

Scikit-Criteria: Colección de métodos de análisis multi-criterio integrado al stack científico de Python

Juan B Cabral^{1,2}, Nadia Ayelen Luczywo^{3,4}, José Luis Zanazzi³

¹ IATE - Instituto De Astronomía Teórica Y Experimental - Observatorio Astronómico Córdoba, UNC - CONICET Laprida 854, X5000BGR, Córdoba, Argentina Email: jbc.develop@gmail.com

² Facultad de Ciencias Exactas, Ingeniería y Agrimensura - UNR, Pellegrini 250 - S2000BTP Rosario, Argentina

³ LIMI - Laboratorio de Ingeniería y Mantenimiento Industrial Facultad de Ciencias Exactas Físicas y Naturales, UNC Av.Velez Sarsfield 1611, X5016GCA Córdoba, Argentina

⁴ Instituto de Estadística y Demografía Facultad de Ciencias Económicas, UNC Avda. Valparaíso s/n - Ciudad Universitaria

Resumen Se realiza un análisis de las herramientas de software existentes en el análisis multi-criterio/multiatributo con la evaluación de las implicancias relacionadas con el funcionamiento de cada paquete como un estanco difícil de integrar e incapaz de aceptar datos en formatos diferentes. También se evalúa la ventaja de PYTHON como lenguaje de programación para la implementación de una plataforma de cómputo científico para el soporte de métodos multi-criterio. Luego, se presenta SCIKIT-CRITERIA una librería que implementa inicialmente un set de métodos multiatributo y funcionalidades para facilitar el cómputo de estos. Finalmente, se presenta un caso de aplicación y se discute una futura implementación de una capa de orientación a objetos para facilitar el uso de la librería.

Keywords: Decisión multicriterio discreta, Software, Python, Scikit-Criteria

1. Introducción

El presente trabajo expone una nueva herramienta *open-source* desarrollada en lenguaje de programación PYTHON⁵ para la aplicación de métodos de Decisión Multicriterio Discreta (DMD)⁶.

Los procesos de toma de decisiones han sido un objeto de estudio ante el desafío de resolver situaciones complejas. En efecto, surgen métodos de asistencia que facilitan la toma de decisiones. Puntualmente, bajo la denominación Decisión Multicriterio Discreta, proliferan métodos de soporte matemático y técnico aplicables para abordar el problema de evaluar un conjunto de alternativas en términos de una serie de criterios [27]. De hecho, el avance en la línea

⁵ <http://python.org>

⁶ El paquete llamado SCIKIT-CRITERIA esta disponible en: <http://scikit-criteria.org/>

de investigación de DMD ha posibilitado la construcción de paquetes enlatados de *software* que soportan estos métodos y colaboran con los decisores en las aplicaciones.

Sin embargo, a pesar de los extensos progresos en programación aún subsisten dificultades no salvadas en los desarrollos propuestos. Entre éstas, pueden citarse las relacionadas con la limitación en el volumen de datos sobre el cual se puede operar en algunos paquetes, con el soporte de un único método o un pequeño conjunto de métodos similares, con la consecuente imposibilidad de migración de datos a otros y con la presencia de casos de código cerrado que requieren licencias pagas[23].

Las implicancias de estas limitaciones repercuten directamente en las aplicaciones y en la investigación teórica, porque acotan el margen de acción en el proceso de estructuración y reestructuración del problema y porque dificultan el análisis teórico comparativo sobre los distintos métodos.

Paralelamente, el lenguaje de programación PYTHON ha llegado a ser uno de los más difundidos en el ámbito científico y ha sido la base para los desarrollos de robustos paquetes de cómputo numérico, tales como NUMPY[24] y SCIPY⁷. Sobre éstos paquetes, diferentes áreas de la ciencia han desarrollado soluciones que generalizan problemas a partir de la definición de algoritmos y estructuras de datos particulares, ejemplos son: SCIKIT-BIO⁸ para biología, ASTROPY [20] para astronomía, SCIKIT-LEARN [19] para aprendizaje automático, SYMPY [12] para matemática simbólica, PYMC [18] para simulación de Montecarlo y cadenas de Markov. Además, a diferencia de R [17] y MATLAB⁹ (dos de los lenguajes más populares en el ámbito científico), PYTHON se destaca como un poderoso lenguaje de propósito general (se puede construir cualquier tipo de aplicación con él) que dispone múltiples alternativas para la integración con variadas fuentes de datos (SQL, CSV, JSON, XML, entre otros) y para la creación de aplicaciones *web* y de escritorio. Consecuentemente, posibilita una mayor flexibilidad en la construcción de aplicaciones para usuarios finales alrededor de cómputo científico.

A raíz de las ventajas del lenguaje PYTHON, y en la búsqueda de superación de las limitaciones que adolecen los paquetes existentes en materia de DMD, esta propuesta se construye sobre este lenguaje de programación y presenta una aplicación para cómputo de métodos DMD.

Finalmente, el objetivo central de este trabajo está referido a disponer de métodos multicriterio en un lenguaje multipropósito de alto nivel de manera que sea posible la integración de la DMD a arquitecturas mayores a las que están disponibles paquetes propietarios o MICROSOFT-EXCEL¹⁰, se facilite el aprovechamiento de las herramientas científicas disponibles en lenguaje de programación de alto nivel, se realicen implementaciones eficientes en complejidad espacial y temporal de los DMD, y pueda lograrse la migración de datos de un método a otro con la menor conversión de datos a través de interfaces programáticas uniformes.

⁷ <http://scipy.org>

⁸ <http://scikit-bio.org/>

⁹ <https://www.mathworks.com>

¹⁰ <https://products.office.com/es/excel>

2. Antecedentes

Dentro de la DMD, existen variadas metodologías que intentan dar un apoyo sobre elecciones con cierta dificultad. Los métodos DMD se han orientado indistintamente a: seleccionar la mejor alternativa, a elegir las alternativas buenas y rechazar las malas, o a realizar un ordenamiento de las alternativas[9]. Así, ante esta gama de posibilidades metodológicas y de resultados, aparecen intentos en designar cual es el método más adecuado para el abordaje de un cierto problema. En este sentido, Guitouni y Martel [10] dan el puntapié inicial y proponen siete lineamientos tentativos para ayudar a elegir los métodos DMD más adecuados aplicables a un caso en particular.

La riqueza en metodologías se proyecta en variedad de software que soportan los métodos multi-criterio, por ejemplo, Vassilev y otros[25] hacen un recuento de esta multiplicidad de software disponibles. Con este objetivo listan software de propósito general (aplicables a cualquier problema) como: VIDMA (Korhonen), EXPERT CHOICE (Saaty), HIVIEW (Peterson), PROMCAL y GAIA (Brans y Mareschal), ELECTRE III-IV (Roy), MACBETH (Bana e Costa y Vansnick), VIP (Dias y Climaco), PREFDI (Zopounidis y Doumpos), DECISION LAB (Brans and Mareschal), WEB-HIPRE (Mustajoki y Hamalainen), IRIS (Dias y Mousseau), IDS (Xu y Yang), MULTICHOICE (Vassilev y otros), KNOWCUBE (Hanne and Trinkaus), y sistemas de software orientados a problemas como el SISTEMA FINCLAS para problemas de clasificación financiera (Zopounidis and Doumpos), el SISTEMA DEL INVERSOR PARA LA SELECCIÓN Y COMPOSICIÓN DEL PORTAFOLIO (Zopounidis and Doumpos), el SISTEMA DE DECISIÓN AGLAND para la propiedad agrícola (Parsons), el SISTEMA DESYRE para la rehabilitación de los sitios contaminados (Carlson y otros), MULTCSYNC para la incorporación de múltiples criterios de planificación de la conservación (Moffett y otros). También, Buede [6], hace una revisión comparativa sobre tres paquetes que corren el método AHP: CRITERIUM, EXPERT CHOICE y HIPRE 3+. Pérez, Rodríguez y Rojo Alboreca [22] presentan el SOFTWARE MPC 2.0 ©, para la aplicación del método AHP. Zavadskas, Ustinovichius, y Peldschus [26] hacen una revisión de ELECTRE III SOFTWARE Y ELECTRE IV SOFTWARE para el método ELECTRE, PREFCALC SOFTWARE para el método UTA, MAPPAC SOFTWARE para MAPPAC, *CARTESIA software* para Cartesia, PROMCALC SOFTWARE para el método promethee. Tervonen [23] presenta una herramienta *open source* JSMAA para trabajar con el método SMAA, (*Stochastic Multicriteria Acceptability Analysis*) que se puede utilizar para buscar alternativas de compromiso en ambientes de elevada incertidumbre a través de la utilización de distribuciones de probabilidad multivariadas.

De la mano de estas múltiples propuestas, aparecen intentos de selección de los paquetes de *software* más adecuados. Li y Thomas[14], proponen un enfoque metodológico para la selección de *software* DMD que contemple correspondencia entre la estructuración de la situación de toma de decisiones, y la estructuración de preferencias de las herramientas. Mustajoki y Marttunen [16], avanzan en una propuesta similar de comparación de programas informáticos para la decisión multi-criterio en términos de las características que proporcionan a partir de la metodología IMPERIA (Evaluación ambiental para la mejora mediante la

adopción de buenas prácticas y herramientas de análisis de decisión multicriterio).

Por lo expuesto, puede apreciarse que la selección del método y *software* adecuado, ha sido un tópico de interés en la literatura. Como en todo problema de interés, los avances en delinear mejores métodos, *software* o herramientas para la comparación de paquetes informáticos pueden ayudar en los procesos de toma de decisiones y hacerlos más flexibles.

Además, cabe destacar el papel preponderante que juega la homogeneidad en los desarrollos de *software*, ya que posibilita la mayor sinergia en los esfuerzos comunitarios, la disminución de duplicaciones, el refuerzo de la investigación de largo plazo y la reproductibilidad de los resultados [21]. Este argumento es reforzado por Li y Thomas[14] quienes reconocen la necesidad de integrar el proceso de selección del método y *software* de aplicación en un sistema de soporte de decisiones. Esta ventaja se evidencia en diferentes paquetes existentes que aunaron esfuerzos comunitarios en un único proyecto que representa todos los problemas de una disciplina en funciones homogéneas. Algunos de ellos son SCIKIT-LEARN, WEKA (ambos para aprendizaje automático para las comunidades de PYTHON y JAVA respectivamente), R (para la comunidad estadística), o el caso de un programa de usuario final como MICROSOFT-EXCEL que es una herramienta fundamental para la gestión empresarial en el área operativa de las empresas.

3. Scikit-Criteria

El proyecto SCIKIT-CRITERIA brinda una capa de abstracción sobre las funcionalidades más comunes de DMD integrado al ecosistema científico del lenguaje de programación PYTHON; de modo que intenta posibilitar que las funciones expuestas sean lo más parecidas posibles entre ellas para que sean fácilmente reemplazables unas con otras. Además, se trata de generar módulos de funcionalidad común para que se pueda utilizar fácilmente en la creación de métodos de terceros.

Las funcionalidades de *Scikit-Criteria* están ordenadas dentro del paquete `skcriteria` de la siguiente forma:

- `skcriteria.common`: Funcionalidades comunes independientes de un DMD en particular
 - `skcriteria.common.norm` funciones de normalización de arreglos multidimensionales.
 - `skcriteria.common.rank` funcionalidades para generar rankings desde valores numéricos y evaluar estos rankings a través de métodos como dominancia, igualdad, dominancia absoluta e índice correlación de Kendall [4]
 - `skcriteria.common.util` funcionalidades comunes.
- Métodos de DMD
 - `skcriteria.ahp` implementa la funcionalidad del método AHP[11] con el índice de consistencia, y algunas funciones de soporte para facilitar la generación de matrices a partir de solo la mitad inferior de las mismas

- `skcriteria.electre` actualmente solo implementa el método ELECTRE-1[8] y funciones auxiliares para la creación de matrices de concordancia, discordancia así como el valor de los umbrales p y q respectivamente.
- `skcriteria.moora` Definiciones de funciones de toda la familia MOORA[13][5] como son: `skcriteria.moora.ratio` MOORA, `skcriteria.moora.refpoint` MOORA con punto de referencia, `skcriteria.moora.fmf` forma multiplicativa completa, y `skcriteria.moora.multimora` Multi-MOORA
- `skcriteria.topsis` soporta el método TOPSIS[3].
- `skcriteria.wprod` despliega el método de producto ponderado[1].
- `skcriteria.wsum` permite computar el método de suma ponderada[1].

Para garantizar la calidad del global del proyecto se utilizó la técnica de creación dirigida por pruebas [2] donde se reprodujo al menos un artículo de cada método implementado y se buscó una cobertura de código ejecutado por los test no menor al 90% [15].

4. Caso de Aplicación

Para demostrar el uso de SCIKIT-CRITERIA, reproduciremos el experimento perteneciente al trabajo “*Heating Losses in a Building by Applying the MULTI-MOORA*” de Kracka et al 2010 [13] utilizando nuestra librería sobre el lenguaje Python.

```

1 # cargamos el paquete de MOORA
2 >>> from skcriteria import mooora
3
4 # definimos la matriz de preferencias donde cada fila es una
5 # alternativa y cada columna es un criterio.
6 >>> mtx = [[33.95, 23.78, 11.45, 39.97, 29.44, 167.10, 3.852],
7           [38.9, 4.17, 6.32, 0.01, 4.29, 132.52, 25.184],
8           [37.59, 9.36, 8.23, 4.35, 10.22, 136.71, 10.845],
9           [30.44, 37.59, 13.91, 74.08, 45.10, 198.34, 2.186],
10          [36.21, 14.79, 9.17, 17.77, 17.06, 148.3, 6.610],
11          [37.8, 8.55, 7.97, 2.35, 9.25, 134.83, 11.935]]
12
13 # vector de criterios: tiene la misma longitud que cantidad de
14 # columnas tiene la matriz de preferencias. util.MIN representa
15 # un criterio de minimo y util.MAX uno de maximo
16 >>> criteria = [util.MIN, util.MIN, util.MIN,
17               util.MIN, util.MAX, util.MIN, util.MAX]
18
19 # ahora aplicamos Multi-MOORA y obtenemos los valores
20 # del ranking de las alternativas
21 >>> rank = mooora.multimooora(mtx, criteria)
22 >>> print rank[0]
23 [5 1 3 6 4 2]
```

El resultado final nos informa en qué lugar del ranking esta cada una de las alternativas de la matriz de preferencias; por ejemplo la posición 2 del vector de resultados vinculada a la alternativa 2 posee el primer lugar del ranking, y la posición 4 se ubica en la sexta posición.

También, es posible reemplazar la llamada a *Multimoora* por la llamada a la función `topsis.topsis(mtx, criteria)` luego de importar `skcriteria.topsis` para utilizar TOPSIS en lugar de MOORA. De esta forma, se puede apreciar la facilidad de migración y adaptación del trabajo de un método a otro.

5. Conclusión y trabajo a futuro

Se ha diseñado un proyecto de *software* que es capaz de integrarse fácilmente a un análisis de datos sobre la plataforma PYTHON. Asimismo, se provee una serie de herramientas comunes (paquete `scikit.common`) para la extensión de las funcionalidades preexistentes de manera de disminuir el esfuerzo en la realización de análisis multi-criterio. A futuro se pretende trabajar en la creación de una capa de abstracción orientada a objetos como la existente en SCIKIT-LEARN para facilitar el encapsulamiento de los datos de los métodos DMD [7].

Referencias

- [1] Barba-Romero, S., Pomerol, J.C.: Decisiones multicriterio: fundamentos teóricos y utilización práctica. Universidad de Alcalá, Alcalá de Henares, Madrid (1997)
- [2] Beck, K.: Test-driven development: by example. Addison-Wesley Professional (2003)
- [3] Bhutia, P.W., Phipon, R.: Application of ahp and topsis method for supplier selection problem. *IOSR Journal of Engineering (IOSRJEN)* Volume **2** (2012) 43–50
- [4] Brauers, W.K.M., Zavadskas, E.K.: Robustness of MULTIMOORA: a method for multi-objective optimization. *Informatica* **23**(1) (2012) 1–25
- [5] Brauers, W.K.M., Zavadskas, E.K., Peldschus, F., Turskis, Z.: Multi-objective decision-making for road design. *Transport* **23**(3) (2008) 183–193
- [6] Buede, D.M.: Software review. Three packages for AHP: Criterium, expert choice and HIPRE 3+. *Journal of Multi-Criteria Decision Analysis* **1**(2) (1992) 119–121
- [7] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., et al.: API design for machine learning software: experiences from the scikit-learn project, European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases, Sep 2013, Prague, Czech Republic (2013)
- [8] Cebrián, L.I.G., Porcar, A.M.: Localización empresarial en Aragón: Una aplicación empírica de la ayuda a la decisión multicriterio tipo ELECTRE I y III. robustez de los resultados obtenidos. *Revista de Métodos Cuantitativos para la Economía y la Empresa* (7) (2009) 31–56
- [9] Giard, V.E., Roy, B.: *Méthodologie multicritère d'aide á la décision*. Editions Economica (1985)
- [10] Guitouni, A., Martel, J.M.: Tentative guidelines to help choosing an appropriate MCDA method. *European Journal of Operational Research* **109**(2) (1998) 501–521
- [11] Ishizaka, A., Labib, A.: Review of the main developments in the analytic hierarchy process. *Expert systems with applications* **38**(11) (2011) 14336–14345
- [12] Joyner, D., Čertík, O., Meurer, A., Granger, B.E.: Open source computer algebra systems: Sympy. *ACM Communications in Computer Algebra* **45**(3/4) (2012) 225–234
- [13] Kracka, M., Brauers, W.K.M., Zavadskas, E.K.: Ranking heating losses in a building by applying the multimoor. *Engineering Economics* **21**(4) (2010) 352–359
- [14] Li, Y., Thomas, M.A.: A multiple criteria decision analysis (MCDA) software selection framework. In: *System Sciences (HICSS)*, 2014 47th Hawaii International Conference on, IEEE (2014) 1084–1094
- [15] Miller, J.C., Maloney, C.J.: Systematic mistake analysis of digital computer programs. *Communications of the ACM* **6**(2) (1963) 58–63

- [16] Mustajoki, J., Marttunen, M.: Comparison of multi-criteria decision analytical software. Finnish Environment Institute (2013)
- [17] Paradis, E., Claude, J., Strimmer, K.: Ape: analyses of phylogenetics and evolution in r language. *Bioinformatics* **20**(2) (2004) 289–290
- [18] Patil, A., Huard, D., Fonnesbeck, C.J.: Pymc: Bayesian stochastic modelling in python. *Journal of statistical software* **35**(4) (2010) 1
- [19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research* **12** (2011) 2825–2830
- [20] Robitaille, T.P., Tollerud, E.J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A.M., Kerzendorf, W.E., et al.: Astropy: A community python package for astronomy. *Astronomy & Astrophysics* **558** (2013) A33
- [21] Robitaille, T.P., Tollerud, E.J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A.M., Kerzendorf, W.E., others: Astropy: A community Python package for astronomy. *Astronomy & Astrophysics* **558** (2013) A33
- [22] Rodríguez, F.P., Alboreca, A.R.: MPC 2.0\copyright, software para la aplicación del método AHP de toma de decisiones multicriterio. *Recursos Rurais* (7) (2012)
- [23] Tervonen, T.: JSMAA: open source software for SMAA computations. *Int. J. Syst. Sci* **2012** (2012) 1–13
- [24] Van Der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering* **13**(2) (2011) 22–30
- [25] Vassilev, V., Genova, K., Vassileva, M.: A brief survey of multicriteria decision making methods and software systems. *Cybernetics and information technologies* **5**(1) (2005) 3–13
- [26] Zavadskas, E.K., Turskis, Z., Kildienė, S.: State of art surveys of overviews on MCDM/MADM methods. *Technological and economic development of economy* **20**(1) (2014) 165–179
- [27] Zavadskas, E.K., Ustinovichius, L., Peldschus, F.: Development of software for multiple criteria evaluation. *Informatica* **14**(2) (2003) 259–272