HAL
archives-ouvertes.fr

Equitable Conceptual Clustering using OWA operator

Noureddine Aribi, Abdelkader Ouali, Yahia Lebbah, Samir Loudni

▶ To cite this version:

Noureddine Aribi, Abdelkader Ouali, Yahia Lebbah, Samir Loudni. Equitable Conceptual Clustering using OWA operator. 22nd Pacific-Asia Conference, PAKDD 2018, Jun 2018, Melbourne, Australia. pp.465-477. hal-01709850

HAL Id: hal-01709850
https://hal.archives-ouvertes.fr/hal-01709850

Submitted on 1 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Equitable Conceptual Clustering using OWA operator

Noureddine Aribi[1], Abdelkader Ouali[2], Yahia Lebbah[1], and Samir Loudni[2].

[1] Lab. LITIO, University of Oran 1, 31000 Oran, Algeria.
[2] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France.

Abstract. We propose an equitable conceptual clustering approach based on multi-agent optimization. In the context of conceptual clustering, each cluster is represented by an agent having its own satisfaction and the problem consists in finding the best cumulative satisfaction while emphasizing a fair compromise between all individual agents. The fairness goal is achieved using an equitable formulation of the Ordered Weighted Averages (OWA) operator. Experiments performed on UCI datasets and on instances coming from real application ERP show that our approach efficiently finds clusterings of consistently high quality.

1 Introduction

Structuring data in knowledge discovery appears as a fundamental task which permits to better understand the data and to define groups with regards to an a priori similarity measure. This is usually referred to clustering in unsupervised learning task. In practice, users often would like to perform further actions at the cluster level, such as interpreting the cluster semantically. Methods such as conceptual clustering address this by attempting to find descriptions of the clusters by means of formal concepts. Numerous approaches have been devised for conceptual clustering. Traditional approaches [13,8] combine the formation of the clusters and of the descriptions. Other techniques [20,19] have instead chosen to decouple finding the descriptions – either before or after the clustering process – and the clustering step. More recently, Constraint Programming (CP) [4] and Integer Linear Programming (ILP) [17] approaches have been proposed to address the problem of finding optimal conceptual clusterings in a declarative framework. They combine two exact techniques: in a first step, a dedicated mining tool (i.e., `LCM` [22]) is used to compute the set of all formal concepts and, in a second step, ILP or CP is used to select the best k clusters (i.e. concepts) that optimizes some given criterion. Most of the optimization measures used in these approaches lead to an unbalanced clustering where one cluster is more dominant than others. Ensuring that the clusters obtained be (roughly) balanced, i.e. of approximately the same number of data points helps in making the resulting clusterings more useful and actionable [2,24].

This paper deals with the concept of *equitably efficient solutions* to conceptual clustering problem in multi-agent decision making, where each agent represents a concept and has its own utility corresponding to a specific measure (e.g. the frequency). Here, *equity* refers to the idea of favoring solutions that fairly share happiness or dissatisfaction among agents [9]. The equity requirement has been fully studied by the multicriteria optimization community [10], and formalized through the three properties:
Symmetry meaning that all agents have the same importance. For instance, both utility vectors $(5, 3, 0)$ and $(0, 3, 5)$ are considered equivalent.

Pareto-monotony which expresses that solution $(x_1, x_2, ..., x_n)$ is better than solution $(y_1, y_2, ..., y_n)$ if and only if $x_i \geq y_i$ for all i, with at least one strict inequality.
Transfer Principle formalizes an important notion of equitable utility distribution [21]. The intuition is that any transfer between some two inequitable utilities x_i and x_j, which preserves the average of utilities, would improve the overall utility. For instance, a more equitable vector $y = (9, 10, 9, 10)$ can be obtained from $x = (11, 10, 7, 10)$ by transferring two units between the first and the third agents.

The common way to deal with the concept of equitably efficient solutions is to define aggregation functions that fulfills the above properties. This defines a family of the equitable aggregations which are *Schur-convex* [11]. In the literature there are several functions to aggregate individual agents' utilities by mean of *collective utility function* (CUF). The most used aggregations are `maxMin`, `minDev` and `maxSum`. The transfer principle is not ensured in the `maxMin` and `minDev`, on all of the utilities, thereby leading to the *drowning effect* [7]. The `maxSum` function is fully compensatory and thus does not capture the idea of equity.

The next section introduces the concepts used in this paper. Section 3 describes how equitable conceptual clustering task can be expressed as ILP problems. We discuss related work in Section 4 before demonstrating our technique's performance in Section 5. Section 6 concludes and points towards future research directions.

2 Background

2.1 Formal Concepts and Conceptual Clustering

Formal Concepts. Let $\mathcal{D}$ be a set of m transactions (numbered from 1 to m), $\mathcal{I}$ a set of n items (numbered from 1 to n), and $R \subseteq T \times \mathcal{I}$ a binary relation that lies transactions to items: $(t, i) \in R$ if the transaction t contains the item $i : i \in t$. An itemset (or *pattern*) is a non-null subset of $\mathcal{I}$. For instance, Table 1a gives a transactional dataset $\mathcal{D}$ with m=11 transactions $t_1, \ldots, t_{11}$ described by n=8 items.

The *extent* of a set $I \subseteq \mathcal{I}$ of items is the set of transactions containing all items in I, i.e., $ext(I) = \{t \in \mathcal{D} | \ \forall i \in I, (t, i) \in R\}$. The *intent* of a subset $T \subseteq \mathcal{D}$ is the set of items contained by all transactions in T, i.e., $int(T) = \{i \in \mathcal{I} | \ \forall t \in T, (t, i) \in R\}$. These two operators induce a Galois connection between $2^{\mathcal{D}}$ and $2^{\mathcal{I}}$, i.e. $T \subseteq ext(I) \Leftrightarrow I \subseteq int(T)$. A pair such that $(I = int(T), T = ext(I))$ is called **formal concept**. This definition defines a **closure property** on dataset $\mathcal{D}$, $\texttt{closed}(I) \Leftrightarrow I = int(ext(I))$. An itemset I for which $\texttt{closed}(I) = \text{true}$ is called *closed pattern*. Using $ext(I)$, we can define the *frequency* of a concept: $\texttt{freq}(I) = |ext(I)|$, and its *diversity*: $\texttt{divers}(I) = \sum_{t \in ext(I)} |\{i \in \mathcal{I} \mid \ (i \notin I) \wedge (i \in t)\}|$. Additionally, we can refer to its *size*: $\texttt{size}(I) = |\{i \mid i \in I\}|$. We note $\mathcal{C}$ the set of all formal concepts.

Conceptual Clustering. Clustering is the task of assigning the transactions in the data to relatively homogeneous groups. Conceptual clustering aims to also provide a distinct description for each cluster - the concept characterizing the transactions contained in it. This problem can be formalized as: "find a set of k clusters, each described by a closed pattern $P_1, P_2, ..., P_k$, covering all transactions without any overlap between clusters".

An evaluation function f that optimizes a given criterion can be used to express the goodness of the clustering. Different optimization criteria may be considered: max-

Trans.	Items
t_1	A B D
t_2	A E F
t_3	A E G
t_4	A E G
t_5	B E G
t_6	B E G
t_7	C E G
t_8	C E G
t_9	C E H
t_{10}	C E H
t_{11}	C F G H

(a) Transactional dataset $\mathcal{T}$.

Sol.	P_1	P_2	P_3
s_1	{A, B, D}	{C, F, G, H}	{E}
s_2	{B}	{C}	{A, E}
s_3	{A}	{C}	{B, E, G}

(b) Three conceptual clusterings for k=3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t_1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
t_2	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0
t_3	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1
t_4	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1
t_5	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1
t_6	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1
t_7	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	0	1
t_8	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1	0	1
t_9	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0
t_{10}	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0
t_{11}	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1	1

(c) $(a_{t,c})$ matrix associated with $\mathcal{D}$.

Table 1: Running example.

imizing the *sum of frequencies* of the selected concepts; minimizing the *sum of diversities* of the selected concepts. For instance, for dataset $\mathcal{D}$ and k=3, minimizing $f(P_1, ..., P_k) = \sum_{1 \leq i \leq k} \texttt{divers}(P_i)$ provides one clustering s_1, with optimal value 18 (see Table 1b). Solution $s_1 = (1, 1, 9)$ has one large cluster (of size 9) covering most of the transactions, and two clusters that cover only one transaction. Such a clustering may be less interesting than those in which the clusters are all of comparable size. A common way to get more balanced clusterings is to consider dedicated optimization settings. This can be formalized in two possible ways:

- *maximizing the minimal frequency* (`maxMin`). We search for solutions in which the minimal frequency of the selected concepts is as large as possible.

- *minimizing the deviation in cluster frequency* (`minDev`). We enforce a small difference between cluster frequencies: `Min` $\texttt{max}(\texttt{freq}(P_1), \ldots)$ - $\texttt{min}(\texttt{freq}(P_1), \ldots)$.

However, as stated in the introduction, these two settings suffer from the so called *drowning effect* [7]. In fact, concerning `maxMin` (resp. `minDev`), the transfer principle is ensured only on the `min` (resp. `min` and `max`) utility, and thus intermediate utilities are not necessarily equitable. To address equity requirement, we consider, in the next section, a sophisticated operator that focuses on the whole utilities.

2.2 Equitable multiagent optimization

Let $N = \{1, .., n\}$ be a set of n agents. A solution of a multiagent optimization problem is characterized by a utility vector $x = (x_1, ..., x_n) \in I\!R^n_+$, where x_i represent the utility (or a degree of satisfaction) of the i^{th} agent. Utility vectors are commonly compared using the Pareto dominance relation (P-dominance). The weak-P-dominance $\succsim_P$ between two utility vectors x, x' is defined as: $x \succsim_P x' \Leftrightarrow [\forall i \in N, x_i \geq x'_i]$, whereas the strict P-dominance $\succ_P$ between x and x' is given by: $x \succ_P x' \Leftrightarrow [x \succsim_P x' \ \wedge not(x' \succsim_P x)]$. A solution x^* is Pareto-optimal (a.k.a *efficient*) if and only if there is no solution x that dominates x^*. The P-dominance can be formulated as: $max \{(x_1, ..., x_n) : x \in Q\}$, where Q is the set of feasible solutions. The P-dominance may lead to a large set of incomparable solutions. Also, the P-dominance is insensitive to *outliers*. To refine the P-dominance, we should specialize a dominance relation so as to favor *equitable* utility vectors. The main intuition behind the equity criterion refers to the idea of selecting solutions that fairly share satisfaction between agents [21]. Formally, an equitable dominance relation $\succsim_{||}$ should fulfill three main properties [11,9]:

Symmetry. Consider a utility vector $x \in I\!R^n_+$. For any permutation σ on N, we have $(x_{\sigma(1)}, ..., x_{\sigma(n)}) \sim (x_1, ..., x_n)$.
P-Monotony. For all $x, y \in I\!R^n_+$, $x \succsim_P y \Rightarrow x \succsim_{\|} y$ and $x \succ_P y \Rightarrow x \succ_{\|} y$.
Transfer principle. (a.k.a *Pigou-Dalton* transfers in Social Choice Theory) Let $x \in I\!R^n_+$ and $x_i > x_j$ for some $i, j \in N$. Let e^z be a vector such that $\forall i \neq z, e^z_i = 0$ and $e^z_z = 1$. For all ϵ where $0 < \epsilon \leq \frac{x_i - x_j}{2}$, we get $x - \epsilon e^i + \epsilon e^j \succsim_{\|} x$. Any slight improvement of x_j at the expense (reduction) of x_i, which preserves the *average of utilities*, would produce a better distribution of the utilities among agents and consequently improve the overall utility of the solution. For example, if we consider two utility vectors $x = (\mathbf{11}, 10, \mathbf{7}, 10)$ and $y = (\mathbf{9}, 10, \mathbf{9}, 10)$, then the transfer principle implies that $y \succsim_{\|} x$, because there is a transfer of size $\epsilon = 2$ (i.e. $\frac{x_1 - x_3}{2}$), which allows to have y from x. Combining P-monotony and the Transfer principle leads to the so called *Generalized Lorenz dominance* defined in [5] (for more details see [9,11]).

2.3 Equitable aggregation functions

A usual way to assess the quality of a utility vector is to aggregate the individual utilities with a *collective utility function* (CUF) [14] $G : I\!R^n_+ \rightarrow I\!R_+$, which improves the overall welfare by $max\{G(x) : x \in Q\}$. The G function can be a linear combination of individual utilities (i.e. $G(x) \stackrel{\text{def}}{=} sum(x)$), which is not suitable to fairness context. Another way to build G is based on the `min` function (i.e. $G(x) \stackrel{\text{def}}{=} min(x)$), but it is sensitive to the *drowning effect* [7]. Other refinements of the `min` function exist (e.g. augmented `min`, `lexmin` [3]), but do not really solve the problem, since all are sensitive to *drowning effect*. In order to guarantee equitable aggregations, G should conform to the three equity properties. The most known way is to use Schur-convex function ψ, which are order preserving the three equity properties : $x \succsim_{\|} y \Leftrightarrow \psi(x) \geq \psi(y)$. Precisely, when some aggregation function G is Schur-convex [11], then it is an equitable aggregation [10]. Thus Schur-convex functions play a key role in equitable aggregations (for more details, see [11,10]). In this line of reasoning, we introduce, in the next section, an aggregation function that ensures equity.

2.4 Ordered Weighted Averages (OWA)

This section focuses on the Ordered Weighted Averages (OWA) [23] defined as follows:

$$G^w(x) = \sum_{i=1}^{n} w_i x_{(i)} \tag{1}$$

where $w = (w_1, \ldots, w_n) \in [0, 1]^n$ and $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$. OWA provides a family of compromises between the `sum` and `min`. Golden and Perny [9] propose coefficients for the OWA aggregation method, such that it is Schur-convex:

Theorem 1 *[9] Let be the following coefficients of the OWA aggregation:* $W(x) = \sum_{k=1}^{n} sin(\frac{(n+1-k)\pi}{2n+1}) x_{(k)}$. *$W$ is a Schur-convex function.*

Theorem 1 is fundamental, since that Schur-convex functions ensure equity [10,9].

3 ILP models

This section describes different ILP models for finding an equitable conceptual clustering. Our approach follows the two steps approach of [17]: (1) a dedicated closed itemset mining tool (i.e., `LCM` [22]) is used to compute the set $\mathcal{C}$ of all closed patterns; (2) ILP is used to select a subset of $\mathcal{C}$ that is a partition of the set $\mathcal{D}$ of transactions and that optimizes some given criterion. To enforce equitable clusterings, we enhance the second step with additional constraints enabling to ensure equitable `OWA` aggregation.

3.1 `OWA` ILP models

This section presents our first ILP formulation, called basic `OWA` ILP model, for computing equitable conceptual clusterings using equitable `OWA` operator. Then, we show how this basic model can be improved by post-processing the `OWA` constraints.

Let $\mathcal{D}$ be a dataset with m transactions defined on a set of n items $\mathcal{I}$. Let $\mathcal{C}$ be the set of p closed patterns (w.r.t. the frequency measure) representing the candidate clusters. Let $a_{t,c}$ be an $m \times p$ binary matrix where $(a_{t,c} = 1)$ iff $c \subseteq t$, i.e., the transaction t belongs to extension of the closed pattern c. The $(a_{t,c})$ matrix associated with dataset $\mathcal{D}$ of Table 1a is outlined in Table 1c. Let v be the list of closed pattern utilities (e.g., frequency, diversity, etc.). For each closed pattern $(c \in \mathcal{C})$, a binary variable x_c is associated s.t. $(x_c = 1)$ iff the cluster c is selected.

(a) Basic `OWA` ILP model. Fig. 1a gives the ILP model for equitable conceptual clustering. It uses two types of constraints: conceptual clustering constraints and `OWA` constraints modeling the sorting operation required in the `OWA` operator:

- Conceptual clustering constraints. Constraints (C1) enforce the subset of selected closed patterns to define a partition of $\mathcal{D}$. Constraints (C2) specify a lower bound k_{min} and/or an upper bound k_{max} on the number of selected closed patterns.

- `OWA` **constraints**. The objective function and constraints (O1) and (O2) implement a known linear programming formulation [16] of the `OWA` operator on the conceptual clustering, where the coefficients ω are fixed by theorem 1. As explained in section 2.4, `OWA` is a weighed sum on the sorted utilities. That is why we introduced r, which is equal to the sorted version of the utility vector v. M is a sufficiently large constant. Let z be $|\mathcal{C}|^2$ boolean matrix dedicated only to formulate the sorting constraints (O1) and (O2), which enforce that the utility vector $v \,.\, x$ of the closed patterns are sorted in ascending order matching the `OWA` coefficients ω. These sorting constraints are fully explained in [16]. It follows that the k^{th} smallest utility value r_k will have the k^{th} biggest weight ω_k. The objective function maximizes the weighted sum using `OWA` weights ω given in theorem 1.

(b) Improved `OWA` ILP model. In order to find efficiently an equitable conceptual clustering, we propose the optimize model (see Fig. 1b) as follows:

- Precisely, sorting constraints (O1) and (O2) are specifically used when the utility values are given in comprehension. Fortunately, the utility values of formal concepts are known beforehand. Thus, sorting is performed immediately after finding closed patterns. We use $v^{\uparrow}$, which is the sorted version of v in ascending order.
- We assign the weights ω of equitable `OWA` to the sorted utility values, so that all equal utilities will have the same weight.

For our experiments, we used the improved `OWA` model. Our preliminary results showed that basic `OWA` model performs very poorly compared to the improved `OWA`

$$
\begin{aligned}
&\texttt{Max} \sum_{c=1}^{|\mathcal{C}|} \omega_c \,.\, r_c \\
&\text{s.t.} \begin{cases}
\text{Clustering.} \begin{cases} \text{(C1)} \quad \sum_{c=1}^{|\mathcal{C}|} a_{t,c} \,.\, x_c = 1, \qquad \forall t \in \mathcal{D} \\ \text{(C2)} \quad k_{min} \leq \sum_{c=1}^{|\mathcal{C}|} x_c \leq k_{max} \end{cases} \\
\texttt{OWA}\text{ sorting.} \begin{cases} \text{(O1)} \quad r_c - (v_i \,.\, x_i) \leq M z_{c,i}, \forall\, i, c = 1, ..., |\mathcal{C}| \\ \text{(O2)} \quad \sum_{i=1}^{|\mathcal{C}|} z_{c,i} \leq c - 1, \qquad \forall\, c = 1, ..., |\mathcal{C}| \end{cases} \\
x_c \in \{0,1\},\ r_c \in I\!R_+, \quad \forall\, c = 1, ..., |\mathcal{C}| \\
z_{c,i} \in \{0,1\}, \quad \forall\, i, c = 1, ..., |\mathcal{C}|
\end{cases}
\end{aligned}
$$

(a) Basic `OWA` ILP model.

$$
\begin{aligned}
&\texttt{Max} \sum_{c=1}^{|\mathcal{C}|} \omega_c \,.\, (v_c^{\uparrow} \,.\, x_c^{\uparrow}) \\
&\text{s.t.} \begin{cases} \text{(C1)}, \quad \text{(C2)} \\ x_c \in \{0,1\}, \\ \quad \forall\, c = 1, ..., |\mathcal{C}| \end{cases}
\end{aligned}
$$

(b) Improved `OWA` ILP model.

Fig. 1: `OWA` ILP models for equitable conceptual clustering.

model in terms of CPU-times. This meanly due from the fact that (n^2) additional constraints and (n^2) additional variables are used to encode the `OWA` sorting constraints. This constitutes a strong limitation of the size of the databases that could be managed.

Proposition 1. *Basic and improved* `OWA` *ILP models are equivalent.*

Proof. Both `OWA` models use weights ω given in Theorem 1, which ensure an equitable aggregation. Improved `OWA` is an optimization of the basic model: (1) It uses an a priori sorting of utilities (no need to sorting constraints); (2) The same weight is assigned to equal utilities (the same satisfaction level), which preserves straightforwardly the conformity with theorem 1. Thus, both `OWA` models are equivalent. □

(c) ILP numerical stability. The set of extracted closed patterns is mostly huge, which leads to a huge `OWA` vector ω in the basic model, and affect the numerical stability of the ILP solver. The optimized `OWA` model tackles this issue, thanks to assigning the same weight to all equal utilities. This makes it possible to solve real-world instances in our experiments reported in Section 5.

3.2 Other ILP models

As described in section 2.1 a linear aggregation of individual utilities $max\{sum(x) : x \in Q\}$, does not fit the equity requirement. This suggests resorting to non-linear aggregation operators, especially the `maxMin` and `minDev`. The `maxMin` aggregation $max\{min(x) : x \in Q\}$ tackles equity by improving the worst utility. This function can be linearized by maximizing a decision variable $z \geq 0$, that is a lower bound for the utility vector $v \,.\, x$ (see Fig. 2a, inequality C3), where v is the clustering criterion to be optimized (e.g. frequency). Thus, the linear formulation for the conceptual clustering is given by the ILP model of Figure 2a.

An alternative way of ensuring equity is by achieving maximum deviation minimization `minDev` between both the best and the worst utilities: `Min` $\{\texttt{max}(x) - \texttt{min}(x) : x \in Q\}$. It can be linearized by introducing $2 \times n$ constraints and two decision variables $z_{max} \geq 0$ and $z_{min} \geq 0$ to maintain the `max` and the `min` values of the utility vector $v \,.\, x$ (see Fig. 2b, inequalities C4-C5). The resulting ILP model is given in Fig. 2b.

4 Related work

Heuristic approaches Several methods have explored the idea of separating cluster-formation from finding the conceptual descriptions. Pensa *et al.* [19] begin by mining

$$\texttt{Max}\ z \quad \text{s.t.} \begin{cases} \text{(C1)}, \quad \text{(C2)} \\ \textbf{(C3)} \quad z \leq v_c \,.\, x_c, \forall\, c = 1, ..., |\mathcal{C}| \\ x_c \in \{0,1\}, \qquad \forall\, c = 1, ..., |\mathcal{C}| \\ z \geq 0 \end{cases}$$

(a) `maxMin` ILP model.

$$\texttt{Max}\ z_{max} - z_{min} \quad \text{s.t.} \begin{cases} \text{(C1)}, \quad \text{(C2)} \\ \textbf{(C4)} \quad z_{max} \geq v_c \,.\, x_c, \forall\, c = 1, ..., |\mathcal{C}| \\ \textbf{(C5)} \quad z_{min} \leq v_c \,.\, x_c, \forall\, c = 1, ..., |\mathcal{C}| \\ x_c \in \{0,1\}, \qquad \forall\, c = 1, ..., |\mathcal{C}| \\ z_{max} \geq 0,\ z_{min} \geq 0 \end{cases}$$

(b) `minDev` ILP model.

Fig. 2: ILP models for the conceptual clustering.

closed (or δ-closed) patterns (itemsets) and their extensions and then perform k-Means clustering on them. Perkowitz and Etzioni [20], reverse the two phases: their *cluster-mining* first uses a clustering technique to form clusters. From the resulting clustering, descriptions are learned by a rule-learning technique. All those techniques are of a heuristic nature and produce results of varying quality. Moreover, they are heavily influenced by the initialization conditions, typically requiring numerous restarts, increasing computational costs.

Declarative approaches. Recently, [17,18] have developed declarative frameworks using ILP, which can find optimal conceptual clusterings, where clusters correspond to concepts. Later, Chabert *et al.* have introduced two new CP models for computing optimal conceptual clusterings. The first model (denoted `FullCP2`) may be seen as an improvement of [6]. The second model (denoted `HybridCP`) follows the two step approach of [17] : the first step is exactly the same; the second step uses CP to select formal concepts. Our work is different in that we study the setting where each clustering must fulfills equity requirements.

Distance-based clustering aims at finding homogeneous clusters only based on a dissimilarity measure between objects. Different declarative frameworks have been developed, which rely on CP [6] or ILP [1,15]. There are a few existing approaches for obtaining balanced clusters. The most prominent one is the approach proposed by [2]. Our adoption of closed patterns cuts down on redundancy compared to other ways of selecting candidate clusters. Moreover, our use of an equitable `OWA` gives stronger guarantees about the obtained clusterings in terms of balancing.

5 Experiments and Results

The experimental evaluation is designed to address the following questions:

1. How do the ILP models compare and scale on the considered datasets?
2. How do the resulting clusters and their description compare qualitatively?
3. How (in terms of CPU-times) does our ILP model compares to the CP models of Chabert *et al.* [4]?

Experimental protocol. All experiments were conducted on Linux cluster[3], where each node has a dual-CPU Xeon E5-2650 with 16 cores, 64 GB RAM, running at 2.00GHz.

[3] `http://www.rx-racim.cerist.dz/?page_id=26`.

Dataset	$\#\mathcal{D}$	$\#\mathcal{I}$	Density(%)	$\#\mathcal{C}$
Soybean	630	50	32	31,759
Primary-tumor	336	31	48	87,230
Lymph	148	68	40	154,220
Vote	435	48	33	227,031
tic-tac-toe	958	27	33	42,711
Mushroom	8124	119	18	221,524
Zoo-1	101	36	44	4,567
Hepatitis	137	68	50	3,788,341
Anneal	812	93	45	1,805,193

(a) UCI datasets.

Dataset	$\#\mathcal{D}$	$\#\mathcal{I}$	Density(%)	$\#\mathcal{C}$
ERP-1	50	27	48	1,580
ERP-2	47	47	58	8,1337
ERP-3	75	36	51	10,835
ERP-4	84	42	45	14,305
ERP-5	94	53	51	63,633
ERP-6	95	61	48	71,918
ERP-7	160	66	45	728,537

(b) ERP datasets.

Table 2: Dataset characteristics. Each row gives the number of transactions (#$\mathcal{D}$), the number of items (#$\mathcal{I}$), the density and the number of closed patterns extracted (#$\mathcal{C}$).

$\mathcal{D}$	k	OWA		minDev		maxMin		maxSum	
		ICS	ICD	ICS	ICD	ICS	ICD	ICS	ICD
Soybean	3	0.447	**0.784**	0.447	**0.784**	**1.000**	0.026	**1.000**	0.026
	4	0.331	**0.865**	0.331	**0.865**	**1.000**	0.026	**1.000**	0.026
	5	0.259	0.895	0.284	**0.905**	**1.000**	0.026	**1.000**	0.026
	6	0.231	**0.940**	0.231	**0.940**	**1.000**	0.026	**1.000**	0.026
	7	0.195	**0.964**	0.195	**0.964**	**0.959**	0.108	**0.959**	0.108
	8	0.186	**0.987**	0.186	**0.987**	0.671	0.474	**0.959**	0.108
	9	0.166	**1.000**	0.166	**1.000**	0.671	0.474	**0.959**	0.108
	10	0.136	**0.999**	0.142	**0.999**	0.670	0.474	**0.959**	0.108

(a) Maximizing frequency.

$\mathcal{D}$	k	OWA		minDev		maxMin		maxSum	
		ICS	ICD	ICS	ICD	ICS	ICD	ICS	ICD
soybean	3	0.447	**0.776**	0.447	**0.776**	**1.000**	0.026	0.447	0.776
	4	0.334	0.839	0.338	**0.854**	**1.000**	0.026	0.406	0.831
	5	0.296	**0.900**	0.301	**0.900**	**1.000**	0.026	0.389	0.843
	6	0.257	0.929	0.265	**0.934**	**1.000**	0.026	0.398	0.851
	7	0.240	**0.956**	0.240	**0.956**	**0.959**	0.106	0.330	0.909
	8	0.220	0.971	0.198	**0.978**	**0.959**	0.106	0.323	0.918
	9	0.183	**0.991**	0.184	0.989	0.959	0.106	0.216	**0.975**
	10	0.170	0.999	0.157	**1.000**	0.959	0.106	0.213	**0.980**

(b) Minimizing diversity.

Table 3: Comparing the quality of the resulting clusterings in terms of ICS and ICD.

We used `LCM` to extract all closed patterns and `CPLEX` v.12.6.1 to solve the different ILP models. For all methods, a time limit of 24 hours has been used.

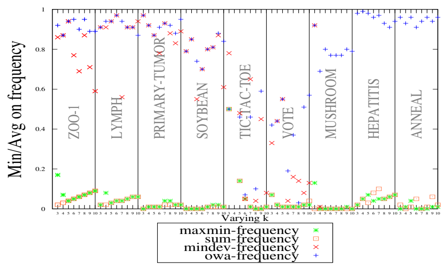
Test instances. We used classical ML datasets, coming from the UCI database. We have also considered the same datasets (called ERP-i, with $i \in [1, 7]$) used in [4] and coming from a real application case[4], which aims at extracting setting concepts from an Enterprise Resource Planning (ERP) software corresponding to groups of parameter settings groups of parameter settings. Table 2 shows the characteristics of all datasets.

To evaluate the quality of a clustering, we test the coherence of a clustering, measured by the intra-cluster similarity *(ICS)* and the inter-clusters dissimilarity *(ICD)*, both of which should be as large as possible. Given a similarity measure s between two transactions t and t', where $s : \mathcal{D} \times \mathcal{D} \mapsto [0, 1]$, $s(t, t') = \frac{|t \cap t'|}{|t \cup t'|}$, $ICS(P_1, ..., P_k) = \frac{1}{2} \sum_{1 \leq i \leq k} (\sum_{t,t' \in P_i} s(t, t'))$ and $ICD(P_1, ..., P_k) = \sum_{1 \leq i < j \leq k} (\sum_{t \in P_i, t' \in P_j} (1 - s(t, t')))$

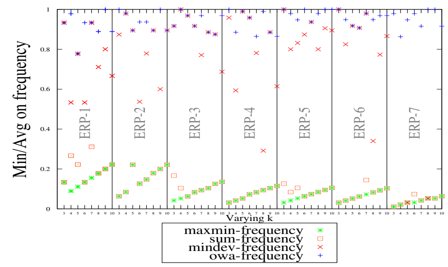
To evaluate how well equitable the clusters are w.r.t frequency, we used three measures: (1) the ratio between the frequency of the smallest cluster to the average cluster frequency (i.e. Min/Avg). For m transactions put into k clusters, Avg is just (m/k); (2) the *Standard Deviation* in cluster frequencies (i.e. $StdDev$); (3) the deviation between the smallest and the largest description of selected concepts (i.e. $devSize$).

(a) Qualitative analysis of clusterings. Fig. 3a compares qualitatively the resulting clusterings of the different ILP models for various values of k on UCI datasets according to the Min/Avg measure. `maxMin` and `maxSum` performs very poorly in terms

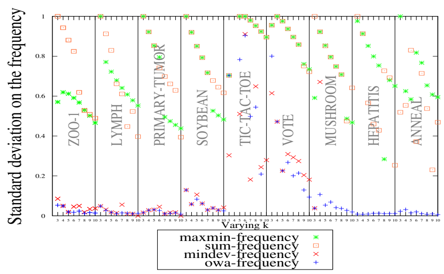
[4] These datasets are available on `http://liris.cnrs.fr/csolnon/ERP.html`.



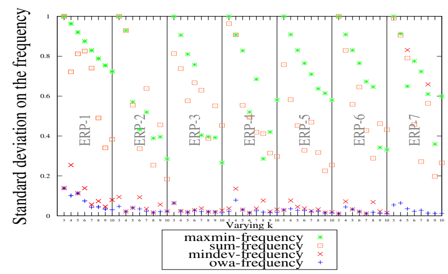
(a) Evaluating (Min/Avg) on UCI datasets.



(b) Evaluating (Min/Avg) on ERP datasets.



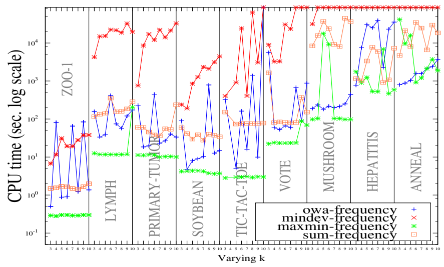
(c) Evaluating $StdDev$ on UCI datasets.



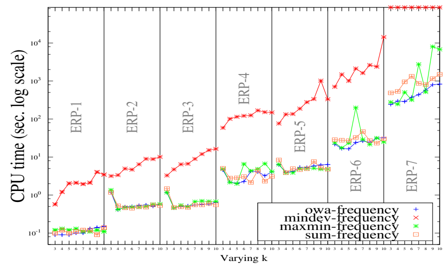
(d) Evaluating $StdDev$ on ERP datasets.

Fig. 3: Quality of balancing of the resulting clusterings of the different ILP models.

of balancing compared to OWA and minDev (maxMin and maxSum always achieve lower Min/Avg values). Interestingly, both OWA and minDev almost achieve similar performance on datasets with number of closed patterns comprise between 10^3 and 10^5. However, for the three most difficult datasets – Mushroom, Hepatitis and Anneal – the disparity between both models become more pronounced: OWA always obtains more equitable clusterings (Min/Avg values close to 1). On these datasets, minDev fails to find a solution even for small values of k. The same behavior is observed on ERP datasets (see Fig. 3b). On ERP-7, minDev was not able to find a solution. This is in part explained by the number of closed patterns (10^6) in comparison to the other ERP instances (from 10^3 to 10^5). When considering $stdDev$ measure (see Figs. 3c and 3d), OWA and minDev achieve the lowest $StdDev$ on all datasets, but OWA performs marginally better than minDev. When examining the description sizes (see Supp. material [12]), we can see that maxMin and maxSum lead to higher $devSize$ values. This is indicative of one (or few) clusters of large frequencies and small description sizes, or clusters of large description sizes and small frequencies. These results are consistent with our previous conclusions. However, for minDev and OWA, the optimal solutions found by both models tend to offer a better compromises between the two criteria. Finally, Tab. 3 compares the four models according to *ICS* and *ICD* (see Supp. material [12]). We can see that minDev and OWA sacrify *ICS* to achieve higher *ICD* values.



(a) UCI datasets: maximizing frequency.



(b) ERP datasets: maximizing frequency.

Instance	OWA with k not fixed $k \in [3, 10]$		OWA with k fixed		OWA with k not fixed $k \in [3, \vert\mathcal{D}\vert - 1]$	
	best k	Time (s.) (2)	best k	Time (s.) (2)	best k	Time (s.) (2)
Soybean	10	27.09	10	**14.82**	501	15.76
Primary-tumor	10	26.81	10	33.34	215	**14.52**
Lymph	10	77.97	10	173.00	147	**20.61**
Vote	10	89.8	10	879.22	342	**42.3**
tic-tac-toe	9	2,104.07	9	**9.95**	956	11.07
Mushroom	10	**377.21**	10	442.34	8,123	982.95
Zoo-1	10	5.47	10	1.37	59	**0.8**
Hepatitis	10	8,462.45	10	35,498.2	136	**607.51**
Anneal	10	3,674.89	10	3,666.82	459	**1,453.04**

(c) Maximizing frequency.

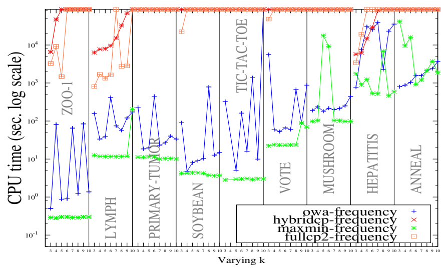
Instance	OWA with k not fixed $k \in [3, 10]$		OWA with k fixed		OWA with k not fixed $k \in [3, \vert\mathcal{D}\vert - 1]$	
	best k	Time (s.) (2)	best k	Time (s.) (2)	best k	Time (s.) (2)
Soybean	10	13.7	10	165.42	501	**9.61**
Primary-tumor	10	46.19	10	210.01	215	**18.5**
Lymph	10	123.84	10	569.63	145	**22.05**
Vote	10	146.72	10	786.84	342	**45.7**
tic-tac-toe	9	37,882.31	9	293.82	956	**7.21**
Mushroom	10	**274.62**	10	667.99	8,123	1,086.13
Zoo-1	10	0.89	10	1.82	59	**0.8**
Hepatitis	10	37,915.3	8	6,275.23	136	**630.91**
Anneal	10	6,839.68	10	25,760.25	459	**2,311.01**

(d) Minimizing diversity.

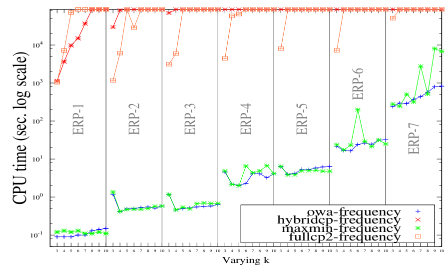
Fig. 4: CPU-times analysis.

This is indicative of more balanced clusters: the *ICS* is necessarily limited by the number of instances per cluster but the *ICD* increases if there are more instances in other clusters to compare against. maxMin and maxSum show the opposite behavior, which is indicative of one (or a few) large clusters, and numerous smaller ones.

(b) Scale-up property analysis. Figs.4a and 4b compare the CPU-times for computing optimal clusterings for various values of k on UCI and ERP datasets when maximizing the sum of frequencies of the selected concepts. The CPU-times include the time spent by LCM to extract all closed patterns. On UCI datasets, minDev performs very poorly compared to the other ILP models. Although the qualitative results of minDev are satisfactory, this model remains hampered by long solving times: it goes beyond the timeout on 32 instances (out of 72), particularly on the three most difficult datasets Mushroom, Hepatitis and Anneal (see Fig. 4a). This probably stems from the fact that $(2 \times n)$ additional constraints are used to capture the minimal deviation. However, OWA yields quite competitive results, while achieving optimal equitable clusterings (see the qualitative analysis). It is able to solve all instances and comes in second position. Overall, maxMin gets the best performances. However, as noticed above, the optimal solutions found are far to be equitable ones; they correspond to extreme solutions (worst cases). This probably explains in part the good behaviour of maxMin model. The same behavior is observed for minDev on ERP datasets. Finally, the three ILP models − OWA, maxMin and maxSum − perform very similarly on all instances. We conclude that OWA model offers a good compromise between solution quality and computing time.



(a) UCI datasets: maximizing frequency.



(b) ERP datasets: maximizing frequency.

Fig. 5: Comparing CPU-times of `maxMin` ILP model with the two CP models.

(c) ILP models vs. CP based models. Figs. 5a and 5b compare the performance of `maxMin` ILP model with the two CP models (`FullCP2` and `HybridCP`) maximizing the minimal frequency of a cluster on UCI and ERP datasets. The CPU-times of `HybridCP` include those for the preprocessing step. `maxMin` ILP model outperforms `FullCP2` and `HybridCP` by several orders of magnitude on all datasets. None of the two CP models scales well for this objective: they fail to find a solution within the time limit for ($k \geq 4$), except for 4 datasets. Moreover, `OWA` ILP model clearly beats the two CP models. Finally, notice that `FullCP2` performs marginally better than `HybridCP`.
(d) `OWA` **model with** k **not fixed**. Our third set of experiments aims at evaluating `OWA` model capability for finding the optimal solution when k is not fixed. For this aims, we selected two settings: $k \in [3, 10]$ (`OWA`-1) and $k \in [3, |\mathcal{D}| - 1]$ (`OWA`-2). Fig. 4c and 4d compare the CPU-times when k is not fixed (Columns 3 and 7), and when k is fixed (Col. 5) on UCI datasets. Col. 4 reports the best values found for k ($3 \leq k \leq 10$) that optimize both objectives. For all datasets but two, `OWA`-1 and `OWA`-2 are the best performing approaches. `OWA`-1 is able to solve 5 (resp. 7) instances quicker when maximising the frequency (resp. diversity). Interestingly, `OWA`-1 and `OWA` (with k fixed) always agree on the best value for k. Compared to `OWA`-1, `OWA`-2 scales well, particularly on the two most difficult datasets Anneal and Hepatitis (speed-up of up to 60.09). Indeed, larger values of k enable to find balanced clustering more quickly than for smaller values of k: there $|\mathcal{D}| - 1$ clusters for 3 datasets, whereas for the remaining datasets the value of k is rather high.

6 Conclusion

We have proposed an efficient approach for equitable conceptual clustering that uses closed itemset mining to discover candidates for descriptions, and ILP implementing an equitable aggregate function based on `OWA` to select the best clusters of balanced frequencies. Contrary to `maxMin` and `minDev` operators, our approach offers a good compromise between solution quality and computing time. We plan to investigate multi-criteria conceptual clustering, where the utilities are not comparable. Exploiting equity constraints within approximate approaches could become interesting to tackle very large datasets.

References

1. B. Babaki, T. Guns, and S. Nijssen. Constrained clustering using column generation. In *CPAIOR 2014*, pages 438–454, 2014.
2. A. Banerjee and J. Ghosh. Scalable clustering algorithms with balancing constraints. *Data Min. Knowl. Discov.*, 13(3):365–395, 2006.
3. Sylvain Bouveret and Michel Lemaître. Computing leximin-optimal solutions in constraint networks. *Artif. Intell.*, 173(2):343–364, 2009.
4. M. Chabert and C. Solnon. Constraint programming for multi-criteria conceptual clustering. In *CP 2017*, volume 10416 of *LNCS*, pages 460–476. Springer, 2017.
5. K. M. Chong. An induction theorem for rearrangements. *Candadian Journal of Mathematics*, 28:154–160, 1976.
6. T-B-H Dao, K-C Duong, and C. Vrain. Constrained clustering by constraint programming. *Artif. Intell.*, 244:70–94, 2017.
7. Didier Dubois and Philippe Fortemps. Computing improved optimal solutions to max-min flexible constraint satisfaction problems. *EJOR*, 118:95–126, 1999.
8. D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.
9. B. Golden and P. Perny. Infinite order lorenz dominance for fair multiagent optimization. In *AAMAS*, pages 383–390, 2010.
10. Michael M. Kostreva, Wlodzimierz Ogryczak, and Adam Wierzbicki. Equitable aggregations and multiple criteria analysis. *EJOR*, 158(2):362–377, 2004.
11. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and its Applications*. Academic Press, London, 1979.
12. Supplementary Material. Available at `http://`.
13. R. S. Michalski and R. E. Stepp. Learning from observation: Conceptual clustering. In *Machine Learning*, pages 331–363. Springer, 1983.
14. Hervi Moulin. *Axioms of Cooperative Decision Making*. Number 9780521360555 in Cambridge Books. Cambridge University Press, 1989.
15. M. Mueller and S. Kramer. Integer linear programming models for constrained clustering. In *DS 2010*, pages 159–173, 2010.
16. Włodzimierz Ogryczak and Tomasz Śliwiński. On solving linear programs with the ordered weighted averaging objective. *EJOR*, 148(1):80 – 91, 2003.
17. A. Ouali, S. Loudni, Y. Lebbah, P. Boizumault, A. Zimmermann, and L. Loukil. Efficiently finding conceptual clustering models with integer linear programming. In *IJCAI 2016*, pages 647–654, 2016.
18. A. Ouali, A. Zimmermann, S. Loudni, Y. Lebbah, B. Crémilleux, P. Boizumault, and L. Loukil. Integer linear programming for pattern set mining; with an application to tiling. In *PAKDD 2017*, pages 286–299, 2017.
19. R. G. Pensa, C. Robardet, and J-F. Boulicaut. A bi-clustering framework for categorical data. In *PKDD 2005*, pages 643–650, 2005.
20. M. Perkowitz and O. Etzioni. Adaptive web sites: Conceptual cluster mining. In *IJCAI 99*, pages 264–269, 1999.
21. A.K. Sen and J.E. Foster. *On economic inequality*. Clarendon Press, Oxford, 1997.
22. T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *DS 2004*, pages 16–31, 2004.
23. Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Systems, Man, and Cybernetics*, 18(1):183–190, 1988.
24. Y. Yang and B. Padmanabhan. Segmenting customer transactions using a pattern-based clustering approach. In *ICDM 2003*, pages 411–418, 2003.