
MindWare

Release beta

Thomas (Yang) Li

Feb 21, 2023

TABLE OF CONTENTS

1	Overview	3
2	Installation Guide	5
2.1	1 System Requirements	5
2.2	2 Preparations before Installation	5
2.3	3 Install MindWare	5
2.4	4 Installation for Advanced Usage (Optional)	6
2.5	5 Trouble Shooting	6
3	Quick Start	9
3.1	Data Preparation	9
3.2	Optimization	9
4	Examples	11
4.1	Single-Objective Black-box Optimization	11
5	Research and Publications	13
5.1	Related Research	13
6	Change Logs	15

Coming soon!

OVERVIEW

To be filled

INSTALLATION GUIDE

2.1 1 System Requirements

Installation Requirements:

- Python \geq 3.6 (3.7 is recommended!)

Supported Systems:

- Linux (Ubuntu, ...)
- macOS
- Windows

2.2 2 Preparations before Installation

We **STRONGLY** suggest you to create a Python environment via [Anaconda](#):

```
conda create -n mindware python=3.7
conda activate mindware
```

Then we recommend you to update your `pip` and `setuptools` as follows:

```
pip install pip setuptools --upgrade
```

2.3 3 Install MindWare

2.3.1 3.1 Installation from PyPI

To install MindWare from PyPI, simply run the following command:

```
pip install mindware
```

2.3.2 3.2 Manual Installation from Source

To install MindWare using the source code, please run the following commands:

```
git clone https://github.com/thomas-young-2013/mindware.git && cd mindware
cat requirements/main.txt | xargs -n 1 -L 1 pip install
python setup.py install
```

2.3.3 3.3 Test for Installation (TODO)

You can run the following code to test your installation:

```
from mindware import run_test

if __name__ == '__main__':
    run_test()
```

If successful, you will receive the following message:

```
===== Congratulations! All trials succeeded. =====
```

If you encountered any problem during installation, please refer to the **Trouble Shooting** section.

2.4 4 Installation for Advanced Usage (Optional)

To use advanced features such as `pyrfr` (probabilistic random forest) surrogate in HPO, please refer to [Pyrfr Installation Guide](#) to install `pyrfr`.

2.5 5 Trouble Shooting

If you encounter problems not listed below, please [File an issue](#) on GitHub or email us via liyang.cs@pku.edu.cn.

If you cannot install openbox correctly, please refer to [OpenBox Installation Guide](#).

2.5.1 Windows

- ‘Error: [WinError 5] Access denied’. Please open the command prompt with administrative privileges or append `--user` to the command line.
- ‘ERROR: Failed building wheel for ConfigSpace’. Please refer to [tips](#).

2.5.2 macOS

- For macOS users who have trouble installing pyrfr, please refer to [tips](#).
- For macOS users who have trouble building scikit-learn, this [documentation](#) might help.

QUICK START

This tutorial helps you run your first example with **MindWare**.

3.1 Data Preparation

First, **prepare data** for the end-to-end AutoML system. Here we use the iris dataset from sklearn as an example.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1,
→ stratify=y)
```

After loading and splitting the dataset, wrap the data by **DataManager**.

```
from mindware.utils.data_manager import DataManager

dm = DataManager(X_train, y_train)
train_data = dm.get_data_node(X_train, y_train)
test_data = dm.get_data_node(X_test, y_test)
```

3.2 Optimization

MindWare provides an efficient way to complete machine learning task in an end-to-end manner. In this example, we use **Classifier** for the classification task. Please specify **time_limit** to set the time budget for optimization. Then simply call **Classifier.fit** and the system will perform feature engineering, model selection, hyper-parameter optimization and model ensemble automatically. For large search spaces, the system employs search space decomposition to accelerate optimization.

```
from mindware.estimators import Classifier

clf = Classifier(time_limit=3600)
clf.fit(train_data)
```

After optimization, call **Classifier.predict** to get predictions of test dataset, made by the best searched feature engineering method, ML model (might be ensemble model) and its hyper-parameters.

```
pred = clf.predict(test_data)
```

(todo: result and visualization)

EXAMPLES

4.1 Single-Objective Black-box Optimization

To be filled

RESEARCH AND PUBLICATIONS

5.1 Related Research

1. **VolcanoML: Speeding up End-to-End AutoML via Scalable Search Space Decomposition** Yang Li, Yu Shen, Wentao Zhang, Jiawei Jiang, Bolin Ding, Yaliang Li, Jingren Zhou, Zhi Yang, Wentao Wu, Ce Zhang and Bin Cui International Conference on Very Large Data Bases (VLDB 2021).
2. **Efficient Automatic CASH via Rising Bandits** Yang Li, Jiawei Jiang, Jinyang Gao, Yingxia Shao, Ce Zhang and Bin Cui Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020). <https://ojs.aaai.org/index.php/AAAI/article/view/5910>
3. **MFES-HB: Efficient Hyperband with Multi-Fidelity Quality Measurements** Yang Li, Yu Shen, Jiawei Jiang, Jinyang Gao, Ce Zhang and Bin Cui Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2021). <https://arxiv.org/abs/2012.03011>
4. **MindWare: A Generalized Black-box Optimization Service** Yang Li, Yu Shen, Wentao Zhang, Yuanwei Chen, Huaijun Jiang, Mingchao Liu, Jiawei Jiang, Jinyang Gao, Wentao Wu, Zhi Yang, Ce Zhang and Bin Cui ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD 2021). <https://arxiv.org/abs/2106.00421>

CHANGE LOGS

Coming Soon!