

Universidad de La Habana
Facultad de Matemática y Computación



Optimización Multiobjetivo para Autogoal

Autor:

Rodrigo Daniel Pino Trueba

Tutores:

Suilan Estevez

Daniel Valdes

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Fecha

github.com/username/repo

Dedicación

Agradecimientos

Agradecimientos

Opinión del tutor

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

Introducción	1
1. Estado del Arte	4
1.1. Optimización Multiobjetivo	4
1.1.1. Técnicas de Escalarización (Scalarization)	5
1.1.2. Algoritmos Numéricos	6
1.1.3. Algoritmos Genéticos Multiobjetivos (MOEA)	6
1.2. Optimización Multiobjetivo y AutoML	7
2. Propuesta	8
3. Detalles de Implementación y Experimentos	9
Conclusiones	10
Recomendaciones	11

Índice de figuras

Ejemplos de código

Introducción

Es frecuente encontrar problemas de Optimización Multiobjetivo tanto en el día a día como en diversos campos de la ciencia y la técnica. Por ejemplo, una empresa de automóviles que desea maximizar la potencia del motor y a la misma vez minimizar el consumo de combustible o un inversionista que quiere un balance adecuado entre ganancia y riesgo.

Optimización Multiobjetivo(OM), también conocida como optimización vectorial o optimización de Pareto, explora un espacio de decisión tratando de optimizar varias funciones objetivos simultáneamente. Estos criterios a optimizar suelen entrar en conflicto haciendo imposible la existencia de una solución óptima única al problema multiobjetivo (MOP por sus siglas en inglés), en cambio existen un conjunto de puntos (no necesariamente finito) donde no es posible mejorar un aspecto del resultado sin empeorar otro. Este conjunto de puntos igual de buenos e inmejorables se les conoce como frente de Pareto. El resultado esperado de un buen algoritmo de OM dado un problema es la búsqueda de un subconjunto representativo del frente de Pareto para que luego el investigador decida cual de todas las posibles soluciones se adapta mejor a su problema.

Motivación

Un modelo de Aprendizaje de Máquina o *Machine Learning(ML)* tiene como objetivo maximizar la relevancia de sus resultados dado una métrica. Para medir dichas relevancia, se utilizan métricas como *accuracy* o *f-score*.

Cuando el proposito de un programa de AutoML es producir un algoritmo ML que minimice la relevancia a costa de cualquier otro tipo de criterio se producen buenas predicciones, pero no necesariamente convenientes. Existen casos donde el investigador esta dispuesto a sacrificar un poco de relevancia en favor de otros criterios dependiendo del contexto donde se encuentre.

Existen situaciones donde la interpretabilidad que implica saber, sin ser un experto en Aprendizaje de Máquina, porque el sistema tomó ciertas decisión puede ser

importante. También existen casos donde es necesario producir sistemas de ML justos, para evitar que el algoritmo de ML tenga algún tipo de sesgo por género, color de piel u otro. Estos problemas se resolverían muy bien con multiobjetivo; en caso de no optimizar la relevancia y optimizar únicamente para justeza o interpretabilidad respectivamente produciría un sistema inútil debido a sus malas predicciones.

Optimización multibobjetivo en AutoML abrió las puertas también a métricas que no tienen mucho sentido usarlas individualmente como *precisión* y *recobrado*. Precisión mide la proporción entre los valores relevantes identificados y todos los valores identificados mientras que recobrado mide la proporción entre los valores relevantes identificados y todos los valores relevantes. Estas medidas usadas independientemente no son representativas pues basta para tener un recobrado perfecto seleccionar todos los elementos del conjunto de datos, y una precisión casi perfecta seleccionando un pequeño conjunto de elementos. Para lograr un balance entre estas se utiliza *f-score*, que relaciona ambas medidas y permite darle más peso a una o a la otra según determine el investigador, no obstante con OM se pueden obtener un sistema ML que optimice para ambas.

Un framework de *Automated Machine Learning*(AutoML) que pueda optimizar simultáneamente para varios objetivos, se beneficia al ser capaz de ofrecer un conjunto de modelos igual de buenos, y dejar al investigador que seleccione el que más le convenga según su contexto.

Antecedentes

En el entorno del grupo de Inteligencia Artificial de la Universidad de La Habana se desarrolla AutoGOAL una herramienta de AutoML que permite obtener modelos optimizados con respecto a una sola métrica.

Problemática

AutoGOAL no presenta actualmente una herramienta para resolver problemas multiobjetivos. Se propone la adición de un decisor que permita producir varios modelos de ML, cada uno perteneciente y a la vez representativo del frente de Pareto.

Objetivo

Objetivo General

Resolver problemas de AutoML utilizando optimización multi-objetivo.

Objetivos Específicos

- Optimización Multiobjetivo utilizando herramientas de scalarization:
 - Linear Weighting
 - Chebychev Distance
- Optimización Multiobjetivo Utilizando metahurísticas (PGE, proveniente de EDA)
- Comparar/Probar los resultados

Estructura de la Tesis

Capítulo 1

Estado del Arte

1.1. Optimización Multiobjetivo

Optimización Multiobjetivo es la rama de la optimización que se dedica a optimizar varias funciones simultáneamente.

Optimización Multiobjetivo: Dado m funciones objetivos: $f_1 : \mathcal{X} \rightarrow \mathbb{R}, \dots, f_m : \mathcal{X} \rightarrow \mathbb{R}$ que dado un vector en el espacio de decisión \mathcal{X} \mathbb{R} . Un problema multiobjetivo esta dado por:

$$\text{mín } f_1(x), \dots, f_m(x), x \in \mathcal{X}$$

En la presencia de varias funciones objetivos ya no es posible hablar de solución única o mínimo. Es usual que estos criterios de decisión entren en conflictos entre sí y no sea posible encontrar una solución que satisfaga ambas métricas, en cambio, se encuentran soluciones que si se intentan optimizar en algun aspecto, inevitablemente empeoran en otros. Estas soluciones son las que se conocen como soluciones del frente de Pareto. Con el concepto de Pareto viene el concepto de Pareto dominación.

Pareto Dominación: Dados vectores en el el espacio objetivo, $x \in \mathbb{R}^m$ y $y \in \mathbb{R}^m$, se dice que x Pareto domina a y (i.e. $x \prec y$), si y solo si:

$$\forall i \in \{1, \dots, m\} : x_i \leq y_i \text{ and } \exists j \in \{1, \dots, m\} : x_j < y_j$$

Frente de Pareto: El conjunto de soluciones que no son dominadas por ningun otra solución en el espacio.

Encontrar soluciones en el frente de Pareto no es tarea difícil, basta con encontrar el mínimo de cada función objetivo, soluciones que indudablemente pertenecen al

frente de Pareto. La tarea difícil se encuentra en tratar una aproximación del frente de Pareto que sea representativa de este.

Se debe notar que el termino Optimización Multiobjetivo suele referirse cuando se intenta optimizar simultáneamente para dos o tres funciones objetivos. Para un cantidad de criterios de optimización grande se le suele conocer coloquialmente Optimización para Muchos Objetivos o *many-objective optimization* en inglés. Existe una diferenciación en este pues al aumentar le número de métricas a optimizar:

1. ya no es posible la visualización del frente;
2. la computación de indicadores o de selección se convierte en problemas NP-duros;
3. existe un rápido crecimiento de puntos no dominados, mientras mayor número de objetivos, la probabilidad de que un punto sea no dominado en un set con distribución normal tiende exponencialmente a 1.

1.1.1. Técnicas de Escalarización (Scalarization)

Los primeros intentos de resolver el MOP fue utilizando técnicas de Escalarización, que en resumen es cuando de alguna manera u otra se combinan todas las funciones objetivos en una sola. Existen varias maneras de esto:

1. Linear Weighting: Todas las funciones objetivos se combinan en una sola, y a cada una se le asigna un peso. Modificando estos pesos se obtiene una solución distinta del frente de Pareto.

$$\min \sum w_i f_i(x), x \in X$$

Esta enfoque presenta el problema de que cuando el frente de Pareto no es convexo (tiene alguna porción cóncava) no es posible obtener soluciones en esta zona, no importa como se modifiquen los pesos w_i

2. ϵ -constrain: Se selecciona una función objetivo como principal y las demás se establecen como restricciones de esta, y tienen que ser menor que cierto ϵ_i con $1 \leq n - 1$ por cada función objetivo.

$$\begin{aligned} \min f_1(x), x \in X, \text{ sujeto a:} \\ g_i(x) \leq \epsilon_i \quad \forall i, 0 \leq i \leq n - 1 \end{aligned}$$

Utilizar esta técnica tiene dos dificultades principales, primero la selección de los ϵ_i , que para una adecuada selección requieren conocimiento previo del frente de Pareto y al igual que *Linear Weighting* no es capaz de obtener soluciones en las partes cóncavas del frente.

3. Chebychev Distance (CSP): Se establece un punto z^* que domina a todo el frente de Pareto y se utiliza la distancia de Chebychev como función objetivo utilizando un vector de pesos λ

$$\min \max_{i \in 1, \dots, m} \lambda_i |f_i(x) - z_i^*|, x \in X$$

CSP dado los pesos indicados puede encontrar cualquier punto del frente de Pareto, no importa si es cóncavo o convexo.

1.1.2. Algoritmos Numéricos

En principio todos los algoritmos de escalarización se pueden resolver utilizando métodos numéricos.

Además, existen métodos numéricos que intentan resolver el problema haciendo cumplir las condiciones de Karush-Kuhn-Tucker.

La idea va de encontrar al menos una solución al sistema de ecuaciones creado por tratar de resolver KKT. Luego utilizando métodos de continuación y homotopía para añadir al conjunto solución soluciones cercanas a estas. Este algoritmo requiere que las soluciones satisfagan las condiciones de convexidad local y diferenciabilidad.

También existen otros métodos para la búsqueda de mínimos globales como *Técnicas de Subdivisión*, *Optimización Global Bayesiana* y *Optimización de Lipschitz*. Estas requieren de que el espacio de decisión tenga una baja dimensionalidad.

Aun más, también se investiga activamente métodos numéricos que no dependan de derivadas para su resolución.

1.1.3. Algoritmos Genéticos Multiobjetivos (MOEA)

Los algoritmos genéticos tuvieron sus inicios en la década del 60 y fueron usados principalmente en resolución de problemas numéricos combinatorios y no convexos. Utilizan paradigmas extraídos de la naturaleza, tal como selección natural, mutación y recombinación para mover una población (o conjunto de vectores de decisión) hacia una solución óptima.

Los algoritmos genéticos multiobjetivos generalizan esta idea, y son diseñados para en cada iteración acercarse cada vez más al frente de Pareto. Como en este caso no existe solución única la manera de seleccionar los individuos cambia fundamentalmente.

Dentro de los MOEA existen tres paradigmas principales:

1. **MOEA basados en el frente de Pareto:** Se identifican porque dividen el proceso de selección en dos etapas. Una primera donde seleccionan los individuos

según su índice de dominación, donde las soluciones que pertenecen al frente de Pareto tienen índice cero. Luego se realiza una segunda selección entre los ya seleccionados utilizando una segunda estrategia de puntuación. NSGA-II y SPEA2 son algoritmos de este tipo.

2. **Basados en indicador:** Estos utilizan un indicador para calcular cuan cercano es el conjunto actual al frente de Pareto (unario), o cuanto mejora el nuevo conjunto de soluciones respecto a la iteración anterior (binario). Ejemplo de este es SMS-EMOEA que suele converger al frente de Pareto con soluciones igualmente distribuidas.
3. **Basados en descomposición:** La idea principal consiste en descomponer el problema en pequeños subproblemas cada uno correspondiente a una sección del frente de Pareto. Por cada sub-problema se resuelve utilizando escalarización con diferente parametrización. El método de escalarización más usado en estos casos suele ser CSP debido a ser capaz de obtener cualquier punto del frente de Pareto. Ejemplo de este paradigma son MOEA/D y NSGA-III.

1.2. Optimización Multiobjetivo y AutoML

Aunque es cada vez un feature de más demanda en la actualidad, no se conoce ningún sistema de AutoML que tenga implementado multiobjetivo, excepto por TPOT que optimiza sus pipelines mutuamente para ‘accuracy’ y tiempo de entrenamiento. No obstante no presenta flexibilidad sobre que métricas optimizar.

Capítulo 2

Propuesta

Capítulo 3

Detalles de Implementación y Experimentos

Conclusiones

Conclusiones

Recomendaciones

Recomendaciones