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 \( Y \subset R^N \) – 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:

\[
Y^R_N = \left\{ y \in Y \mid \text{does not exists } \ z \in Y - \{y\}, \text{ such } (z,y) \in R \right\}
\]  
(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](image)

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 \(N\). Each lexicographic relation \(L\) leads to the linear ordering of set \(Y \subset R^N\) [1, 8]. Relation \(L\) can be defined as follows:

\[
L = \left\{ (y, z) \in Y \times Y, \exists k \in N, \text{ that } y_k > z_k \text{ and } y_l = z_l, l < k \right\}
\]  
(3)

Analogical for all other permutations of set of objectives numbers \(N\).
2. Recurrent Pareto filter (RPF)

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

$$r(Y) = \left(Y^R_N(k), k = 1, 2, \ldots, K\right)$$  \hspace{1cm} (4)

where:

$$Y^R_N(k) = \left(Y - \bigcup_{m=0}^{m=k-1} Y^R_N(m)\right)_N.$$ \hspace{1cm} (5)

Figure 2 shows the RPF scheme.

![Fig. 2. Recurrent Pareto Filter (RPF)](image)

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

![Fig. 3. Flowchart of the RPF](image)

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 $\mathcal{R}$ \cite{3,4}:

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

b) $Y^E_N \subset Y^L_N$ (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 $\succ$

$$y^i \succ y^j \iff y^i_n \geq y^j_n, \; n = 1, 2, \ldots, N$$  \hspace{1cm} (6)

**Example 1** ($M=10, N=5$)

Assume that, as result of initial sorting process original set $Y$ is stored in the list $L = \{y^1, y^2, y^3, \ldots, y^M\}$ where:

$$y^i = \{y^i_1, y^i_2, \ldots, y^i_N\}, \; i = 1, \ldots, M$$  \hspace{1cm} (7)

In general, we can use arbitrary preference but for simplicity, in below example the input set was sorted using preference order $(1, 2, 3, \ldots, N)$. 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: \( 0 < k \leq N \) such that \( y^i_k > y^j_k \) and \( y^m_m = y^m_m \) 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 \( y^i \) is dominated by another element \( y^j \) (in Pareto sense), we can sure that \( y^i \) is better than \( y^j \) in all objectives (in Pareto sense). But, if an element \( y^j \) is dominated by another element \( y^i \) (in Pareto sense), we can sure that \( y^i \) is better than \( y^j \) in all objectives (in Pareto sense) [3, 4].

From definition of ranking task, element \( y^h \) has rank \( r^i = k \), if and only if exists other element \( y^{h-i} \) which has rank \( r^{h-i} = k - 1 \), so that \( y^{h-i} \succeq y^h \), which means that exists \( y^{h-i} \succeq y^{h-k} \) and \( r^{h-k} = k - 2 \) and so on. In other words, exists a dominance chain:

\[
y^h \succeq y^{h_1} \succeq ... \succeq y^{h_{k}} \succeq y^i
\]

where \( r^h = 1, r^{h_1} = 2, ..., r^{h_k} = k \).

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 \( R_N^Y(k) \), which contains all elements with rank \( k \) is said that dominating element \( y^j \) \((Y_N^Y(k) \succeq y^j)\) if and only if exists \( y^j \in Y_N^Y(k) \), so that \( y^j \succeq y^i \). From this definition, we can see that, if an element \( y^i \) isn’t dominated by cluster \( Y_N^Y(k) \) so it also can’t be dominated by another cluster \( Y_N^Y(k') \) with \( k' > k \). Because in other way, if \( y^j \) is dominated by \( Y_N^Y(k') \) hence exists at least one dominance chain:

\[
y^h \succeq y^{h_1} \succeq ... \succeq y^{h_k} \succeq y^{h_{k-1}} \succeq ... \succeq y^{h_1} \succeq y^i
\]

In another word, cluster \( Y_N^Y(k) \) 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 \( Y_N^Y(k) \succeq y^j \)

where \( k = 1, \max(r^1, r^2, ..., r^{j-1}) \) 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:

<table>
<thead>
<tr>
<th>( j )</th>
<th>( k )</th>
<th>( r^j )</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>does not exist ( k )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>does not exist ( k )</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>( y^1 \succeq y^3 )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>( y^3 \succeq y^4 )</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>( y^3 \succeq y^5 )</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>( y^5 \succeq y^6 )</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>( y^5 \succeq y^7 )</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>( y^3 \succeq y^8 )</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>( y^7 \succeq y^9 )</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>( y^9 \succeq y^{10} )</td>
<td></td>
</tr>
</tbody>
</table>

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

\[
Y_N^R(1) = \{y^1, y^2\} - (\text{gold category})
\]

\[
Y_N^R(2) = \{y^3\} - (\text{silver category})
\]

\[
Y_N^R(3) = \{y^4, y^5, y^6, y^8\} - (\text{bronze category})
\]

\[
Y_N^R(4) = \{y^7\}, etc.
\]

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

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. Based 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.

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

<table>
<thead>
<tr>
<th>Rank</th>
<th>H index</th>
<th>Total Refs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>108</td>
<td>2888</td>
</tr>
<tr>
<td>2</td>
<td>233</td>
<td>9315</td>
</tr>
<tr>
<td>3</td>
<td>244</td>
<td>4220</td>
</tr>
<tr>
<td>4</td>
<td>302</td>
<td>8882</td>
</tr>
<tr>
<td>5</td>
<td>246</td>
<td>8009</td>
</tr>
<tr>
<td>6</td>
<td>585</td>
<td>30034</td>
</tr>
<tr>
<td>7</td>
<td>171</td>
<td>1620</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>M</th>
<th>N</th>
<th>RPF(BF)</th>
<th>LBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>7559</td>
<td>2</td>
<td>0.745</td>
<td>0.141</td>
</tr>
<tr>
<td>4105</td>
<td>3</td>
<td>0.290</td>
<td>0.050</td>
</tr>
<tr>
<td>6441</td>
<td>4</td>
<td>1.162</td>
<td>0.103</td>
</tr>
<tr>
<td>8312</td>
<td>5</td>
<td>1.283</td>
<td>0.197</td>
</tr>
<tr>
<td>2805</td>
<td>6</td>
<td>0.254</td>
<td>0.055</td>
</tr>
</tbody>
</table>

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


Leksykograficzno-binarna implementacja rekurencyjnego filtra Pareto w procedurach kategoryzacji

A. AMELJAŃCZYK, Ch. TRAN QUANG


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