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

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 minimzar 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 necesariemente convenientes. Existen casos donde el investigador esta dispuesto a sacrificar un poco de precisión en las predicciones en favor de otras métricas 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 necsario 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 multibojetivo en AutoML abré las puertas también a métricas que no tienen mucho sentido usarlas individualemente como *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 métircas usadas por separadas no son represantivas 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 smultáneamente 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 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

Optimización Multiobjetivo es la rama de la Ciencia y la Matemática que se dedica a optimizar para varias funciones objetivos simúltaneamente.

Optimización Multiobjetivo: Dado m funciones objetivos: $f_1 : \mathcal{X} \to \mathbb{R}, ..., f_m : \mathcal{X} \to \mathbb{R}$ que dado un vector en el espacio de decisión $\mathcal{X}$ es transformado en un valor de $\mathbb{R}$. Más formalmente:

$$\text{MOP: mín } f_1(x), ..., f_m(x), x \in \mathcal{X}$$

En presencia de varias funciones objetivos ya no es posible hablar de una solución única mejor que el resto. Como las solucions se comparan respecto a dos o más aspectos, se habla de dominación.

Pareto Dominación: Dados vectors 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:

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

Existe el caso en el conjunto de soluciones que existan soluciones que no sean dominadas por ninguna otra solución. Estas soluciones vienen siendo el conjunto solucion del problema de optimziación multiobjetivo.

Frente de Pareto: Todos los vectores x del espacio objetivo $\mathcal{Y}$ tal que no exista $y \prec x$.

$$\mathcal{P} = \{x | x, y \in \mathcal{Y}, \neg \exists y \prec x\}$$

Encontrar puntos del frente de Pareto no es tarea díficil, puede ser tan trivial con encontrar el mínimo de cada función objetivo. La complejidad yace en encontrar un conjunto solución que sea lo más aproximado posible al frente de Pareto.

Se debe notar que el termino Optimización Multiobjetivo se utiliza cuando se optimza simultáneamente para dos o tres funciones objetivos. Para un cantidad de métricas mayor se le conoce coloquialmente en la literatura como Optimización para Muchos Objetivos o *many-objective optimization* en inglés. Se hace enfásis en esta diferenciación este pues al aumentar le número de métricas a optimizar:

1. ya no es posible visualizar el frente obtenido;
2. la computación de indicadores o de selección para muchos algoritmos 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 manera clásica de resolver MOP es utilizando técinas de Escalarización que es cuando de alguna manera u otra se combinan todas las funciones objetivos en una sola. Existen diferentes técinas de combinar estas funciones:

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.

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

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

3. Chebychev Distance (CSP): Se establece un punto de referencia z^* y se utiliza la distancia de Chebychev de los vectores objetivos hacia este como función objetivo utilizando un vector de pesos $\lambda \in \mathbb{R}^m_{\prec 0}$, donde $\mathbb{R}^m_{\prec 0} = \{x | x \in \mathbb{R}^m, x \prec 0\}$.

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

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

1.1.2. 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.

1.1.3. 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. **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 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 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 entranamiento. No obstante no presenta flexibiliad sobre que métricas optimizar.

Capítulo 2

Propuesta

Capítulo 3

Detalles de Implementación y Experimentos

Conclusiones

Conclusiones

Recomendaciones

Recomendaciones