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. Algoritmos Numéricos 5

1.1.2. Algoritmos Genéticos Multiobjetivos (MOEA) 5

1.2. Optimización Multiobjetivo y AutoML 6

2. Propuesta **7**

3. Detalles de Implementación y Experimentos **8**

Conclusiones **9**

Recomendaciones **10**

Índice de figuras

Ejemplos de código

Introducción

Optimización Multiobjetivo(OM), también conocida como optimización vectorial o optimización de Pareto, explora un espacio de decisión tratando de optimizar múltiples criterios simultáneamente. Usualmente, estos criterios entran en conflicto haciendo imposible la existencia de una solución óptima única de cada uno, en cambio existen una serie de puntos (no necesariamente finita) donde no es posible optimizar un criterio sin empeorar el resultado de otro. Este conjunto de puntos igual de buenos conforman lo que se conoce como el 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.

Es frecuente encontrar problemas de OM 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 minimzar el consumo de combustible, o a la hora de realizar inversiones, se prefiere no tener riesgos innecesarios si la ganancia no lo merece.

Motivación

Un modelo de Aprendizaje de Máquina o *Machine Learning(ML)* tiene como objetivo maximizar la relevancia de sus resultados dado una sola métrica. Para medir dichas relevancia, se utilizan métricas como la *precisión* y *recobrado*.

Precisión mide la proproción 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 resultan útiles 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 evitar estos defectos y lograr un balance entre ellas se utiliza *F-score*, que relaciona ambas medidas y permite darle más peso a una o a la otra según determine el investigador.

Esto resuelve en parte la limitación de solo ser posible optimizar según un solo criterio. En el case de desear añadir métricas adicionales, requerería la modificación de F-score o la creación de una nueva que agrupe todas las métricas. Al permitir optimizar según varios criterios simultáneamente se remueve completamente este obstáculo, y es posible agregar con relativa facilidad todas las métricas que se deseen y de cualquier tipo como métricas de equidad (miden el nivel de justeza cuando el modelo trata con capital humano).

Un framwork de *Automated Machine Learning(AutoML)* se beneficia de modelos que permiten optimizar para varios objetivos simultáneamente, pudiendo retornar un conjunto de estos, cada modelo representando un punto del frente de Pareto.

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 represantivo 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

Existen múltiples técnicas para la Optimización Multiobjetivo. No es problemático encontrar soluciones en el frente de Pareto, pues cada solucion de cada funcion objetivo pertenece a este, sino encontrar un conjunto de soluciones que sean representativo de este.

Existen tres enfoques prinincipales para resolver estos problemas.

Técnicas de Escalarización (Scalarization)

Consisten en convertir el problema multiobjetivo, en un problema de un solo objetivo combinando las funciones objetivos. Ejemplos:

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.

$$\text{mín} \sum w_i f_i(x), x \in X$$

   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 sierto ϵ_i con $1 \leq n-1$ por cada función objetivo.

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

Utilizar esta técninca tiene dos dificultades principales, primero la selección de los ϵ_i, que para una adecuada selección requiren conocimiento previo del frente de Pareto y al igual que *Linear Weighting* no es capaz de obtner 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

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

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

1.1.1. Algoritmos Numéricos

En principio todos los algoritmos de escalarización se pueden resolver utilzando métodos númericos.

Además, existen métodos númericos 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 require que las soluciones satisfagan las condiciones de convexidad local y diferenciabilidad.

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

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

1.1.2. Algoritmos Genéticos Multiobjetivos (MOEA)

Los algorimtos genéticos tuvieron sus inicios en la decada del 60 y fueron usados principalmente en resolucion de problemas numéricos combinatorios y no convexos. Utilizan paradigmas extraídos de la naturaleza, tal como seleccion natural, mutación y recombinación para mover una población (o conjunto de vectores de decisión) hacian 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 selccionar 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 invidiuos seguún su indíce de dominación, donde las soluciones que pertenecen al frente de Pareto tienen indíce 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. **Basado 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 suele converger al frente de Pareto con soluciones igualmente distribuidas.

3. **Basado en descomposición**: La idea principal consiste en descomponer el problema en pequeños subproblemas cada una correspondiente a una sección del frente de Pareto. Por cada sub-problema se resuelve utilizando escalarización con dierente paramtrización. 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 NSPGA-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 entranamiento. No obstante no presenta flexibiliad sobre que métricas optimizar.

Se debe notar que el termino Optimización Multiobjetivo suele referirse en la literatura cuando se intenta optimzar simultaneamente para dos o tres funciones objetivos. Añadir métricas adicionales corresponde un perfomance hit pues el frente de pareto crece exponencialmente.

Cuando se quieren optimizar para muchas métircas se conoce como Optimización para Muchos Objetivos, o *many-objective optimization* en inglés. Para este último los enfoques más capaces han sido los basados en descomposción.

Capítulo 2

Propuesta

Capítulo 3

Detalles de Implementación y Experimentos

Conclusiones

Conclusiones

Recomendaciones

Recomendaciones