

---

# **advisor Documentation**

**tobe**

**Oct 18, 2018**



---

## Contents:

---

<b>1</b>	<b>Advisor</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Supported Algorithms . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Pip . . . . .	3
2.2	From Source . . . . .	3
2.3	Docker . . . . .	3
2.4	Docker Compose . . . . .	3
2.5	Kubernetes . . . . .	4
<b>3</b>	<b>Quick Start</b>	<b>5</b>
<b>4</b>	<b>Server</b>	<b>7</b>
4.1	Command-line . . . . .	7
4.2	Docker . . . . .	7
4.3	Docker Compose . . . . .	7
4.4	Kubernetes . . . . .	7
4.5	From Source . . . . .	8
<b>5</b>	<b>Command Line Interface</b>	<b>9</b>
5.1	Start Server . . . . .	9
5.2	Stop Server . . . . .	9
5.3	Submit Job . . . . .	9
5.4	List Study . . . . .	9
5.5	Describe Study . . . . .	9
5.6	List Trials . . . . .	10
<b>6</b>	<b>SDK</b>	<b>11</b>
6.1	Create Client . . . . .	11
6.2	Create Study . . . . .	11
6.3	Get Study . . . . .	12
6.4	Get Trials . . . . .	12
6.5	Generate Parameters . . . . .	12
6.6	Run Training . . . . .	12
6.7	Complete Trial . . . . .	13

<b>7</b>	<b>Configuration File</b>	<b>15</b>
7.1	YAML Example . . . . .	15
7.2	JSON Example . . . . .	15

### 1.1 Introduction

Advisor is the hyper parameters tuning system for black box optimization.

It is the open-source implementation of [Google Vizier](#) with these features.

- Easy to use with API, SDK, WEB and CLI
- Support abstractions of Study and Trial
- Included search and early stop algorithms
- Recommend parameters with trained model
- Same programming interfaces as Google Vizier
- Command-line tool just like Microsoft NNI.

### 1.2 Supported Algorithms

- [x] Grid Search
- [x] Random Search
- [x] Bayesian Optimization
- [x] TPE(Hyperopt)
- [x] Random Search(Hyperopt)
- [x] Simulate Anneal(Hyperopt)
- [x] Quasi Random(Chocolate)
- [x] Grid Search(Chocolate)
- [x] Random Search(Chocolate)

- Bayes(Chocolate)
- CMAES(Chocolate)
- MOCMAES(Chocolate)
- SMAC Algorithm
- Early Stop First Trial Algorithm
- Early Stop Descending Algorithm
- Performance Curve Stop Algorithm

### 2.1 Pip

```
pip install advisor
```

### 2.2 From Source

```
git clone git@github.com:tobegit3hub/advisor.git  
cd ./advisor/advisor_client/  
python ./setup.py install
```

### 2.3 Docker

```
docker run -d -p 8000:8000 tobegit3hub/advisor
```

### 2.4 Docker Compose

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/docker-compose.yml  
docker-compose up -d
```

## 2.5 Kubernetes

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/kubernetes_advisor.  
↪yaml  
kubectl create -f ./kubernetes_advisor.yaml
```



## CHAPTER 3

---

### Quick Start

---

Install with pip.

```
pip install advisor
```

Start the server.

```
advisor_admin server start
```

Go to <http://127.0.0.1:8000> in the browser.

Submit tuning jobs.

```
git clone --depth 1 https://github.com/tobegit3hub/advisor.git && cd ./advisor/  
advisor run -f ./advisor_client/examples/python_function/config.json
```

Get result of jobs.

```
advisor study describe -s demo
```



### 4.1 Command-line

```
advisor_admin server start
```

### 4.2 Docker

```
docker run -d -p 8000:8000 tobegit3hub/advisor
```

### 4.3 Docker Compose

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/docker-compose.yml  
docker-compose up -d
```

### 4.4 Kubernetes

```
wget https://raw.githubusercontent.com/tobegit3hub/advisor/master/kubernetes_advisor.  
→yaml  
kubect1 create -f ./kubernetes_advisor.yaml
```

## 4.5 From Source

```
git clone --depth 1 https://github.com/tobegit3hub/advisor.git && cd ./advisor/  
pip install -r ./requirements.txt  
./manage.py migrate  
./manage.py runserver 0.0.0.0:8000
```

---

## Command Line Interface

---

### 5.1 Start Server

```
advisor_admin server start
```

### 5.2 Stop Server

```
advisor_admin server stop
```

### 5.3 Submit Job

```
advisor run -f ./advisor_client/examples/python_function/config.json
```

### 5.4 List Study

```
advisor study list
```

### 5.5 Describe Study

```
advisor study describe -s demo
```

## 5.6 List Trials

```
advisor trials list
```

## 6.1 Create Client

```
client = AdvisorClient()
```

## 6.2 Create Study

```
study_configuration = {
  "goal":
  "MINIMIZE",
  "randomInitTrials":
  1,
  "maxTrials":
  5,
  "maxParallelTrials":
  1,
  "params": [
    {
      "parameterName": "gamma",
      "type": "DOUBLE",
      "minValue": 0.001,
      "maxValue": 0.01,
      "feasiblePoints": "",
      "scalingType": "LINEAR"
    },
    {
      "parameterName": "C",
      "type": "DOUBLE",
      "minValue": 0.5,
      "maxValue": 1.0,
      "feasiblePoints": "",

```

(continues on next page)

```
        "scalingType": "LINEAR"
    },
    {
        "parameterName": "kernel",
        "type": "CATEGORICAL",
        "minValue": 0,
        "maxValue": 0,
        "feasiblePoints": "linear, poly, rbf, sigmoid, precomputed",
        "scalingType": "LINEAR"
    },
    {
        "parameterName": "coef0",
        "type": "DOUBLE",
        "minValue": 0.0,
        "maxValue": 0.5,
        "feasiblePoints": "",
        "scalingType": "LINEAR"
    },
]
}
study = client.create_study("Study", study_configuration,
                           "BayesianOptimization")
```

## 6.3 Get Study

```
study = client.get_study_by_id(6)
```

## 6.4 Get Trials

```
trials = client.get_suggestions(study.id, 3)
```

## 6.5 Generate Parameters

```
parameter_value_dicts = []
for trial in trials:
    parameter_value_dict = json.loads(trial.parameter_values)
    print("The suggested parameters: {}".format(parameter_value_dict))
    parameter_value_dicts.append(parameter_value_dict)
```

## 6.6 Run Training

```
metrics = []
for i in range(len(trials)):
    metric = train_function(**parameter_value_dicts[i])
    metrics.append(metric)
```



## 6.7 Complete Trial

```
for i in range(len(trials)):
    trial = trials[i]
    client.complete_trial_with_one_metric(trial, metrics[i])
is_done = client.is_study_done(study.id)
best_trial = client.get_best_trial(study.id)
print("The study: {}, best trial: {}".format(study, best_trial))
```



---

## Configuration File

---

### 7.1 YAML Example

```
name: "demo"
algorithm: "BayesianOptimization"
trialNumber: 10
path: "./advisor_client/examples/python_function/"
command: "./min_function.py"
search_space:
  goal: "MINIMIZE"
  randomInitTrials: 3
  params:
    - parameterName: "x"
      type: "DOUBLE"
      minValue: -10.0
      maxValue: 10.0
```

### 7.2 JSON Example

```
{
  "name": "demo",
  "algorithm": "BayesianOptimization",
  "trialNumber": 10,
  "concurrency": 1,
  "path": "./advisor_client/examples/python_function/",
  "command": "./min_function.py",
  "search_space": {
    "goal": "MINIMIZE",
    "randomInitTrials": 3,
    "params": [
      {
```

(continues on next page)

(continued from previous page)

```
        "parameterName": "x",
        "type": "DOUBLE",
        "minValue": -10.0,
        "maxValue": 10.0,
        "scalingType": "LINEAR"
    ]
}
```