Lexicographical binary implementation of the Recurrent Pareto Filter in categorization procedures

A. AMELJAŃCZYK
andrzej.ameljanczyk@wat.edu.pl
Ch. TRAN QUANG
leadotc@gmail.com

Military University of Technology, Faculty of Cybernetics
Institute of Computer and Information Systems
Kaliskiego Str. 2, 00-908 Warsaw, Poland

The paper presents the possibility of using Recurrent Pareto Filter (RPF) to the categorization procedures of objects (data). The paper presents a new implementation of the RPF algorithm, that uses lexicographical sorting objects and binary search Pareto optimal elements. The functioning of the algorithm illustrated by an example categorization procedure of scientific journals contained in the Scimago Scientific Journals Base.

Keywords: Pareto filter, data clustering, multi-criteria ranking, categorization of objects, recurrent Pareto filter.

1. Introduction

The work is a direct continuation of the papers [1, 2, 5, 11] pursuant to the categorization procedures of objects. Categorization procedure of objects is understood as a generalization of multi-criteria ranking a set of objects [1, 5, 9, 11]. Let therefore $Y \subset \mathcal{R}^N$ – non--empty, finite set of elements (objects), which is to be the ranking procedure. $R \subset Y \times Y$ – precedence (rankig) relation, understood as follows: the pair (y, z) belongs to the relation if and only if "element y is before the element z". Sentence "y is before z" (or "y precedes z") can be understood very widely. Frequently it is understood in the context of quality "y is better than z" [2], [3]. A relation *R* is sometimes called the relation of preferences (precedence) or ranking relation. Pair (*Y, R*) will be called a set with relation [2]. Pareto relation is defined as follows:

$$R = \left\{ (y,z) \in Y \times Y \,\middle|\, y_n \geq z_n,\ n \in \mathcal{N} \right\} \quad (1)$$

where $\mathcal{N} = \{1,2,...,n,...,N\}$.

The Pareto Filter (PF) is an algorithm enabling determination from any set of elements the subset of elements of 'the highest quality' in this set (in the meaning of Pareto relation) [2, 3, 6]. The effect (result) Y_N^R of applying the Pareto filter on set *Y* is so-called 'Pareto front' (set of nondominated (minimal)) elements in the meaning of Pareto relation defined as follows:

$$Y_N^R = \left\{ \begin{array}{l} y \in Y \,\middle|\, \text{does not exists} \\ z \in Y - \{y\},\ \text{such } (z,y) \in R \end{array} \right\} \quad (2)$$

Therefore, the result of the filtration process is decisive for the adopted preferences (filtration) relation R (in more detail – its properties). So, such a relation is frequently (commonly) called a preference filter or briefly: filter. The general reflection of the Pareto filter is a cone filter (CF), in which the filtration reaction is generated by a cone [1, 2, 3, 5, 12, 13].

Fig. 1. Pareto Filter

The other known preference principle is the lexicographic principle [3, 4, 7, 8] (considering the order (importance, hierarchy) of objectives. Its basis is formed by the set of permutations of set $\mathcal{N}$. Each lexicographic relation $\mathcal{L}$ leads to the linear ordering of set $Y \subset \mathcal{R}^N$ [1, 8]. Relation $\mathcal{L}$ can be defined as follows:

$$\mathcal{L} = \left\{ \begin{array}{l} (y,z) \in Y \times Y, exists\ k \in \mathcal{N}, \\ that\ y_k > z_k\ and\ y_l = z_l, l < k \end{array} \right\} \quad (3)$$

Analogical for all other permutations of set of objectives numbers $\mathcal{N}$.

2. Recurrent Pareto filter (RPF)

Using by recurrent way, Pareto filter (PF) for the filtration of *Y* set leads [5] to the division of Y set to the categories (clusters) [1, 9, 10, 11]. The effect of operation of the RPF is a sequence of clusters (categories) [1, 5]:

$$r(Y)=\left(Y_N^R(k), k=1,2,...,K\right) \tag{4}$$

where:

$$Y_N^R(k)=\left(Y-\bigcup_{m=0}^{m=k-1} Y_N^R(m)\right)_N^R. \tag{5}$$

Figure 2 shows the RPF scheme.

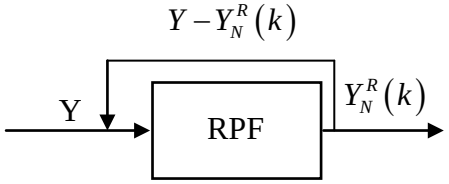

Fig. 2. Recurrent Pareto Filter (RPF)

A set $Y_N^R(k)$ is called a category (cluster) number (*rank*) *k*. Figure 3 shows the flowchart of the RPF.

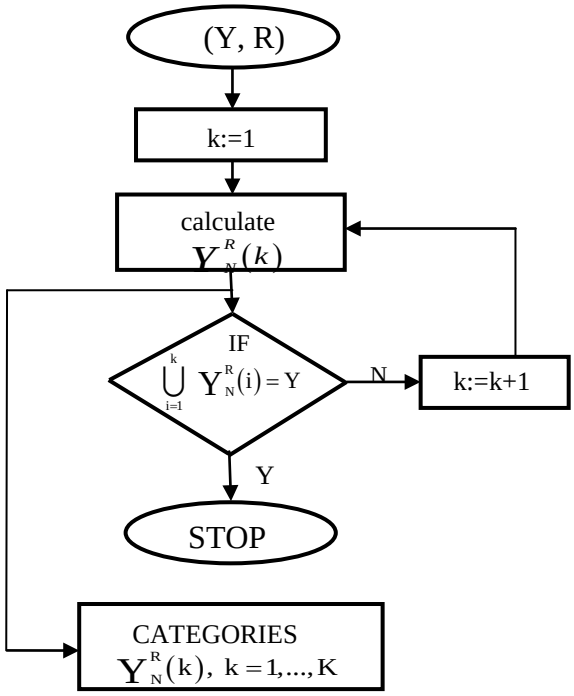

Fig. 3. Flowchart of the RPF

The main outcome of this work is to propose a new, faster implementation of the recurrent Pareto filter applied to procedures for categorizing a set of objects Y. One proposed the algorithm called Lexicographical Binary Sorted Algorithm (LBS) uses lexicographical pre-sorting of Y to accelerate the algorithm RPF.

3. Lexicographical Binary Sort Algorithm (LBS)

As the name mentioned, algorithm LBS is a combination of lexicographical sort and binary search. Algorithm LBS uses order property of elements after applied sorting lexicographically in finding ranking of elements using binary search. LBS method uses next properties lexicographical relation $\mathcal{L}$ and Pareto relation R [3,4]:

a) $R\subset\mathcal{L}$ (if $(y,z)\in R$ so $(y,z)\in\mathcal{L}$);

b) $Y_N^{\mathcal{L}}\subset Y_N^R$ (each lexicographical solution is nondominated in Pareto sense as well).

For the convenience of recording further the Pareto relation R will be denoted by symbol $\succ$

$$y^i\succ y^j \Leftrightarrow y_n^i\geq y_n^j,\ n=1,2,...,N \tag{6}$$

Example 1 (*M* =10, *N* = 5)
Assume that, as result of initial sorting process original set *Y* is stored in the list $L=\left\langle y^1,y^2,y^3,...,y^M\right\rangle$
where:

$$y^i=\left(y_1^i,y_2^i,...,y_N^i\right),\ i=1,...,M \tag{7}$$

In general, we can use arbitrary preference but for simplicity, in below example the input set was sorted using preference order (1, 2, 3,…, *N*). It means that the first objective is the most important, next is the second one etc. So we have:

$y^1=(8,10,9,9,9)$
$y^2=(8,9,8,10,7)$
$y^3=(7,8,8,8,9)$
$y^4=(6,6,6,7,2)$
$y^5=(5,6,4,4,6)$
$y^6=(5,2,2,8,6)$
$y^7=(4,5,2,3,3)$
$y^8=(2,3,7,4,9)$
$y^9=(2,1,1,0,2)$
$y^{10}=(0,1,0,0,0)$

Because the original set was sorted lexicographically, so we are guaranteed if an

element y^i stays before another element y^j in sorted list then exists: $0 < k \le N$

that $y_k^i > y_k^j$ *and* $y_m^i = y_m^j$ *for* $m < k$, on the other hand in sorted list we cannot guarantee that, y^i is better than y^j in all objectives (in Pareto sense) [3, 4].

But, if an element y^j is dominated by another element y^i (in Pareto sense), we can sure that $i < j$ in sorted list. Moreover, if we iterate to element y^j in order of sorted list, we can sure that, every elements y^i ($i < j$) were assigned to proper cluster (had final ranking). From this observation, we can reduce the number of comparing operations while finding ranking for element y^j by checking dominance of y^j only with other elements y^i ($i < j$).

From definition of ranking task, element y^{l_k} has rank $r^l = k$, if and only if exists other element $y^{l_{k-1}}$ which has rank $r^{l_{k-1}} = k-1$, so that $y^{l_{k-1}} \succ y^{l_k}$, which means that exists $y^{l_{k-2}} \succ y^{l_{k-1}}$ and $r^{l_{k-2}} = k-2$ and so on. In other words, exists a dominance chain:

$y^{l_1} \succ y^{l_2} \succ ... \succ y^{l_{k-1}} \succ y^{l_k}$

where $r^{l_1} = 1, r^{l_2} = 2, ..., r^{l_k} = k$

Our problem is finding the shortest dominance chain for every element.

Before going further, we define domination of a cluster against an element. A k-th cluster $Y_N^R(k)$, which contains all elements with rank *k* is said that dominating element y^j ($Y_N^R(k) \succ y^j$) if and only if exists $y^i \in Y_N^R(k)$, so that $y^i \succ y^j$. From this definition, we can see that, if an element y^j isn't dominated by cluster $Y_N^R(k)$ so it also can't be dominated by another cluster $Y_N^R(k')$ with $k' > k$. Because in other way, if y^j is dominated by $Y_N^R(k')$ hence exists at least one dominance chain:

$y^{i_1} \succ y^{i_2} \succ ... \succ y^{i_k} \succ y^{i_{k+1}} \succ ... \succ y^{i_{k'}} \succ y^{i_j}$

In another word, cluster $Y_N^R(k)$ must dominates y^j. From there we can apply the idea of binary search and LBS algorithm which can be described as follow:

Step 1

Lexicographically sorting set *Y* (original list of objects).

Step 2

For every element y^j, $j = 1,...,M$ use binary search to find the biggest number *k* so that cluster $Y_N^R(k) \succ y^j$

where $k = 1,..., \max\left(r^1, r^1, ..., r^{j-1}\right)$ and assign $r^j = k+1$. If there is not such as number *k*, assign $r^j = 1$.

Step 3

Present final rank of all elements.

For mentioned example, we can illustrate algorithm LBS as follow:

Tab. 1. Algorithm LBS – a running example

j	*k*	r^j	*comment*
1		1	does not exist *k*
2		1	does not exist *k*
3	1	2	$y^1 \succ y^3$
4	2	3	$y^3 \succ y^4$
5	2	3	$y^3 \succ y^5$
6	2	3	$y^3 \succ y^6$
7	3	4	$y^5 \succ y^7$
8	2	3	$y^3 \succ y^8$
9	4	5	$y^7 \succ y^9$
10	4	5	$y^7 \succ y^{10}$

The sequence of clusters (categories) received by LBS algorithm is as follows:

$Y_N^R(1) = \{y^1, y^2\}$ – (*gold category*)

$Y_N^R(2) = \{y^3\}$ – (*silver category*)

$Y_N^R(3) = \{y^4, y^5, y^6, y^8\}$ – (*bronze category*)

$Y_N^R(4) = \{y^7\}$, *etc.*

$Y_N^R(5) = \{y^9, y^{10}\}$

A Java implementation of algorithm LBS will be applied to a particular example in next section. In addition, other randomized tests also will be used. *Java's Collections.sort* was used to lexicographical sort the original list. All tests are run on same computer with *Java 8 64-bit* update 60.

4. A case study

Web portal SCIMAGO journals rank (http://www.scimagojr.com) collects a set of journals' ranking, according many various criterions. Information about journals are collected in a certain period of time. Journals in this database are sorted by SCIMAGO SJR index which is a measure of journal's impact, influence or prestige [14]. It expresses the average number of weighted citations received in the selected year by the documents published in the journal in the three previous years. Base on collected information in this database we can rank these journals in other perspective, in which every criterion is treated equally. In other words, we are trying multi-objectives sorting the journals.

Example 2
For illustratable purpose, to sort top 10 journals by *SJR index*, we consider only two criterions, for example: *H index* and *Total Refs.*

Tab. 2. Top 10 journals sorted by SJR index

Rank	Title	SJR
1	Ca-A Cancer Journal for Clinicians	37,384
2	Reviews of Modern Physics	29,826
3	Annual Review of Immunology	28,577
4	Nature Reviews Molecular Cell Biology	24,294
5	Nature Reviews Genetics	23,991
6	Cell	23,588
7	Quarterly Journal of Economics	22,541
8	Nature Reviews Immunology	22,472
9	Nature Reviews Cancer	21,831
10	Annual Review of Astronomy and Astrophysics	21,109

Tab. 3. H index and Total Refs. of top 10 journals sorted by SJR index

Rank	H index	Total Refs.
1	108	2888
2	233	9315
3	244	4220
4	302	8882
5	246	8009
6	585	30034
7	171	1620
8	267	8279
9	297	9722
10	132	4231

Result, when apply recurrent Pareto filter (RPF) mentioned in section 2 (in the *Brute Force version* (BF) [11], see Fig. 3 also) as follow:

Rank 1 (*gold category*):
- Cell

Rank 2 (*silver category*):
- Nature Reviews Molecular Cell Biology
- Nature Reviews Cancer

Rank 3 (*bronze category*):
- Nature Reviews Immunology
- Reviews of Modern Physics

Rank 4 (*etc.*…):
- Nature Reviews Genetics

Rank 5:
- Annual Review of Immunology
- Annual Review of Astronomy and Astrophysics

Rank 6:
- Quarterly Journal of Economics
- Ca-A Cancer Journal for Clinicians.

The same result was obtained when using LBS algorithm. To test performance, we run LBS algorithm against others randomized test cases with various size of data. In all tests, LBS algorithm generated proper result within significant reduction runtime (measured in seconds).

Tab. 4. Runtime examples of algorithm LBS

M	N	RPF(BF)	LBS
7559	2	0.745	0.141
4105	3	0.290	0.050
6441	4	1.162	0.103
8312	5	1.283	0.197
2805	6	0.254	0.055

5. Conclusion

The main outcome of the work is a proposal a new, faster implementation of the recurrent Pareto filter, applied to procedures for categorizing set of objects Y. One proposed the algorithm, called Lexicographical Binary Sorted (LBS) uses lexicographical pre-sorting of Y to

accelerate the algorithm RPF. In decision making problem, when we don't know additional information about meaning of criterions or these criterions have the same effect on final decision, applying scalar methods (like using *SJR index* when ranking journals in SCIMAGO database) depends on preference of decision maker. However, applying Pareto relation can give us more fair result while treat criterions equally. Proposed algorithm LBS solved this problem: assign elements of original set to proper cluster so that an element of k-th cluster is dominated in Pareto definition by at least one element in (k-1)-th cluster. In addition, algorithm LBS take advantages of sorting lexicographical and binary search to reduce complexity of algorithm. Section 4 presents the results of tests using the proposed procedures for categorizing by the LBS implementation and algorithm the RPF in *Brute Force* version (RPF) (BF). Achieved results confirm the significant advantage LBS algorithm. The presented method can be used in the procedures of categorization of any set of objects which are multicriterial evaluated.

6. References

[1] Ameljańczyk A., "Mathematical aspects of ranking theory", *Computer Science and Mathematical Modelling*, No. 2, 5–10 (2015).

[2] Ameljańczyk A., "Pareto filter in the process of multi-label classifier synthesis in medical diagnostics support algorithms", *Computer Science and Mathematical Modelling* , No. 1, 5–10 (2015).

[3] Ameljańczyk A., *Optymalizacja wielokryterialna w problemach sterowania i zarządzania*, Ossolineum, Wrocław, 1984.

[4] Ameljańczyk A., *Multiple optimization*, WAT, Warszawa, 1986.

[5] Ameljańczyk A., "Metoda podziału zbioru obiektów na wielokryterialne klastry jakościowe", *Biuletyn Instytutu Systemów Informatycznych*, Nr 12, 1–7 (2013).

[6] Bouyssou D., Marchant T., "An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories", *EJOR*, 178(1), 217–245 (2007).

[7] Brans J.P., Vincke Ph., "A preference ranking organization method: The PROMETHEE method for Multiple Criteria Decision-Making", *Management Science*, Vol. 31, No. 6, 647–656 (1985).

[8] Rasiowa H., *Wstęp do matematyki współczesnej*, PWN, Warszawa, 2005.

[9] Saaty T.L., "Rank from comparisons and from ratings in the analytic hierarchy/network processes", *EJOR*, 168(2), 557–570 (2006).

[10] Seo F., Sakawa M., *Multiple Criteria Decision Analysis in Regional Planning*, D. Reidel-Kluwer, Dordrech–Boston–Lancaster–Tokyo, 1988.

[11] Tran Quang Ch., *Procedures multi-criteria clustering data*, praca magisterska, WAT, Warszawa, 2015.

[12] Yu P.L., Leitmann G., "Compromise solutions, domination structures and Salukwadze's solution", *JOTA*, Vol. 13, 14–21 (1974).

[13] Yu P.L., Leitmann G., "Nondominated decision and cone convexity in dynamic multicriteria decision problems", *JOTA* Vol. 14, 195–203 (1974).

[14] http://www.scimagojr.com/journalrank.php, SCIMAGO Scientific Journal Rankings, (2015.12.12).

Leksykograficzno-binarna implementacja rekurencyjnego filtra Pareto w procedurach kategoryzacji

A. AMELJAŃCZYK, Ch. TRAN QUANG

W pracy przedstawiono możliwość wykorzystania Rekurencyjnego Filtra Pareto (RPF) w procedurach kategoryzacji obiektów (danych). Przedstawiono nową implementację algorytmu RPF, wykorzystującą leksykograficzne sortowanie obiektów i binarne poszukiwanie elementów optymalnych w sensie Pareto (LBS). Funkcjonowanie algorytmu zilustrowano przykładem z obszaru kategoryzacji czasopism naukowych zawartych w Bazie Scimago Scientific Journals.

Słowa kluczowe: filtr Pareto, klasteryzacja danych, ranking wielokryterialny, kategoryzacja obiektów, rekurencyjny filtr Pareto.