JMcdM: A Julia package for multiple-criteria decision-making tools

Mehmet Hakan Satman, Bahadır Fatih Yıldırım, and Ersagun Kuruca

1 Department of Econometrics, Istanbul University, Istanbul, Turkey 2 Department of Transportation and Logistics, Istanbul University, Istanbul, Turkey 3 Independent researcher

Summary

JMcdM is a Julia package that implements some leading multiple-criteria decision-making tools for both researchers and developers. By having a REPL tool, Julia is well suited for researchers to perform their analysis using different methods and comparing their results. JMcdM also provides the necessary infrastructure, utility functions, and a standardized API for implementing recently published methods. The package brings MCDM (Multiple-Criteria Decision-Making) tools to a relatively new language such as Julia with its significant performance promises. Besides Julia being a new language, the methods developed in the package are designed to be familiar to users who previously used the R and Python languages. This paper presents the basics of the design, example usage, and code snippets.

Introduction

The one-dimensional array \( a \) is in ascending order if and only if \( a_i \leq a_{i+1} \) where \( i = 1, 2, \ldots, n - 1 \), and \( n \) is the length of the array. In other terms, the process of ordering numbers requires the logical \( \leq \) operator to be perfectly defined. Since the operator \( \leq \) is not defined for any set of points in higher dimensions, \( \mathbb{R}^p \) for \( p \geq 2 \), there is not a unique ordering of points. In the multi-dimensional case, the binary domination operator \( \succ \) applied on points \( a \) and \( b \), \( a \succ b \), is true if each item in \( a \) is not worse than the corresponding item in \( b \) and at least one item is better than the corresponding item in \( b \) (Deb et al., 2002). On the other hand, the more relaxed operator \( \succeq \) returns true if each item in \( a \) is as good as the corresponding item in \( b \) (Greco et al., 2016). Several outranking methods in MCDM (Multiple-Criteria Decision Making) define a unique ranking mechanism to select the best alternative among others.

Suppose a decision process has \( n \) alternatives and \( m \) criteria that are either to be maximized or minimized. Each single criterion has a weight \( 0 \leq w_i \leq 1 \) where \( \sum_i w_i = 1 \) and is represented by a function \( f_i \) which is either maximum or minimum. \( g_j(.) \) is an evolution function and it is taken as \( g_j(x) = x \) in many methods. A multiple criteria decision problem can be represented using the decision table shown in Table 1 without loss of generality. When \( A_1, A_2, \ldots, A_n \) are alternatives and \( C_1, C_2, \ldots, C_m \) are different situations of a single criterion then the decision problem is said to be a single criterion decision problem. If \( A_i \) and \( C_j \) are strategies of two game players then \( g_j(A_i) \) is the gain of the row player when she selects the strategy \( i \) and the column player selects the strategy \( C_j \).
Table 1: Decision table

<table>
<thead>
<tr>
<th>Criteria</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>...</th>
<th>$C_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weights</td>
<td>$w_1$</td>
<td>$w_2$</td>
<td>...</td>
<td>$w_m$</td>
</tr>
<tr>
<td>Functions</td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>...</td>
<td>$f_m$</td>
</tr>
<tr>
<td>$A_1$</td>
<td>$g_1(A_1)$</td>
<td>$g_2(A_1)$</td>
<td>...</td>
<td>$g_m(S_A)$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$g_1(A_2)$</td>
<td>$g_2(A_2)$</td>
<td>...</td>
<td>$g_m(A_2)$</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_n$</td>
<td>$g_1(A_n)$</td>
<td>$g_2(A_n)$</td>
<td>...</td>
<td>$g_m(A_n)$</td>
</tr>
</tbody>
</table>

State of the field

Multiple-criteria decision-making (MCDM) tools provide several algorithms for ordering or selecting alternatives and/or determining the weights when there is uncertainty. Although some algorithms are suitable for hand calculations, computer software is often required. While some previous applications only focused on a single method, some applications appear to include multiple methods. PyTOPS is a Python tool for TOPSIS (Yadav et al., 2019). Super Decisions is a software package that is mainly focused on AHP (Analytic Hierarchy Process) and ANP (Analytic Network Process) (Adams & Saaty, 2003). Visual Promethee implements the Promethee method on Windows platforms (Mareschal & Smet, 2009). M-BA CBETH is another commercial software product that implements MACBETH with an easy to use GUI (Bana e Costa et al., 2011). Sanna is a standard MS Excel add-in application that supports several basic methods for multi-criteria evaluation of alternatives (WSA, TOPSIS, ELECTRE I and III, PROMETHEE I and II, MAPPAC and ORESTE) (Jablonsky, 2014). The DEAFrontier software requires an Excel add-in that can solve up to 50 DMUs with unlimited number of inputs and outputs (subject to the capacity of the standard MS Excel Solver) (Zhu, 2014).

Statement of need

While the applications mentioned above are lacking in features such as the number of methods included, being programmable, being free, and the results being comparable by the researcher, JMcdM clearly differs as it has all of these features. JMcdM is designed to provide a developer-friendly library for solving multiple-criteria decision problems in Julia (Bezanson et al., 2017). Since Julia is a dynamic language, it is also useful for researchers who are familiar with REPL (Read-Eval-Print-Loop) environments. The package includes multi-criteria decision methods as well as a game solver for zero-sum games, and methods for single criterion methods.

The package implements methods for AHP (Saaty, 1977), ARAS (Edmunds Kazimieras Zavadskas & Turskis, 2010), COCOSO (Yazdani et al., 2019), CODAS (Keshavarz Ghoreabae et al., 2016), COPRAS (Edmunds Kazimieras Zavadskas et al., 1994), CRITIC (Diakoulaki et al., 1995), DEMATEL (Gabus & Fontela, 1972), EDAS (Ghorabae et al., 2015), ELECTRE (Roy, 1968), Entropy (Shannon, 1948), GRA (Ju-Long, 1982), MABAC (Pamučar & Ćirović, 2015), MAIRCA (Pamučar et al., 2014), MARCOS (Stević et al., 2020), MOORA (Brauers & Zavadskas, 2006), NDS (Deb et al., 2002), PROMETHEE (Brans & Vincke, 1985), SAW (Churchman & Ackoff, 1954; Triantaphyllou & Mann, 1989), TOPSIS (Hwang & Yoon, 1981), VIKOR (Opricovic, 1998; Opricovic & Tzeng, 2002), WSPAS (E. K. Zavadskas et al., 2012), and WPM (Triantaphyllou & Mann, 1989) for multiple-criteria tools. This list of selected methods includes both classical (TOPSIS, ELECTRE, PROMETHEE, etc.) and modern (COCOSO, MABAC, MARCOS, etc.) tools of the relevant literature.

The package also performs Data Envelopment Analysis (DEA) (Charnes et al., 1978) and includes a method for solving zero-sum games. Although these methods may seem different from the methods mentioned above, they are basically members of the same method family and solve similar problems. DEA differs from the above methods in that it is not an outranking method but compares efficiencies of decision units. Solving zero-sum games is also a multi-criteria decision-making problem, but this time, unlike outranking methods, both the rows and columns of the decision matrix show alternative strategies.

The full set of other tools and utility functions are listed and documented in the source code as well as in the online documentation.

**Installation and basic usage**

JMcDM can be downloaded and installed using the Julia package manager by typing

```
julia> using Pkg
julia> Pkg.add("JMcDM")
```

and can be loaded before using any functions by typing

```
julia> using JMcDM
```

in Julia REPL.

Suppose a decision problem is given in the table below.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Age</th>
<th>Size</th>
<th>Price</th>
<th>Distance</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weights</td>
<td>0.35</td>
<td>0.15</td>
<td>0.25</td>
<td>0.20</td>
<td>0.05</td>
</tr>
<tr>
<td>Functions</td>
<td>min</td>
<td>max</td>
<td>min</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>$A_1$</td>
<td>6</td>
<td>140</td>
<td>150000</td>
<td>950</td>
<td>1500</td>
</tr>
<tr>
<td>$A_2$</td>
<td>4</td>
<td>90</td>
<td>100000</td>
<td>1500</td>
<td>2000</td>
</tr>
<tr>
<td>$A_3$</td>
<td>12</td>
<td>140</td>
<td>75000</td>
<td>550</td>
<td>1100</td>
</tr>
</tbody>
</table>

In this sample problem, a decision maker is subject to select an apartment by considering the age of the building, size (in $m^2$), price (in $\), distance to city centre (in $m$), and nearby population. The data can be entered as a two-dimensional array (matrix) or as a DataFrame object:

```
julia> using JMcDM
julia> df = DataFrame(
    :age => [6.0, 4, 12],
    :size => [140.0, 90, 140],
    :price => [150000.0, 100000, 75000],
    :distance => [950.0, 1500, 550],
    :population => [1500.0, 2000, 1100]);
```

The weight vector $w$, vector of directions $fns$, and topsis() function call can be performed using the Julia REPL.

```
julia> w = [0.35, 0.15, 0.25, 0.20, 0.05];
julia> fns = makeminmax([minimum, maximum, minimum, minimum, minimum, maximum]);
```
julia> result = topsis(df, w, fns);

julia> result.scores
3-element Array{Float64,1}:
0.5854753145549456
0.6517997936899308
0.41850223305822903

julia> result.bestIndex
2

In the output above, it is shown that the alternative $A_2$ has a score of 0.65179 and it is selected as the best. The same analysis can be performed using saw() for the method of Simple Additive Weighting

julia> result = saw(df, w, fns);

julia> result.bestIndex
2

as well as using wpm for the method of Weighted Product Method

julia> result = wpm(df, w, fns);

julia> result.bestIndex
2

For any method, ?methodname shows the documentation as in the same way in other Julia packages.

References


Gabus, A., & Fontela, E. (1972). World problems, an invitation to further thought within the framework of DEMATEL. In Battelle Geneva Research Center, Geneva, Switzerland (pp. 1–8).


