A]}| & \text{in Case 2.} \end{cases}$$

Thus, the choice between adding an object to an existing cluster and creating a new cluster is based on comparing the numbers

$$\min_k \sum_A |C_k \oplus B^A_{t[A]}| \text{ and } \sum_A |B^A_{t[A]}|.$$

If the first number is smaller, we add t to a cluster C_k for which $\sum_A |C_k \oplus B^A_{t[A]}|$ is minimal ; otherwise, we create a new one-object cluster.

```
Input : data set S,
        fraction of supervised set p,
        ''not-yet'' threshold α
Output : clustering C_1, ..., C_n
Method :obtain a random sample of objects T from
        the set of objects S such that |T|/|S| = p ;
        compute the seed clustering of the set T
              κ_0 = {D_1, ..., D_ℓ} = A(T, α)
        compute the final clustering
              κ = C(S, T, κ_0, α)
```

FIG. 1 – Pseudocode of the semi-supervised clustering algorithm

For incremental clustering algorithms certain object orderings may result in rather poor clusterings. To diminish the ordering effect problem we expand the initial algorithm by adopting the "not-yet" technique introduced in (Roure and Talavera, 1998). A new cluster is created only when the effect of adding the object t on the total distance is significant enough. This is the case when $\frac{\sum_A |B^A_{t[A]}|}{\min_k \sum_A |C_k \oplus B^A_{t[A]}|} < \alpha$, where $\alpha \leq 1$ is a parameter provided by the user. Otherwise, the object t in placed in a `NOT-YET` buffer. All experiments described in Section 4 used $\alpha = 0.95$.

When $\frac{\sum_A |B^A_{t[A]}|}{\min_k \sum_A |C_k \oplus B^A_{t[A]}|} > 1$, the object t is placed in an existing cluster C_k that minimizes $\sum_A |C_k \oplus B^A_{t[A]}|$. This approach limits the number of new singleton clusters that would be otherwise created. After all objects of the set S have been examined, the objects contained by the `NOT-YET` buffer are processed with $\alpha = 1$. This prevents new insertions in the buffer and results in either placing these objects in existing clusters or in creating new clusters.

Thus, the construction of the final clustering κ of S starts with an initial clustering partition κ_0 of a subset T and with a parameter α. We denote the final clustering κ by $\mathcal{C}(S, T, \kappa_0, \alpha)$.

The partition created on the initial set of objects T is denoted by $\kappa_0 = \mathcal{A}(T, \alpha)$ and it uses the same algorithm as above.

The algorithm is given next :

4 Experimental Results

We applied the semi-supervised clustering to several categorical databases obtained from the UCI data set (C. L. Blake et C. J. Merz, 1998). Each experiment was applied using a series of increasing percentages for the semi-supervised data set, averaged over five random samples.

The quality of the clustering for categorical data requires a specialized treatmet

since distances between objects cannot be defined naturally. We evaluated clusterings using the averaged Gini index of the clusters (Demiriz *et al.*, 1999).

Let K the class attribute and let $\{B^K_{k_1}, \ldots, B^K_{k_p}\}$ be the partition of the object set S. The class-impurity of a set of objects U is defined as the Gini index of the "trace partition" $\{U \cap B^K_{k_j} \mid 1 \leq j \leq p\}$:

$$\mathsf{gini}_K(U) = 1 - \sum_{j=1}^{p} \left(\frac{|U \cap B^K_{k_j}|}{|U|} \right)^2 .$$

Note that if a cluster U is pure, that is, it contains objects that belong to only one class, then $\mathsf{gini}_K(U) = 0$.

For a clustering $\kappa = \{U_1, \ldots, U_\ell\}$ of the set of objects S the average Gini index is given by

$$\mathsf{imp}_K(\kappa) = \sum_{i=1}^{\ell} \frac{|U_i|}{|S|} \mathsf{gini}_K(U_i).$$

Clearly, low values of $\mathsf{imp}_K(\kappa)$ indicate good clusterings.

The algorithm was applied to the `MUSHROOM` data set. This data set contains 8124 mushroom records and is typically used as test set for classification algorithms, where the task is to construct a classifier that is able to predict the poisonous/edible character of the mushrooms based on the values of the attributes of the mushrooms. The clusters show quite a remarkable degree of purity. For example, for a semi-supervised portion of 10% we obtained the following clusters :

Cluster number	Instances	edib./pois.	Percent of dominant group
1	4225	3575/650	84.615
2	165	0/165	100
3	3055	0/3055	100
4	394	393/1	99.746
5	2	0/2	100
6	55	48/7	87.273
7	36	0/36	100
8	192	192/0	100

It is quite remarkable that five of the eight clusters obtained in this manner are pure and the remaining clusters have a high degree of purity. For other sizes of the supervised sample we obtained the following results :

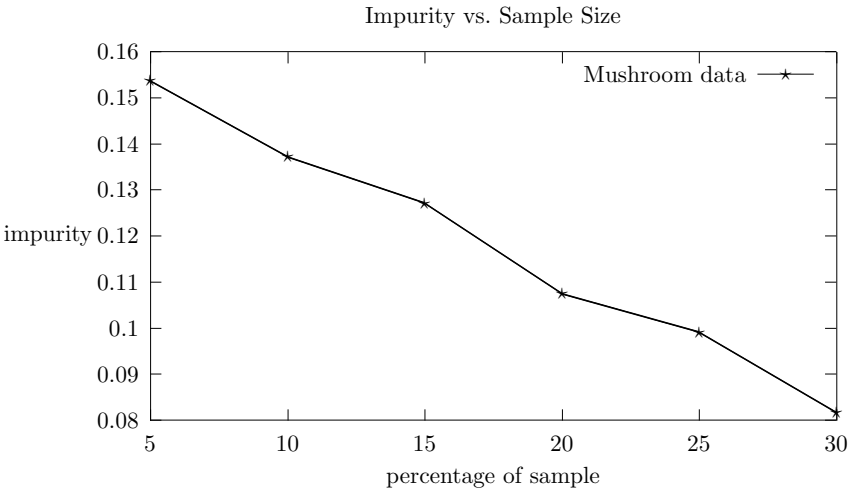

FIG. 2 – Impurity Decrease with Sample Size for MUSHROOMS

Mushroom database

Percent supervised	Number of clusters	Impurity	Time (ms)
5%	8	0.15362536	2443
10%	8	0.1371508	2454
15%	8	0.12705285	2444
20%	8	0.10735634	2374
25%	9	0.09911141	3545
30%	9	0.0816238	3415

The dependency of the impurity measure on the fraction of the supervised sample is shown in Figure 2.

A similar, albeit slower improvement of the quality of clustering can be observed for ZOO, another categorical data set from UCI (C. L. Blake et C. J. Merz, 1998) :

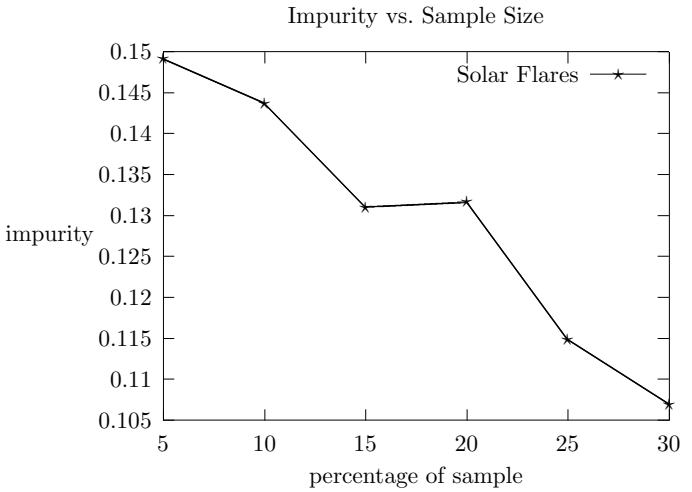

FIG. 3 – Impurity Decrease with Sample Size for SOLAR_FLARES

Zoo database

Percent supervised	Number of clusters	Impurity	Time (ms)
5%	3	0.39802246	110
10%	4	0.37841779	90
15%	4	0.37841779	110
20%	6	0.28431165	130
25%	7	0.33374854	141
30%	7	0.33981398	114

The variation of the average impurities for five experiments with each sample size for the `SOLAR_FLARES` database is shown in Figure 3.

5 Conclusion and Future Work

Semi-supervised incremental clustering is an efficient clustering algorithm for categorical data that generates almost homogeneous clusters relative to classifications based on attribute values.

A natural idea for development of the semi-supervised approach would be to use a boosted model (Freund, 1995) of the semi-supervised incremental clustering where several small training samples would be used to generate clusterings ; an object would then be classified according to its positions relative to the ensemble of clusters.

We will explore the semi-supervised incremental clustering in the context of clustering streams of objects, which is an important type of data in internet mining and network security. The ordering of objects is irrelevant in this realm since objects must be dealt with as they arrive.

References

J.P. Barthélemy et B. Leclerc, The median procedure for partitions, in *Partitioning Data Sets*, pages 3–34, Providence, 1995, American Mathematical Society.

J.P. Barthélemy, Remarques sur les propriétés metriques des ensembles ordonnés, *Math. Sci. hum.*, 61 :39–60, 1978.

M. Bilenko, S. Basu, et R. J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in *International Conference on Machine Learning*, Banff, Canada, 2004.

G. Birkhoff, Lattice Theory, American Mathematical Society, Providence, 1973.

C. L. Blake et C. J. Merz, *UCI Repository of machine learning databases*, University of California, Irvine, Dept. of Information and Computer Sciences, http ://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

F. Can, E. A. Fox, C. D. Snavely, et R. K. France, Incremental clustering for very large document databases : Initial MARIAN experience, *Inf. Sci.*, 84 :101–114, 1995.

F. Can, Incremental clustering for dynamic information processing, *ACM Transaction for Information Systems*, 11 :143–164, 1993.

G. Carpenter et S. Grossberg, Art3 : Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures, *Neural Networks*, 3 :129–152, 1990.

M. Charikar, C. Chekuri, T. Feder, et R. Motwani, Incremental clustering and dynamic information retrieval, in *STOC*, pages 626–635, 1997.

H. Cheung et D. Y. Yeung, Locally linear metric adaptation for semi-supervised clustering, in *International Conference on Machine Learning*, Banff, Canada, 2004.

D. Cohn, R. Caruana, et A. McCallum, Semi-supervised clustering with user feedback, Technical report, 2003.

A. Demiriz, K. P. Bennett, et M. E. Embrechts, Semi-supervised clustering using genetic algorithms, Technical Report Math 9901, Rensselaer Polytechnical Institute, Troy, New York, 1999.

M. Ester, H. P. Kriegel, J. Sander, M. Wimmer, et X. Xu, Incremental clustering for mining in a data warehousing environment, in *VLDB*, pages 323–333, 1998.

D. Fisher, Knowledge acquisition via incremental conceptual clustering, *Machine Learning*, 2 :139–172, 1987.

Y. Freund, Boosting a weak learning algorithm by majority, *Information and Computation*, 121 :256–285, 1995.

J. A. Hartigan, *Clustering Algorithms*, John Wiley, New York, 1975.

T. Langford, C. G. Giraud-Carrier, et J. Magee, Detection of infectious outbreaks in hospitals through incremental clustering, in *Proceedings of the 8th Conference on AI in Medicine (AIME)*, pages 30–39. Springer, 2001.

J. Lin, M. Vlachos, E. J. Keogh, et D. Gunopulos, Iterative incremental clustering of time series, in *EDBT*, pages 106–122, 2004.

B. Monjardet, Metrics on partially ordered sets – a survey, *Discrete Mathematics*, 35 :173–184, 1981.

J. Roure et Luis Talavera, Robust incremental clustering with bad instance orderings : A new strategy, in *IBERAMIA*, pages 136–147, 1998.

Z. Zhu, Y. Pilpel, et G.M. Church, Computational identification of transcription factor binding sites via a transcription-factor-centric clustering (tfcc) algorithm, *Journal of Molecular Biology*, 318 :71–81, 2002.

A proof of inequality (1)

Let π, σ be two partitions of the finite set S, such that $\pi = \{B_1, \ldots, B_m\}$ and $\sigma = \{C_1, \ldots, C_n\}$. It is known (see (Birkhoff 1973), for example) that $\pi \wedge \sigma$ consists of all sets of the form $B_i \cap C_j$ such that $B_i \cap C_j \neq \emptyset$. On another hand, $\pi \vee \sigma$ has a more complicated description; namely, $x, y \in S$ belong to the same block D of $\pi \vee \sigma$ if there exists a sequence of elements of S, $z_0, \ldots, z_k$ such that $x = z_0$, $z_k = y$ and for each pair (z_p, z_{p+1}) there is a block B_i of π or a block C_j of σ such that both z_p and z_{p+1} belong to B_i or to C_j for $1 \leq p \leq k-1$.

Consider the bipartite graph $G_{\pi,\sigma}$ whose set of vertices consists of the blocks of π and the blocks of σ. An edge (B_i, C_j) exists only if $B_i \cap C_j \neq \emptyset$. If $\mathcal{K}$ is a connected component of this graph it is easy to see that $\bigcup\{B_i \in \pi \mid B_i \in \mathcal{K}\} = \bigcup\{C_j \in \sigma \mid C_j \in \mathcal{K}\}$. Further, each block D of $\pi \vee \sigma$ equals the union of the blocks of π (or the blocks of σ) that belong to a connected component $\mathcal{K}$ of $G_{\pi,\sigma}$.

Example 5.1 Let $S = \{a_i \mid 1 \leq i \leq 12\}$ and let $\pi = \{B_i \mid 1 \leq i \leq 5\}$ and $\sigma = \{C_j \mid 1 \leq j \leq 4\}$, where

$$\begin{array}{ll} B_1 = \{a_1, a_2\}, & C_1 = \{a_2, a_4\}, \\ B_2 = \{a_3, a_4, a_5\}, & C_2 = \{a_1, a_3, a_5, a_6, a_7\}, \\ B_3 = \{a_6, a_7\}, & C_3 = \{a_8, a_{11}\}, \\ B_4 = \{a_8, a_9, a_{10}\}, & C_4 = \{a_9, a_{10}, a_{12}\}, \\ B_5 = \{a_{11}, a_{12}\}. & \end{array}$$

The graph $G_{\pi,\sigma}$ shown in Figure 4 has two connected components that correspond to the blocks

$$\begin{array}{rcl} D_1 & = & \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\} \\ & = & B_1 \cup B_2 \cup B_3 \\ & = & C_1 \cup C_2, \\ D_2 & = & \{a_8, a_9, a_{10}, a_{11}, a_{12}\} \\ & = & B_4 \cup B_5 \\ & = & C_3 \cup C_4. \end{array}$$

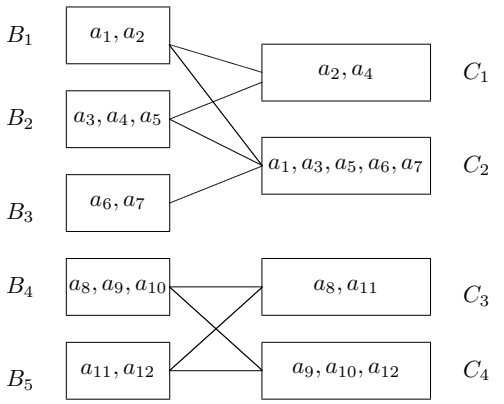

FIG. 4 – The graph $G_{\pi,\sigma}$

of the partition $\pi \vee \sigma$.

The partition $\pi \wedge \sigma$ consists of 9 blocks that correspond to the edges of the graph :

$$\begin{array}{lll} B_1 \cap C_1 = \{a_2\}, & B_1 \cap C_2 = \{a_1\}, B_2 \cap C_1 = \{a_4\}, & \\ B_2 \cap C_2 = \{a_3, a_5\}, & B_3 \cap C_2 = \{a_6, a_7\}, & B_4 \cap C_3 = \{a_8\}, \\ B_4 \cap C_4 = \{a_9, a_{10}\}, & B_5 \cap C_3 = \{a_{11}\}, & B_5 \cap C_4 = \{a_{12}\}. \end{array}$$

□

Let $D_1, \ldots, D_r$ be the blocks of the partition $\pi \vee \sigma$. For a block D_k define the sets $I_k \subseteq \{1, \ldots, m\}$ and $J_k \subseteq \{1, \ldots, n\}$ where $I_k = \{i \mid B_i \cap D_k \neq \emptyset\}$ and $J_k = \{j \mid$

$B_i \cap D_k \neq \emptyset\}$. Note that

$$\begin{aligned} v(\pi \vee \sigma) &= \sum_{k=1}^{r} |D_k|^2, \\ v(\pi \wedge \sigma) &= \sum_{k=1}^{r} \sum_{i \in I_k} \sum_{j \in J_k} |B_i \cap C_j|^2, \\ v(\pi) &= \sum_{k=1}^{r} \sum_{i \in I_k} |B_i|^2 \\ &= \sum_{k=1}^{r} \sum_{i \in I_k} \left(\sum_{j \in J_k} |B_i \cap C_j| \right)^2, \\ v(\sigma) &= \sum_{k=1}^{r} \sum_{j \in J_k} |C_j|^2 \\ &= \sum_{k=1}^{r} \sum_{j \in J_k} \left(\sum_{i \in I_k} |B_i \cap C_j| \right)^2. \end{aligned}$$

It is immediate to verify the inequality :

$$\begin{aligned} &\left(\sum_{i \in I_k} \sum_{j \in J_k} |B_i \cap C_j|\right)^2 + \sum_{i \in I_k} \sum_{j \in J_k} |B_i \cap C_j|^2 \\ &\geq \sum_{i \in I_k} \left(\sum_{j \in J_k} |B_i \cap C_j|\right)^2 + \sum_{j \in J_k} \left(\sum_{i \in I_k} |B_i \cap C_j|\right)^2 \end{aligned}$$

This is equivalent to :

$$\begin{aligned} &|D_k|^2 + \sum_{i \in I_k} \sum_{j \in J_k} |B_i \cap C_j|^2 \\ &\geq \sum_{i \in I_k} \left(\sum_{j \in J_k} |B_i \cap C_j|\right)^2 + \sum_{j \in J_k} \left(\sum_{i \in I_k} |B_i \cap C_j|\right)^2 \end{aligned}$$

Adding up the similar inequalities for $1 \leq k \leq r$ we have the desired inequality : $v(\pi \vee \sigma) + v(\pi \wedge \sigma) \geq v(\pi) + v(\sigma)$.,

Summary

Semi-supervised clustering combines supervised and unsupervised learning to produce better clusterings. In the initial supervised phase of the proposed algorithm a training set is generated by sampling. It is assumed that the examples of the training set are labelled by a class attribute. Then, an incremental algorithm developed for categorical data is used to produce a set of pure clusters (such that the instances of each cluster have the same label) that serve as "seeding clusters" for the second, unsupervised phase. In this phase the incremental algorithm is applied to unlabelled data. The quality of the clustering is evaluated by the average Gini index of the clusters. Experiments demonstrate that very good clusterings can be obtained with relatively small training sets.