



Universiteit
Leiden
The Netherlands

Efficient tuning of automated machine learning pipelines

Nguyen, D.A.

Citation

Nguyen, D. A. (2024, October 9). *Efficient tuning of automated machine learning pipelines*. Retrieved from <https://hdl.handle.net/1887/4094132>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4094132>

Note: To cite this publication please use the final published version (if applicable).

Nederlandse Samenvatting

Automatische machine learning technieken (AutoML) zijn sterk in opkomst vanwege het success van deze technieken in het reduceren van de ontwikkelingscyclus in praktische toepassingen. Optimalisatie speelt een cruciale rol in AutoML-frameworks door te helpen bij het identificeren van een nauwkeurig afgestelde ML-pijplijn die geschikt is voor een bepaald praktisch probleem. Er zijn verschillende geavanceerde optimalisatiebenaderingen voorgesteld, waaronder Bayesiaanse optimalisatie, Bandit-leren en raceprocedures, om de prestaties van AutoML te verbeteren.

De bestaande studies formuleren de AutoML-optimalisatie echter vaak als een Hyper Parameter Optimalisatie (HPO)-probleem met behulp van de Combined Algorithm Selection and Hyperparameter optimalisatie (CASH)-benadering. Dit beperkte perspectief kan hun effectiviteit bij het aanpakken van het onderliggende probleem beperken.

In dit proefschrift gaan we uitvoerig in op deze beperking door *AutoML-optimalisatie te formuleren*, die de volledige ML-pijpijnsynthese omvat, zoals besproken in Hoofdstuk 1. Specifiek introduceren we een nieuwe klasse hyperparameter – genaamd "*algorithm choice*", die het modelleren van algoritmeselectie mogelijk maakt. Deze klasse bevat een uniek attribuut dat de hiërarchische organisatie van algoritmen mogelijk maakt op basis van hun technische overeenkomsten. Voor zover wij weten, is dit de eerste poging om de relatie tussen algoritmen te visualiseren.

Vervolgens vindt u in de hoofdstukken 2 en 3 een uitgebreid onderzoek om een holistisch begrip van AutoML en de optimalisatie ervan te bieden. Hoofdstuk 2 onderzoekt AutoML en zijn integrale concepten en aspecten uitgebreid. Het diept in de fundamenteën van AutoML, onderzoekt de motivaties achter de ontwikkeling ervan en draagt bij aan het verkorten van de ontwikkelingscyclus van machine learning voor toepassingen in de echte wereld. Hoofdstuk 3 richt zich specifiek op AutoML-optimalisatie. Dit hoofdstuk gaat dieper in op verschillende

optimalisatietechnieken en algoritmen die worden gebruikt om de prestaties en efficiëntie van AutoML-frameworks te verbeteren. Het onderzoekt benaderingen zoals rasterzoeken, willekeurig zoeken, Bayesiaanse optimalisatie, Bandit-leren en raceprocedures, en biedt een diepgaande analyse van hun principes, sterke punten en beperkingen in de context van AutoML-optimalisatie.

Een ander punt van zorg is benchmarking om de robuustheid en algemene toepasbaarheid van optimalisatiebenaderingen empirisch te evalueren. In Hoofdstuk 4 worden twee sets benchmarkexperimenten gepresenteerd. Deze benchmarks bestaan uit 117 overeengekomen datasets die verwerking vereisen via een ML-pijplijn van meerdere operatoren, zoals codering, normalisatie, hersteekproeven en classificatie. De gekozen datasets kunnen categorische gegevens bevatten, onvolledige instanties hebben, of een onevenwichtige distributie hebben. Verder omvatten de benchmarks gestandaardiseerde zoekruimtes, samen met een experimentele methodologie en opstelling voor benchmarkdoeleinden.

In Hoofdstuk 5 hebben we een gedetailleerde studie uitgevoerd naar de effectiviteit van AutoML-optimalisatie bij het omgaan met problemen van klassenonevenwichtigheid. Onze bevindingen suggereren dat Bayesiaanse Optimalisatie (BO) een zeer efficiënte aanpak is om dit probleem aan te pakken. Deze ontdekking stelde ons in staat om ons te concentreren op het verbeteren van BO om AutoML-optimalisatieproblemen op te lossen. Als gevolg daarvan konden we twee effectieve optimalisatiealgoritmen creëren op basis van BO in Hoofdstuk 7 en Hoofdstuk 8, evenals succesvol de AutoML-optimalisatiebenadering implementeren in een praktische toepassing in Hoofdstuk 6.

Een andere belangrijke bijdrage is de ontwikkeling van een prestatie-metrick die specifiek is ontworpen om de dubbele uitdagingen van klassenonevenwichtigheid en ongelijke belangrijkheidsproblemen aan te pakken. Deze metrick, besproken in Hoofdstuk 6, biedt een uitgebreid evaluatiekader voor het beoordelen van de prestaties van ML-modellen in dergelijke scenario's.

Naast de genoemde bijdragen presenteert deze scriptie twee nieuwe AutoML-optimalisatiebenaderingen. In Hoofdstuk 7 hebben we een combinatie sampling methode gepresenteerd die AutoML-optimalisatieproblemen efficiënt oplost via Bayesiaanse optimalisatie. Onze aanpak is ontworpen om de dekking van ML-algoritme-monsters in de zoekruimte te maximaliseren, wat resulteert in een robuust surrogaatmodel voor BO. We hebben experimenten uitgevoerd op 117 benchmark-datasets en ontdekt dat het gebruik van onze sampling benadering de prestaties van BO aanzienlijk verbetert in vergelijking met BO zonder onze aanpak. Naast

de voorgestelde sampling benadering hebben we ook een optimalisatiesoftware geïntroduceerd met de naam BO4ML in dit hoofdstuk. Deze software is geïmplementeerd op basis van de voorgestelde sampling benadering en kan worden gebruikt voor optimalisatie. Verder, in Hoofdstuk 8, hebben we een verbeterde AutoML-optimalisatietechniek voorgesteld, bekend als DACOpt, een efficiënte concurrerende procedure. Deze innovatieve methode beoogt het effectiever toewijzen van afstemmingsbronnen aan optimalisatoren over zoekgebieden. Specifiek beloont DACOpt middelen aan gebieden die belofte tonen, terwijl het optimalisatieproces wordt beëindigd in gebieden die minder waarschijnlijk gewenste resultaten opleveren. Onze experimenten op 117 benchmarkdatasets geven aan dat onze voorgestelde aanpak aanzienlijke prestatievoordelen biedt ten opzichte van BO voor AutoML-optimalisatieproblemen.