Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
Demonstrations Track

AutoML for Outlier Detection with Optimal Transport Distances

Prabhant Singh, **Joaquin Vanschoren**
Eindhoven University of Technology
{p.singh, j.vanschoren}@tue.nl

Abstract

Automated machine learning (AutoML) has been widely researched and adopted for supervised problems, but progress in unsupervised settings has been limited. We propose "LOTUS", a novel framework to automate outlier detection based on meta-learning. Our premise is that the selection of the optimal outlier detection technique depends on the inherent properties of the data distribution. We leverage optimal transport to find the dataset with the most similar underlying distribution, and then apply the outlier detection techniques that proved to work best for that data distribution. We evaluate the robustness of our framework and find that it outperforms all state-of-the-art automated outlier detection tools. This approach can also be easily generalized to automate other unsupervised settings.

1 Introduction

AutoML [Hutter *et al.*, 2019] has shown robust and reliable performance in model selection and hyperparameter optimization [Hutter *et al.*, 2019; Feurer *et al.*, 2015]. However, research in automated machine learning has been highly focused on supervised machine learning, where we can use model performance evaluated on a held-out validation set as a ground truth metric to optimize while searching over the model search space [Thornton *et al.*, 2013]. Unsupervised settings lack such a ground truth, hence AutoML research in this area is rather sparse. Outlier detection (OD) is an example of one of these unsupervised problems. It aims to identify data points that are significantly different from the rest of the data. These outliers can be caused by errors in the data collection process, incorrect values, or unusual events. Detecting these allows us to improve the quality of the data or help find unusual events that could be interesting to different business and scientific domains.

In this work, we propose a novel AutoML framework for unsupervised tasks that leverages meta-learning [Vanschoren, 2018] and optimal transport [Peyré and Cuturi, 2019; Scetbon and Cuturi, 2022] to transfer information from similar prior datasets (or synthetic datasets) on which outliers are known. We call this framework *Learning to learn with Optimal Transport for Unsupervised Scenarios*, or **LOTUS**.

In this work, we make the following three contributions:

- **A Meta-learner for outlier detection**: We propose a state-of-the-art meta-learning technique that recommends outlier detection algorithms for a given dataset, based on a collection of historical datasets and prior experiments.
- **Open source code and demo** We open-source the code for LOTUS for researchers to use and reproduce our experiments. Our tools can be easily extended with additional algorithms and meta-data. We also provide a graphical interface for quick experimentation.
- **AutoML tool integration:** We provide an extension to the AutoML library GAMA [Gijsbers and Vanschoren, 2021], called GAMA-OD, that allows GAMA to solve outlier detection tasks using LOTUS. It includes an extensive model search space for outlier detection tasks, as well as tools to collect rich metadata on outlier detection performance across many datasets.

2 AutoML for Outlier Detection

AutoML for outlier detection is an extremely hard problem due to the lack of a ground truth optimization metric [Bahri *et al.*, 2022]. One can argue that the use of internal metrics such as Excess-Mass [Goix, 2016], Mass-Volume [Goix, 2016], and IREOS [Marques *et al.*, 2015] can be used instead. However, it has been shown that these internal metrics are computationally very expensive and do not scale well to large datasets [Ma *et al.*, 2021]. This makes it unfeasible to use these metrics in AutoML tools for most real-world scenarios, especially since AutoML algorithms perform many evaluations. In this work, we focus on tabular data, which has a considerably higher variance between datasets than image data, making it harder to find an optimal OD strategy. Tabular data is also common in industrial applications such as fraud detection [Cartella *et al.*, 2021] and network anomaly detection [Datta *et al.*, 2022; Liang *et al.*, 2022]. Table 1 summarizes how LOTUS compares to related AutoML approaches that either use meta-learning or OD. Of these, the most related is MetaOD [Zhao *et al.*, 2021], which is the current state-of-the-art technique for outlier detection on tabular data. PyODDS [Li *et al.*, 2020] is a related framework but it requires ground truth data to select specific OD techniques.

Technique	Meta-learning approach	Unsupervised Tasks	Use
AutoSklearn 2.0 [Feurer *et al.*, 2020]	Pipeline Portfolios	✗	warm-starting
FLAML [Wang *et al.*, 2021]	Built-in metafeatures	✗	warm-starting
MetaBu [Rakotoarison *et al.*, 2022]	Metafeatures (with labels) + FusedGW	✗	warm-starting
MetaOD [Zhao *et al.*, 2021]	Metafeatures + CF	Outlier detection only	model selection
LOTUS (Ours)	Preprocessing + GWLR	✓	model selection

Table 1: Comparison of different meta-learning AutoML frameworks

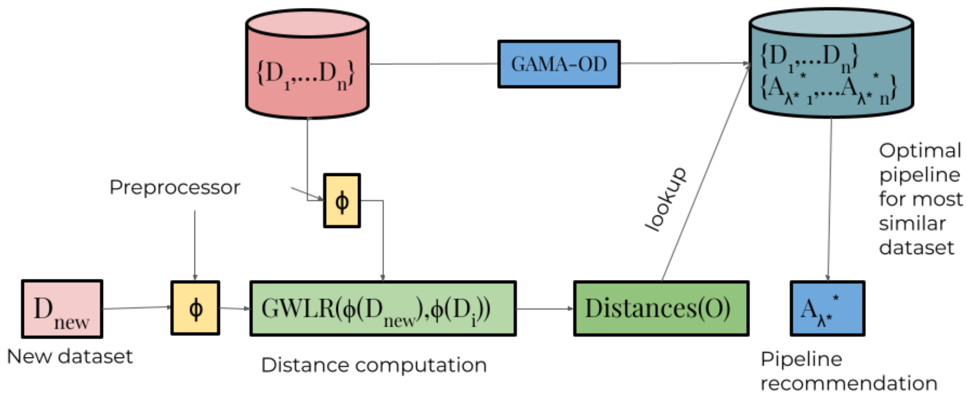

Figure 1: An overview of LOTUS. The top part corresponds to the meta-training phase, and the bottom part to the meta-testing phase.

3 Methodology

The LOTUS algorithm consists of a *meta-training* phase, which finds the optimized algorithms $A^*_{\lambda^*}$ for every prior dataset D_i, and a *meta-testing* phase that predicts the optimal algorithms for new, unseen tasks. The overall algorithm is illustrated in Figure 1, and the pseudo-code for each phase is shown in Algorithm 1 and 2, respectively.

3.1 Meta-training

Problem Statement: Given a new dataset without any labels, our meta-learner needs to select an optimal algorithm with associated hyperparameters from a collection of previously evaluated algorithms. Since we cannot further optimize the given model on the new dataset this is a *zero-shot model recommendation problem*, unless some (downstream) evaluation metric is available.

Formally, given a new unlabeled dataset $D_{new} = (X_{new})$, select a model $A^*_{\lambda^*} \in \mathcal{A}$ to employ on X_{new}, where $A^*_{\lambda^*}$ is the optimal model with tuned hyperparameters λ^* for the dataset D_i that is most similar to X_{new}.

Problem Formulation: For supervised tasks, this problem can be represented as a Combined Algorithm Selection and Hyperparameter optimization (CASH) problem [Thornton *et al.*, 2013], stated in equation 1, where $A^*_{\lambda^*}$ is the combination of the optimal learning algorithm from search space A with associated hyperparameter space Λ_A evaluated over k cross-validation folds of dataset $D = \{X, y\}$ with training and validation splits. L is our evaluation measure.

$$A^*_{\lambda^*} = \underset{\substack{\forall A^j \in \boldsymbol{A} \\ \forall \lambda \in \boldsymbol{\Lambda_A}}}{\operatorname{argmin}} \frac{1}{k} \sum_{f=1}^{k} L\left(A^j_\lambda, \{\boldsymbol{X}^{train}_f, \boldsymbol{y}^{train}_f\}, \{\boldsymbol{X}^{val}_f, \boldsymbol{y}^{val}_f\}\right) \tag{1}$$

The CASH problem from Equation 1 relies on the validation split to optimize for the optimal configuration. However, in unsupervised settings, such validation splits are not relevant. We run estimators on all unlabeled data, and use the ground truth labels only to evaluate them, as shown in Algorithm 1. Our modified CASH formulation to select the optimal unsupervised algorithm **with access to labels** is as follows:

$$A^*_{\lambda^*} = \underset{\substack{\forall A^j \in \boldsymbol{A} \\ \forall \lambda \in \boldsymbol{\Lambda_A}}}{\operatorname{argmin}} L\left(A^j_\lambda, \{\boldsymbol{X}\}\{\boldsymbol{y}\}\right) \tag{2}$$

To collect the necessary meta-data, we developed GAMA-OD, an extension to the popular AutoML tool GAMA [Gijsbers and Vanschoren, 2021].

3.2 Meta-testing

Our premise is that, if a prior dataset exists that is very similar to the new dataset, then its optimal algorithms will likely work well on the new dataset. We consider two datasets similar if they have the same underlying data distribution, which we measure using Optimal Transport [Peyré and Cuturi, 2019].

We first require a preprocessor ϕ, which is necessary to make input dataset compatible with the OT distance function. This preprocessing can involve the normalization of pixels in raw image data, encoders and scalers in tabular data. Next, we calculate the dataset similarity $\mathcal{O}$ based on Gromov Wasserstein [Peyré and Cuturi, 2019]:

$$\mathcal{O} = GW(\phi(D_a), \phi(D_b)) \tag{3}$$

Algorithm 1 Pseudocode for Meta-training

Inputs: $\mathcal{D}_{meta}, L, \boldsymbol{A}, \boldsymbol{\Lambda}_{\boldsymbol{A}}$

1: **while** $D_i \in \mathcal{D}_{meta}$ **do**
2: $\quad A^*_{\lambda^* i} \leftarrow \text{argmin}_{\substack{\forall A^j \in \boldsymbol{A} \\ \forall \lambda \in \boldsymbol{\Lambda}_{\boldsymbol{A}}}} L\left(A^j_\lambda, \{\boldsymbol{X}\}\{\boldsymbol{y}\}\right)$
3: $\quad \mathcal{A} \leftarrow A^*_{\lambda^* i}$
4: **end while**

Algorithm 2 Pseudocode for LOTUS (meta-testing)

Inputs: $D_{new}, \mathcal{D}_{meta}, \mathcal{A}$

1: **while** $D_i \in \mathcal{D}_{meta}$ **do**
2: $\quad \mathcal{O}_i \leftarrow GWLR(\phi(D_{new}, D_i))$ {Distance calculation}
3: **end while**
4: $s \leftarrow \text{argmin}\{\mathcal{O}_1, ..., \mathcal{O}_n\}$ {Retrieval of most similar dataset}
5: $A^*_{\lambda^*_{new}} \leftarrow A^*_{\lambda^*_s}$ {Model Selection}

We adopt the Low-Rank Gromov-Wasserstein distance [Scetbon and Cuturi, 2022] on these preprocessed datasets for faster computation, as summarized in Equation 4, where r is the selected rank hyperparameter for distance computation.

$$\mathcal{O} = \text{GW-LR}^{(r)}(\phi(D_a), \phi(D_b)) \tag{4}$$

The most similar prior dataset $D_{similar} \in \mathcal{D}_{meta}$ is the dataset with the smallest distance to the new dataset D_{new}.

LOTUS then assigns the optimal configuration from $\mathcal{A}$: $A^*_{\lambda^*_{new}} = A^*_{\lambda^*_s}$ where $A^*_{\lambda^*_s}$ is predicted as the optimal configuration for D_{new}, as also shown in Algorithm 2. The Python code for using LOTUS is shown in Listing 1.

Listing 1 Example code for using LOTUS.

```python
from lotus import LotusMetaData
from lotus import LotusModel

md = LotusMetaData(
    data_list, 'accuracy',
    dataloader = dataloader,
    out = 'csv')
md.create_lotus_metadata()
dataset = new_dataset

model = LotusModel(
    new_dataset=dataset,
    meta_data_obj=md,
    distance = 'gwlr',
    preprocessing = 'ica')

best_model, distance, score =
model.find_model()
```

Estimator	p**(LOTUS)**	p**(rope)**	p**(Estimator)**
MetaOD	**0.740**	0.074	0.186
ABOD	**1.0**	0.0	0.0
OCSVM	**1.0**	0.0	0.0
LODA	**1.0**	0.0	0.0
KNN	**1.0**	0.0	0.0
HBOS	**999.82**$\cdot 10^{-3}$	0.0	$0.18 \cdot 10^{-3}$
IForest	**999.54**$\cdot 10^{-3}$	0.0	$0.46 \cdot 10^{-3}$
COF	**1.0**	0.0	0.0
LOF	**1.0**	0.0	0.0

Table 2: Rope testing results with LOTUS vs PyOD baselines with rope=1% (Higher is better)

4 Experimental Setup

To evaluate LOTUS, we use ADBench [Han *et al.*, 2022] which is a comprehensive tabular anomaly detection benchmark on 57 datasets. GAMA-OD uses an asynchronous evolutionary algorithm to iterate over the search space and return the optimal pipeline. We use the area under the ROC curve (AUC) as the optimization metric L during the search phase. We use standard anomaly detection algorithms from PyOD [Zhao *et al.*, 2019], which is the largest outlier detection library in Python. We use these algorithms with default hyperparameters as additional baselines.

5 Results and Discussion

We use the Bayesian Wilcoxon signed-rank test (or ROPE test [Benavoli *et al.*, 2017; Benavoli *et al.*, 2014]) to analyze the results of our experiments. We first compare the results with the state-of-the-art(MetaOD) and then other baselines.

LOTUS vs MetaOD (State-of-the-Art)

We show the pairwise comparison of LOTUS and MetaOD using the ROPE test in Table 2. We find that, based on experiments, there is a 74.0 % probability($p(LOTUS) = 0.74$) that LOTUS will outperform MetaOD. $p(LOTUS) > p(MetaOD)$ shows that LOTUS is more robust.

LOTUS vs PyOD Baselines

The results of the ROPE test comparing LOTUS with individual outlier detection techniques are summarized in Table 2. LOTUS proves to be significantly better than all other techniques, with default parameters. In this case $p(LOTUS) >> p(Estimator)$.

6 Conclusion

We propose an easy-to-use zero-shot-model-recommendation AutoML tool for outlier detection which uses Gromov-Wasserstein distances to find the optimal outlier detection algorithms on a given task, based on previously learned meta-knowledge. We show via experiments and analyses that our approach is robust and outperforms current state-of-the-art algorithms. In future work, we will extend this work to other unsupervised scenarios such as clustering, covariance estimation, and distance metric learning.

References

[Bahri *et al.*, 2022] Maroua Bahri, Flavia Salutari, Andrian Putina, and Mauro Sozio. AutoML: state of the art with a focus on anomaly detection, challenges, and research directions. *International Journal of Data Science and Analytics*, February 2022.

[Benavoli *et al.*, 2014] Alessio Benavoli, Giorgio Corani, Francesca Mangili, Marco Zaffalon, and Fabrizio Ruggeri. A bayesian wilcoxon signed-rank test based on the dirichlet process. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1026–1034, Bejing, China, 22–24 Jun 2014. PMLR.

[Benavoli *et al.*, 2017] Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research*, 18(77):1–36, 2017.

[Cartella *et al.*, 2021] Francesco Cartella, Orlando Anunciação, Yuki Funabiki, Daisuke Yamaguchi, Toru Akishita, and Olivier Elshocht. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *ArXiv*, abs/2101.08030, 2021.

[Datta *et al.*, 2022] Debanjan Datta, S. Muthiah, John Simeone, Amelia Meadows, and Naren Ramakrishnan. Scrutinizing shipment records to thwart illegal timber trade. *ArXiv*, abs/2208.00493, 2022.

[Feurer *et al.*, 2015] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[Feurer *et al.*, 2020] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: Hands-free automl via meta-learning. *arXiv:2007.04074 [cs.LG]*, 2020.

[Gijsbers and Vanschoren, 2021] Pieter Gijsbers and Joaquin Vanschoren. Gama: A general automated machine learning assistant. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track*, 2021.

[Goix, 2016] Nicolas Goix. How to evaluate the quality of unsupervised anomaly detection algorithms?, 2016.

[Han *et al.*, 2022] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. ADBench: Anomaly detection benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[Hutter *et al.*, 2019] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. Automated machine learning: Methods, systems, challenges. *Automated Machine Learning*, 2019.

[Li *et al.*, 2020] Yuening Li, Daochen Zha, Na Zou, and Xia Hu. Pyodds: An end-to-end outlier detection system with automated machine learning. *Companion Proceedings of the Web Conference 2020*, 2020.

[Liang *et al.*, 2022] Dong Liang, Jun Wang, Wenping Zhang, Yuqi Liu, Lei Wang, and Xiaoyong Zhao. Tabular data anomaly detection based on density peak clustering algorithm. *2022 International Conference on Big Data, Information and Computer Network (BDICN)*, pages 16–21, 2022.

[Ma *et al.*, 2021] Martin Q. Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. A large-scale study on unsupervised outlier model selection: Do internal strategies suffice? *CoRR*, abs/2104.01422, 2021.

[Marques *et al.*, 2015] Henrique O. Marques, Ricardo J. G. B. Campello, Arthur Zimek, and Jorg Sander. On the internal evaluation of unsupervised outlier detection. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, SSDBM '15, New York, NY, USA, 2015. Association for Computing Machinery.

[Peyré and Cuturi, 2019] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11:355–607, 2019.

[Rakotoarison *et al.*, 2022] Herilalaina Rakotoarison, Louisot Milijaona, Andry RASOANAIVO, Michele Sebag, and Marc Schoenauer. Learning meta-features for autoML. In *International Conference on Learning Representations*, 2022.

[Scetbon and Cuturi, 2022] Meyer Scetbon and Marco Cuturi. Low-rank optimal transport: Approximation, statistics and debiasing. *NeurIPS 2022*, abs/2205.12365, 2022.

[Thornton *et al.*, 2013] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 847–855, 2013.

[Vanschoren, 2018] Joaquin Vanschoren. Meta-learning: A survey. *ArXiv*, abs/1810.03548, 2018.

[Wang *et al.*, 2021] Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. Flaml: A fast and lightweight automl library. In *MLSys*, 2021.

[Zhao *et al.*, 2019] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *J. Mach. Learn. Res.*, 20:96:1–96:7, 2019.

[Zhao *et al.*, 2021] Yue Zhao, Ryan Rossi, and Leman Akoglu. Automatic unsupervised outlier model selection. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4489–4502. Curran Associates, Inc., 2021.