TRABAJO FIN DE GRADO

INGENIERÍA INFORMÁTICA

Estudio y Análisis de Metaheurísticas modernas para el problema de Selección de Características

Autor

Miguel García López

Directores

Daniel Molina Cabrera

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, Enero de 2024

Título del proyecto

Subtítulo del proyecto.

Autor
Nombre Apellido1 Apellido2 (alumno)

Directores
Nombre Apellido1 Apellido2 (tutor1)
Nombre Apellido1 Apellido2 (tutor2)

Estudio y Análisis de Metaheurísticas modernas para el problema de Selección de Características

MIGUEL GARCÍA LÓPEZ

Palabras clave: Metaheurística, selección, de características, binario, optimización, población, fitness, aprendizaje automático

Resumen

En el ámbito del Machine Learning, algunos algoritmos, como KNN o SVM, muestran excelentes resultados, pero muchas veces estos requieren de un procesamiento previo para identificar las características más relevantes. Ya sea por escoger un problema amplio y complejo, por falta de conocimiento o por falta de contexto a la hora de recolectar los datos, es posible que se llegue a aglomerar información de más, dando lugar a conjuntos de datos inmensos. Incluso aquellos algoritmos que internamente realizan esta identificación de características pueden beneficiarse de un procesamiento adicional. Este preprocesamiento, conocido como selección de características (features selection), se considera un problema complejo de optimización combinatoria.

Las metaheurísticas son algoritmos diseñados para resolver problemas de optimización complejos cuando los recursos son limitados. Aunque inicialmente se desarrollaron para abordar principalmente problemas combinatorios, en la actualidad se están proponiendo y aplicando cada vez más en problemas que implican variables continuas o reales. En respuesta a la creciente demanda en el campo de la selección de características, se han adaptado versiones especializadas de estas metaheurísticas para abordar este tipo de problemas combinatorios. Sin embargo, a pesar del número creciente de propuestas en este campo, las comparaciones objetivas entre ellas son limitadas. Aunque existen revisiones bibliográficas, muchas de ellas carecen de comparaciones adecuadas debido a la importancia y actualidad del problema de selección de características. Por lo tanto, hay una necesidad de estudios que proporcionen una evaluación comparativa más rigurosa y exhaustiva de las diferentes propuestas en este ámbito.

En este trabajo de carácter científico, se llevará a cabo una revisión bibliográfica de diversas metaheurísticas recientes para abordar el problema de selección de características. Se estudiarán e implementarán aquellas consideradas más prometedoras, con el objetivo de construir un repertorio amplio y variado de propuestas. Posteriormente, se realizará un estudio comparativo exhaustivo utilizando diversos algoritmos de machine learning y conjuntos de datos

representativos. Finalmente, se llevará a cabo un análisis crítico utilizando diversas métricas y valoraciones, como la tasa de acierto y el tiempo de ejecución, entre otras.

Study and Analysis of Modern Metaheuristics for the Feature Selection Problem

MIGUEL GARCÍA LÓPEZ

Keywords: Metaheuristic, feature selection, binary, optimization, population, fitness, machine learning

Abstract

In the field of Machine Learning, some algorithms like KNN or SVM often yield excellent results, but they often require preprocessing to identify the most relevant features. Whether due to tackling a broad and complex problem, lack of domain knowledge, or collecting data without proper context, it's possible to gather excessive information, resulting in vast datasets. Even algorithms internally performing feature identification can benefit from additional processing. This preprocessing, known as feature selection, is considered a complex combinatorial optimization problem.

Metaheuristics are algorithms designed to solve complex optimization problems when resources are limited. Initially developed mainly for combinatorial problems, they are increasingly proposed and applied to problems involving continuous or real variables. In response to the growing demand in feature selection, specialized versions of these metaheuristics have been adapted to tackle combinatorial problems. However, despite the increasing number of proposals in this field, objective comparisons between them are limited. Although there are literature reviews, many lack adequate comparisons due to the importance and timeliness of the feature selection problem. Therefore, there is a need for studies providing a more rigorous and exhaustive comparative evaluation of different proposals in this area.

In this scientific work, a literature review of various recent metaheuristics for feature selection will be conducted. The most promising ones will be studied and implemented to build a broad and varied repertoire of proposals. Subsequently, a comprehensive comparative study will be conducted using various machine learning algorithms and representative datasets. Finally, a critical analysis will be carried out using various metrics and assessments, such as accuracy rate and execution time, among others.

Yo, **Miguel García López**, alumno de la titulación Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77159865E, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Miguel García López

Granada a 29 de Enero de 2024.

D. **Daniel Molina Cabrera**, Profesor del Departamento Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Estudio y Análisis de Metaheurísticas modernas para el problema de Selección de Características***, ha sido realizado bajo su supervisión por **Miguel García López**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 29 de Enero de 2024.

El director:

Daniel Molina Cabrera

Agradecimientos

Poner aquí agradecimientos...

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Definición del problema

El problema de la selección de características se define como el proceso de seleccionar un subconjunto de características relevantes [1]. Una característica es una propiedad individual medible de un fenómeno concreto. Este problema es considerado un problema **NP duro**. La reducción de dimensionalidad, y con ello de características, suele ser necesario a la hora de crear un modelo predictivo por medio del aprendizaje automático ya que muchas de las características dentro de un conjunto de datos pueden no llegar a ser relevantes para solucionar aquellos problemas que se intentan solucionar, ya sea por que no aporta información, porque puede ser agrupada junto a otras tantas en una sola propiedad o incluso porque hay ruido en los datos, lo cual es inevitable [2].

Gracias a la reducción de características es posible mejorar tanto la capacidad de generalización como la precisión del modelo predictivo gracias a la reducción de *ruido*.

Siendo f la función objetivo a predecir, H^n el conjunto de hipótesis o conjunto de modelos de dimensión n posibles, $h^*(x)$ el mejor modelo aprendido y x una variable de entrada. El ruido conocido como ruido estocástico es aquel que atiende a una variación aleatoria que puede surgir de diversos factores, como mediciones imprecisas de señales o la falta de precisión en sensores. Por otro lado, el ruido determinista está directamente relacionado con la complejidad de un modelo. Su presencia aumenta la probabilidad de sobreajuste. El ruido determinista puede explicarse como la parte de la función f que el conjunto de hipótesis H^n no puede capturar, es decir, $f(x) - h^*(x)$. Este tipo de ruido se considera así porque la función (modelo) no es lo suficientemente compleja como para comprender esa parte. Este ruido depende de H^n y permanece constante para un valor dado de x [2].

La reducción de características ayuda a manejar ambos tipos de ruido [1, 2] al simplificar el modelo, lo que puede reducir el impacto del ruido estocástico y disminuir la complejidad del modelo, lo que a su vez puede ayudar a mitigar el ruido determinista al mejorar la capacidad del modelo para capturar las características relevantes y descartar las irrelevantes. Esto puede conducir a una mejor capacidad de generalización y a una reducción del sobreajuste.

Además de la simplificación del modelo, que conduce a una reducción del ruido, la selección de características es un preprocesamiento necesario por varias razones:

1. Interpretabilidad: La presencia de características irrelevantes puede complicar innecesariamente la interpretación y el rendimiento de los modelos de aprendizaje automático [1]. La selección de un subconjunto relevante de características puede simplificar el modelo resultante, haciéndolo más comprensible y fácilmente interpretable.

2. Mejora de la eficiencia computacional: La reducción de la dimensionalidad puede conducir a un ahorro significativo en términos de tiempo y recursos computacionales necesarios para el entrenamiento y la evaluación de modelos. Al eliminar características irrelevantes, se reduce la complejidad del problema y se acelera el proceso de aprendizaje.

3. Evita la maldición de la dimensionalidad: Cuando la dimensionalidad se incrementa en un problema, el volumen del espacio también lo hace, y esto ocurre tan rápido que hace que los datos disponibles se vuelvan dispersos. De forma que para obtener un resultado seguro/fiable, la cantidad de datos necesitados debe verse incrementada de manera exponencial con la dimensionalidad [3]. A menor dimensionalidad (características en el conjunto de datos) menos datos harán falta para obtener un buen modelo.

En este trabajo, se lleva a cabo una investigación y análisis comparativo entre varios métodos de la familia **wrapper** o métodos de envoltura. Existen multitud de estrategias [1] que intentan dar solución a este problema. Los métodos de búsqueda más famosos son los de filtrado (**filter**), los cuáles seleccionan las características más discriminativas según la naturaleza de los datos [1]. Por lo general, estos métodos realizan la selección de características antes de las tareas de clasificación y agrupamiento. Ejemplos de algoritmos de filtrado son *relieF* [4] o F-statistic [5].

Los métodos **wrapper**, en cambio, utilizan el algoritmo de aprendizaje usado postprocesamiento para evaluar las características y seleccionar así las más útiles [1].

Los algoritmos clasificatorios de aprendizaje utilizados en este trabajo son

SVM [6] y *kNN* [7, 8], siendo las máquinas de vectores de soporte un método robusto y eficiente y los vecinos más cercanos un método simple, interpretable y muy eficaz. Se analizará el resultado entre ambos clasificadores entre otros muchos análisis comparativos.

1.1.1. Motivación

El reciente interés del problema de la selección de características en el ámbito de las metaheurísticas en los últimos años es más que evidente. Puede comprobarse como en los últimos años hay una tendencia en la publicación de artículos presentando nuevos métodos metaheurísticos, mejores con respecto a los clásicos o incluso comparativas y análisis entre distintos algoritmos.

Esta crecimiento viene acompañado, sin embargo, de comparaciones que distan de ser objetivas por varios motivos. Entre varios artículos se comparan algoritmos del mismo tipo con soluciones y resultados muy variables entre sí a pesar de mismas configuraciones a la hora de experimentar, artículos sin código referenciado, de forma que sea más fácil interpretar los resultados o duplicarlos, y algoritmos novedosos presentados por su autor o autores que superanmal resto en alguna métrica concreta sin llegar a la rigurosidad adecuada.

Por ello, la motivación principal de este trabajo es la de proveer información no sesgada y todo lo objetiva posible por medio de un análisis comparativo entre los algoritmos optimizatorios metaheurísticos más populares y más citados junto con los algoritmos más robustos y clásicos en el campo de la optimización pseudo estocástica.

1.1.2. Objetivos

Objetivo General:

Realizar una comparación exhaustiva y objetiva de diversas metaheurísticas utilizadas en la selección de características, con el propósito de proporcionar una visión integral y evaluativa sobre su eficacia y aplicabilidad en diferentes contextos de análisis de datos.

Objetivos Específicos:

1. Evaluar el desempeño de las metaheurísticas más relevantes en el ámbito de la selección de características, analizando métricas clave como precisión, estabilidad de las soluciones y eficiencia computacional.

Se emplearán conjuntos de datos de referencia y metodologías de validación cruzada para garantizar la robustez de los resultados.

2. Investigar la transferibilidad de las técnicas diseñadas para dominios continuos y binarios en el contexto de la selección de características. Se analizará si las metaheurísticas efectivas en un dominio son igualmente eficaces cuando se aplican a otro, identificando posibles ventajas y limitaciones de cada enfoque.

3. Identificar las fortalezas y debilidades de cada metaheurística según el tipo de representación de las características. Se realizará un análisis detallado del comportamiento de las técnicas en problemas de selección de características con diferentes tipos de datos, destacando su rendimiento relativo y sus áreas de aplicación más adecuadas.

4. Proporcionar recomendaciones prácticas basadas en los resultados obtenidos, con el objetivo de orientar a practicantes y académicos en la selección y aplicación de metaheurísticas en problemas reales de selección de características.

5. Evaluar los resultados de las metaheurísticas en problemas de selección de característica usando distintos como algoritmos de aprendizaje los métodos *kNN* y *SVM*. Se realizará una comparativa a nivel de eficiencia en tiempo, estabilidad y calidad de los resultados.

1.1.3. Planificación

Un trabajo de fin de grado consta de 12 créditos ECTS, donde se estima que cada crédito debe valer unas 25 horas de trabajo aproximadamente. Teniendo en cuenta estos datos, se calcula que la duración del TFG no debería ser superior a 300 horas. Ha de tenerse en cuenta también que el alumno trabaja 25 horas semanales y debe superar algunas asignaturas además de su projecto final para terminar la carrera. Por lo tanto, el proyecto se planifica con una duración extendida en el tiempo, pero con una carga de trabajo semanal menos intensiva.

Se planifica una duración de 5 meses aproximadamente. Se utilizará un diagrama de Gantt [9] para describir la planificación del proyecto, de manera que se realizarán tareas en un orden cronológico. Sin embargo, se reconoce que algunas tareas probablemente requerirán iteraciones posteriores, ya que es probable que se mejore y perfeccione el proyecto a lo largo de su ciclo de vida.

Las fases del ciclo de vida son:

- **Investigación inicial** Esto incluye investigar sobre conceptos básicos ya aprendidos, en forma de repaso sobre conceptos generales de

aprendizaje automático, tipos de metaheurísticas, tipos de codificación, optimización de funciones, test estadísticos y conceptos básicos, código Python y librerías asociadas, instalación de estas a partir de un entorno virtual, configuración del entorno de trabajo e investigación sobre el problema de selección de características.

- **Diseño del software**: Planificación de la estructura general del código, uso de patrones de diseño que puedan ser de utilidad de cara a al mantenimiento del software a lo largo del desarrollo, concepto de modularización inicial del código (estructura del proyecto), uso de entornos virtuales.

- **Investigación metaheurísticas**: Realización de un estudio más exhaustivo acerca de las metaheurísticas a implementar y sus diferentes versiones binarias. Esto incluye un listado de 12 metaheurísticas, siendo estas:

  - Binary Firefly Algorithm
  - Binary Whale Optimization Algorithm
  - Binary Bat Swarm Optimizer
  - Binary Grey Wolf Optimizer
  - Binary Dragonfly Algorithm
  - Binary Grasshoper Algorithm
  - Binary Cuckoo Search
  - Binary Differential Algorithm
  - Ant Colony Optimization
  - Binary Artificial Bee Colony Optimization
  - Binary Particle Swarn Optimization
  - Genetic Algorithm (binary & real)

  De cada una de ellas se investigará su inspiración, funcionamiento, implementación y versiones binarias, normalmente asociadas al problema de selección de características.

- **Implementación del software**: Una vez claros los requisitos programáticos quedan establecidos, se implementará el software base. Esto incluye código en Python para la generación de gráficas, manejo de datasets en formato *arff*, codificación de los algoritmos metaheurísticos en versión binaria, implementación de función objetivo (*fitness*) y parametrización del programa para distintas pruebas.

- **Pruebas y refactorizado**: En esta etapa se llevarán a cabo pruebas exhaustivas para verificar la robustez y eficacia de los diferentes algoritmos implementados. Además, se considerará la refactorización del código si es necesario, con el fin de mejorar su estructura, claridad y mantenibilidad.
- **Análisis de resultados**: En esta fase se recopilarán datos de la ejecución de los algoritmos en sus diferentes versiones, así como entre ellos, utilizando los conjuntos de datos seleccionados para el proyecto. Esta recopilación de métricas permitirá una evaluación del rendimiento y la eficacia de cada algoritmo en comparación con los demás, así como su comportamiento en diferentes conjuntos de datos.
- **Documentación**: En esta etapa final se generará una documentación del proyecto que incluirá de forma general la descripción del problema, los objetivos, planificación, implementación, resultados y pruebas.

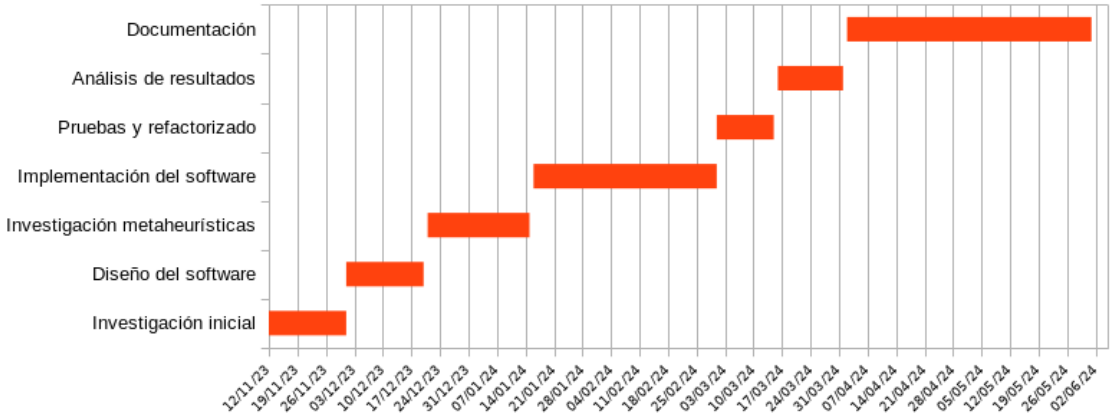

Figura 1.1: Diagrama de Gantt inicial

La planificación inicial tiene en cuenta un curso ideal del ciclo de vida del proyecto, siendo las estapas más extensas la de creación del software e implementación de la documentación. Son etapas excluyentes, no pueden ocurrir a la vez según este tipo de planificación. Al terminar una etapa se pasa inmediatamente a la siguiente.

La planificación final del proyecto se ha modificado significativamente debido a una serie de contratiempos y obstáculos surgidos durante su desarrollo, así como la influencia de numerosos eventos externos que han afectado a su cronograma. En particular, se han experimentado retrasos y bloqueos que han incidido en la duración prevista del proyecto. Por ejemplo, las etapas de implementación del software y la investigación de las metaheurísticas se han

entrelazado debido a que la implementación efectiva del algoritmo se facilitaba una vez que se había estudiado a fondo la metaheurística correspondiente. Esto ha llevado a una reevaluación de la estrategia de planificación original, reconociendo que no habría sido eficiente estudiar todas las metaheurísticas simultáneamente y luego proceder con su implementación.

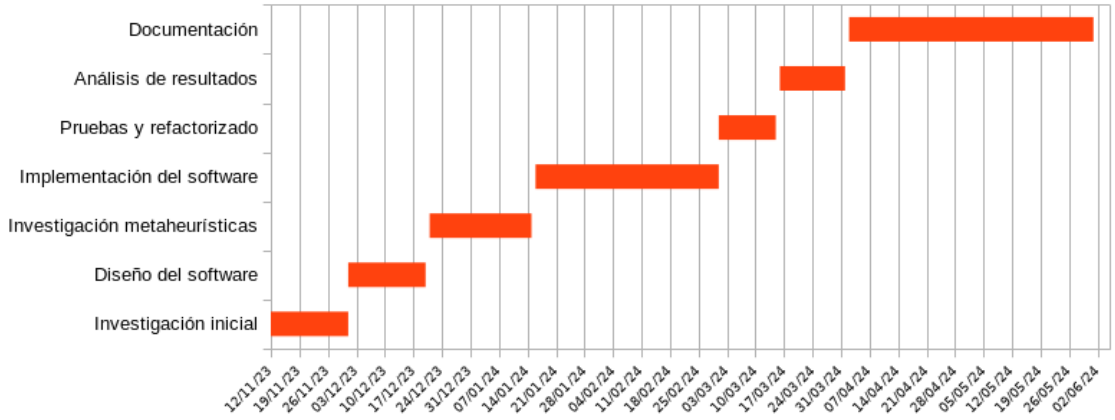

Figura 1.2: Diagrama de Gantt final

El coste estimado del proyecto se divide en varios subcostes:

- **Sueldo**: Teniendo en cuenta que el sueldo medio en España para un científico de datos es de 39.000€ [10] al año y que el proyecto consta de un solo trabajador, se puede estimar un salario de 18.75€/hora.

Capítulo 2

Bibliografía

[1] J. Miao y L. Niu, «A Survey on Feature Selection,» *Procedia Computer Science*, Promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016), vol. 91, págs. 919-926, ene. de 2016, ISSN: 1877-0509. DOI: 10.1016/j.procs.2016.07.111. URL: https://www.sciencedirect.com/science/article/pii/S1877050916313047 (visitado 02-04-2024).

[2] Y. S. Abu-Mostafa, M. Magdon-Ismail y H.-T. Lin, *Learning From Data*. AMLBook, 2012.

[3] Udacity, *Curse of Dimensionality - Georgia Tech - Machine Learning*, Retrieved 2022-06-29, feb. de 2015.

[4] K. Kira y L. Rendell, «A Practical Approach to Feature Selection,» English, 1992, págs. 249-256, ISBN: 978-1-55860-247-2. DOI: 10.1016/B978-1-55860-247-2.50037-1.

[5] C. Ding y H. Peng, «Minimum redundancy feature selection from microarray gene expression data,» English, *Journal of Bioinformatics and Computational Biology*, vol. 3, n.º 2, págs. 185-205, 2005, ISSN: 0219-7200. DOI: 10.1142/S0219720005001004.

[6] C. Cortes y V. Vapnik, «Support-vector networks,» en, *Machine Learning*, vol. 20, n.º 3, págs. 273-297, sep. de 1995, ISSN: 0885-6125, 1573-0565. DOI: 10.1007/BF00994018. URL: http://link.springer.com/10.1007/BF00994018 (visitado 02-04-2024).

[7] T. Cover y P. Hart, «Nearest neighbor pattern classification,» en, *IEEE Transactions on Information Theory*, vol. 13, n.º 1, págs. 21-27, ene. de 1967, ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.1967.1053964. URL: http://ieeexplore.ieee.org/document/1053964/ (visitado 02-04-2024).

[8] E. Fix y J. L. Hodges, «Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties,» *International Statistical Review / Revue Internationale de Statistique*, vol. 57, n.º 3, págs. 238-247, 1989, Publisher: [Wiley, International Statistical Institute (ISI)], ISSN: 0306-7734. DOI: 10.2307/1403797. URL: https://www.jstor.org/stable/1403797 (visitado 02-04-2024).

[9] W. Clark, W. N. Polakov y F. W. Trabold, *The Gantt chart, a working tool of management*, English. New York: The Ronald press company, 1922.

[10] PayScale. «Data Scientist Salary in Barcelona.» (Year), URL: https://www.payscale.com/research/ES/Job=Data_Scientist/Salary/9b2d8f8e/Barcelona.