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; Schetbon 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.
<table>
<thead>
<tr>
<th>Technique</th>
<th>Meta-learning approach</th>
<th>Unsupervised Tasks</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoSklearn 2.0 [Feurer et al., 2020]</td>
<td>Pipeline Portfolios</td>
<td>✓</td>
<td>warm-starting</td>
</tr>
<tr>
<td>FLAML [Wang et al., 2021]</td>
<td>Built-in metafeatures</td>
<td>✓</td>
<td>warm-starting</td>
</tr>
<tr>
<td>MetaBu [Rakotoarison et al., 2022]</td>
<td>Metafeatures (with labels) + FusedGW</td>
<td>✓</td>
<td>warm-starting</td>
</tr>
<tr>
<td>MetaOD [Zhao et al., 2021]</td>
<td>Metafeatures + CF</td>
<td></td>
<td>model selection</td>
</tr>
<tr>
<td>LOTUS (Ours)</td>
<td>Preprocessing + GWLR</td>
<td>✓</td>
<td>model selection</td>
</tr>
</tbody>
</table>

Table 1: Comparison of different meta-learning AutoML frameworks

![Diagram of LOTUS](image)

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 $A^*_{\lambda^*} \in A$ to employ on $X_{new}$, where $A^*_{\lambda^*}$ is the optimal model with tuned hyperparameters $\lambda^*$ 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 $\Lambda_A$ evaluated over $k$ cross-validation folds of dataset $D = \{X, y\}$ with training and validation splits. $L$ is our evaluation metric.

$$A^*_{\lambda^*} = \arg\min_{\forall A^*_{\lambda^*} \in A \forall \lambda^* \in \Lambda_A} \sum_{f=1}^{k} L \left( A^*_{\lambda^*}, \{X^f_{train}, y^f_{train}\}, \{X^f_{val}, y^f_{val}\} \right)$$

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

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 $O$ based on Gromov Wasserstein [Peyré and Cuturi, 2019]:

$$O = GW(\phi(D_a), \phi(D_b))$$

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:

$$A^*_{\lambda^*} = \arg\min_{\forall A^*_{\lambda^*} \in A \forall \lambda^* \in \Lambda_A} L \left( A^*_{\lambda^*}, \{X\} \{y\} \right)$$

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 $\phi$, 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 $O$ based on Gromov Wasserstein [Peyré and Cuturi, 2019]:

$$O = GW(\phi(D_a), \phi(D_b))$$
Algorithm 1 Pseudocode for Meta-training

Inputs: \(D_{\text{meta}}, L, A, \Lambda, \lambda\)
1: while \(D_i \in D_{\text{meta}}\) do
2: \(A_{\lambda, i}^* \leftarrow \arg\min_{A^i \in A} \mathcal{L}(A^i, \{X\} \{y\})\)
3: \(A \leftarrow A_{\lambda, i}^*\)
4: end while

Algorithm 2 Pseudocode for LOTUS (meta-testing)

Inputs: \(D_{\text{new}}, D_{\text{meta}}, A\)
1: while \(D_i \in D_{\text{meta}}\) do
2: \(\mathcal{O} \leftarrow \text{GW-LR}(\phi(D_{\text{new}}, D_i))\) \{Distance calculation\}
3: end while
4: \(s \leftarrow \arg\min\{\mathcal{O}_1, ..., \mathcal{O}_n\}\) \{Retrieval of most similar dataset\}
5: \(A_{\lambda, \text{new}}^* \leftarrow A_{\lambda, s}^*\) \{Model Selection\}

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= 'gwir',
    preprocessing = 'ica')

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

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.

\[
\mathcal{O} = \text{GW-LR}(r)(\phi(D_a), \phi(D_b))
\] (4)

The most similar prior dataset \(D_{\text{similar}} \in D_{\text{meta}}\) is the dataset with the smallest distance to the new dataset \(D_{\text{new}}\). LOTUS then assigns the optimal configuration from \(A\): \(A_{\lambda, \text{new}}^* = A_{\lambda, s}^*\) where \(A_{\lambda, s}^*\) is predicted as the optimal configuration for \(D_{\text{new}}\), as also shown in Algorithm 2. The Python code for using LOTUS is shown in Listing 1.

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

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(p(\text{LOTUS}))</th>
<th>(p(\text{rope}))</th>
<th>(p(\text{Estimator}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetaOD</td>
<td>0.740</td>
<td>0.074</td>
<td>0.186</td>
</tr>
<tr>
<td>ABOD</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>OCSVM</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>LODA</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>KNN</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>HBOS</td>
<td>999.82 (10^{-3})</td>
<td>0.0</td>
<td>0.18(10^{-3})</td>
</tr>
<tr>
<td>iForest</td>
<td>999.54 (10^{-3})</td>
<td>0.0</td>
<td>0.46(10^{-3})</td>
</tr>
<tr>
<td>COF</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>LOF</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

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\((p(\text{LOTUS}) = 0.74)\) that LOTUS will outperform MetaOD. \(p(\text{LOTUS}) > p(\text{MetaOD})\) 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 \(p(\text{LOTUS}) >> p(\text{Estimator})\).

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 metadata. 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


