Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection

David Charte

April 14, 2018

Contents

1 Resumen

La clave de este trabajo es desarrollar una técnica de selección de características mediante un algoritmo evolutivo (AE) que sea simultáneamente paralela y multi-objetivo.

1.1 Introducción

Algunas aplicaciones que presentan un alto número de características frente a ejemplos se encuentran en la bioinformática: análisis de microarrays, de secuencias, clasificación de electroencefalogramas (EEG). La selección de características es una técnica que reduce la dimensión de las instancias, sorteando la maldición de la dimensionalidad. Trabajos previos consideran selección paralela o selección multiobjetivo pero no ambos tipos conjuntamente. La selección de

características es útil tanto para clasificación [supervisada] como aprendizaje no supervisado.

1.2 Optimización multi-objetivo en selección de características no supervisada

De entre los métodos de reducción de dimensionalidad los autores se centran en selección de características con métodos *wrapper*. Se define el problema como la búsqueda de un vector de variables que verifique un conjunto de restricciones y "optimice" un vector de funciones, donde optimizar se entiende en el sentido de Pareto, en el que pueden existir varias soluciones no dominadas.

En clasificación no supervisada es importante considerar el sesgo de las medidas de calidad hacia soluciones de menor dimensionalidad (ya que las distancias entre puntos tienden a ser más similares conforme aumenta la dimensionalidad).

1.3 Algoritmos evolutivos multi-objetivo paralelos

Descomposición para paralelización:

- funcional (diferentes tareas simultáneas)
- de datos (misma tarea sobre diferentes datos simultáneamente)

Los AE se pueden paralelizar mediante descomposición de datos. Se van a considerar técnicas de paralelización que provean speedups superiores a los alcanzables simplemente ejecutando tareas independientes simultáneamente, aunque varíe el comportamiento respecto del algoritmo secuencial.

La descomposición de datos en un AE se puede abordar mediante:

- computación distribuida de la evaluación (*fitness*), lo cual no modifica la convergencia del algoritmo pero necesita comunicación entre el proceso que calcula los operadores y los que calculan la evaluación. Se implementa mediante arquitectura maestro-réplica.
- el uso de subpoblaciones, lo cual reduce la cantidad de comunicación requerida entre distintos procesos pero altera el comportamiento del algoritmo. Se puede implementar mediante arquitecturas maestro-réplica, de islas o de difusión.

Al usar subpoblaciones se puede decidir repartir los individuos asignando a la misma subpoblación aquellos de la misma zona del frente de Pareto, o bien los de las mismas zonas del espacio de búsqueda.

1.4 Selección de catacterísticas no supervisada multi-objetivo paralela

El objetivo es encontrar un conjunto de subpoblaciones lo suficientemente diverso como para realizar una búsqueda extensa del espacio de soluciones. En

las alternativas propuestas se divide el espacio de decisión entre los distintos procesadores y se dedica cada subpoblación a explorar zonas casi disjuntas. Para conseguir esto, algunas de las componentes de todos los individuos de una subpoblación se fijan en cada evolución, de forma que recorren un subespacio del espacio de búsqueda. En algunas aplicaciones esto tiene más sentido considerando que las características se organizan en subconjuntos con significado propio (e.g. clasificación de EEG).

Notación: Si tenemos p subpoblaciones, r individuos por población, e grupos de características y f características por grupo, un individuo es una asignación de grupos y características al conjunto binario $\{0, 1\}$ de la forma:

$$\alpha : \{1, \ldots, e\} \times \{1, \ldots, f\} \to \{0, 1\}$$

El espacio de soluciones $\mathcal{D}$ está formado por todas las posibles asignaciones (equivalentemente, los vértices del hipercubo unidad de dimensión ef), y su tamaño es 2^{ef}. Una subpoblación es un subconjunto de $\mathcal{D}$ de tamaño r. En el trabajo se asume la simplificación $e = p$.

Las tres alternativas descritas a continuación se diferencian en el método para combinar subpoblaciones tras varias generaciones.

1.4.1 Alternativa 1

Se recogen las p subpoblaciones y se genera una nueva población "base" de r individuos. Para ello, se tienen en cuenta qué individuos son no dominados: si hay más de r se utiliza alguna estrategia para descartar los sobrantes, por ejemplo eliminando los individuos más cercanos a otro; si hay menos de r no dominados se seleccionan algunos de entre el segundo nivel de no-dominación, etc. La nueva población seleccionada se toma como base de las p subpoblaciones, y en cada una se fijan todos los grupos de características salvo uno (así, la subpoblación i-ésima recorre el i-ésimo grupo).

1.4.2 Alternativa 2

A diferencia de la alternativa 1, en este caso se fuerza a que las componentes constantes de cada subpoblación tengan los mismos valores para todos los individuos, de forma que todos ellos exploran el mismo subespacio. En la etapa de combinación de subpoblaciones se selecciona una población de pr individuos no dominados y se envían a cada isla, que los evalúan y definen r individuos que comparten las mismas componentes constantes.

1.4.3 Alternativa 3

En este caso, se comienza cada subpoblación como en la alternativa 2. El procedimiento de combinación genera una nueva j-ésima subpoblación escogiendo $r - q$ individuos no dominados de la misma y añadiéndole un número fijo q de soluciones no dominadas encontradas por el resto de subpoblaciones. El procesador que gestiona la subpoblación j recibe esta nueva subpoblación y la mejora durante varias generaciones.

1.5 Resultados experimentales

Se han realizado experimentos para comparar cuatro variantes de NSGA-II: una en la que se utiliza la computación distribuida de la evaluación y tres en las que se implementan cada una de las alternativas para el uso de subpoblaciones descritas anteriormente. Estas se comparan también en ejecuciones con distintos números de procesadores. Se ha escogido SOM como clasificador con dos medidas de rendimiento a optimizar, buscando mínimas distancias entre individuos cercanos y máximas entre individuos lejanos. Además, los datasets que se han seleccionado como benchmark verifican que tienen más características que ejemplos.

Se ha comparado la técnica desarrollada de selección de características con otros métodos previos de tipo wrapper, tanto con clasificadores supervisados como no supervisados (las métricas de evaluación internas son adecuadas para cada tipo), en 2 de los benchmarks. La propuesta resulta ser competitiva frente a las comparadas.

Usando todos los benchmarks se han comparado las cuatro variantes paralelas propuestas frente a una implementación secuencial del algoritmo evolutivo. Las diferencias en general no resultan significativas, pero se obtienen mejoras en eficiencia superlineales para las alternativas 2 y 3, con 4, 6, y 8 procesadores. Se puede llegar a observar un compromiso entre la bondad de una solución y la mejora en velocidad de la técnica utilizada.

Por último, se han estimado los parámetros de modelos que explican la ganancia de velocidad de cada alternativa mediante los datos experimentales. Alt. 2 y 3 presentan comportamientos similares, con ganancias superlineales, mientras que la versión de cálculo distribuido y Alt. 1 consiguen una ganancia menor de 1 por procesador. Se plantea si una distribución del trabajo por islas o difusión aprovecharía mejor los recursos (e.g. multicomputadores). Se plantean también las posibles variaciones en el comportamiento al sustituir NSGA-II por otro AE multi-objetivo como SPEA2.

1.6 Conclusiones

Se han desarrollado nuevas estrategias paralelas de resolución de problemas multi-objetivo, y se han aplicado a datos de EEG. Algunas de estas estrategias incluso mejoran en ocasiones concretas al algoritmo secuencial. Las ganancias de velocidad son buenas, y Alt. 2 y 3 mejoran a las otras dos estrategias, presentando un compromiso entre calidad y velocidad. Estudios futuros podrían abordar más AEs multiobjetivo base, la arquitectura de islas, el efecto de los parámetros evolutivos en el rendimiento y las funciones objetivo en cuando a clustering no supervisado.

2 Comentario

Este trabajo aborda un problema de selección de características mediante algoritmos evolutivos multiobjetivo paralelos. Se centra en una aplicación a señales

de EEG con clustering mediante mapas autoorganizativos (SOM) pero podría ser aplicable a más ámbitos. Las propuestas realizadas son el cálculo distribuido de la evaluación y tres estrategias novedosas que enfocan el problema en realizar una buena exploración del espacio de búsqueda, repartiendo en distintos procesadores diferentes subespacios o subconjuntos del mismo, para después reunir el aprendizaje realizado y volver a iterar. Los autores apoyan sus propuestas en una completa experimentación frente a otros selectores de características de tipo *wrapper* y frente a la versión secuencial del AE. Se consideran tanto los resultados de rendimiento en agrupamiento como los de ganancia de velocidad en el cómputo de las soluciones, y se mencionan algunos aspectos (arquitectura paralela, algoritmo base) que podrían variar el comportamiento de las propuestas.

El trabajo realizado presenta una dificultad notable ya que requiere del uso y dominio de conceptos variados, como los algoritmos evolutivos, la paralelización, los mapas autoorganizativos y el problema de selección de características. Los resultados obtenidos son interesantes y abrirían las puertas a utilizar estas estrategias en otras aplicaciones.

Un aspecto a pulir podría ser la motivación del propio problema de selección de características. Al hablar de la maldición de la alta dimensionalidad (*curse of dimensionality*) referencia a Raudys y Jain que hablan únicamente de las consecuencias de tener pocas instancias frente a características. Más adelante referencia a Handl y Knowles que sí mencionan además alguna consecuencia de la alta dimensionalidad independiente del número de instancias, en tanto que las distancias en alta dimensionalidad suelen ser más similares entre sí que en pocas dimensiones. Sin embargo, hay más resultados interesantes como Beyer et al. que habla de cómo el punto más lejano y el más cercano a uno dado tienden a estar a distancias muy similares conforme se aumenta el número de dimensiones. En definitiva, limitar este fenómeno al caso donde se tienen más características que instancias supondría cerrar las puertas a un gran número de aplicaciones donde la cantidad de características sigue siendo un problema, incluso aunque se tengan muchas instancias.

Un punto contraintuitivo en cuanto a la presentación del trabajo son las figuras 4 y 5. En la descripción de la figura 4 se afirma que cada eje representa los valores de un grupo de características, lo cual se puede ver como una simplificación para aportar interpretabilidad (pese a que los posibles valores de cada grupo de características forman un conjunto discreto de 2^f puntos). Sin embargo, a las soluciones que quedan más cerca de los ejes se las designa como "no dominadas", lo cual causa confusión ya que la gráfica no representa los valores de la función objetivo, luego sería imposible saber a simple vista cuáles de ellas son no dominadas. Además, la exploración de áreas en la figura 4 se representa mediante rectas, que podría interpretarse como subespacios afines del espacio de búsqueda. Al no ser el espacio de búsqueda un conjunto con valores continuos, menos aún un espacio vectorial real, esto puede resultar contraintuitivo y requeriría una aclaración de lo que se está intentando representar.

Como posibles extensiones del trabajo, más allá de lo comentado en las conclusiones, una posibilidad sería considerar técnicas de extracción de caracterís-

ticas y su posible entrenamiento mediante algoritmos evolutivos paralelizados. Por ejemplo, se podrían entrenar autoencoders (Charte et al.) mediante neuroevolución (Petroski et al.).