



HYPEROPT



# Hyperopt

- Docs: <http://hyperopt.github.io/hyperopt/>
- Github: <https://github.com/hyperopt/hyperopt>



# Hyperopt – Search algorithms

- Random Search
  - `rand.suggest`
- Annealing → a SMBO with GP alternative
  - `anneal.suggest`
- Tree-structured Parzen Estimators (TPE)
  - `tpe.suggest`

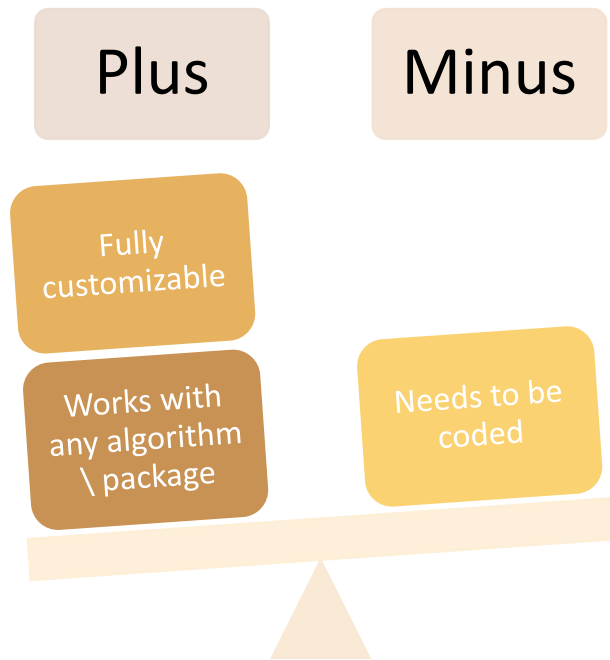
# Hyperopt – Search algorithms

The search algorithm is selected within the minimization driver - fmin

```
search = fmin(  
    fn=objective,  
    space=param_grid,  
    max_evals=50,  
    rstate=np.random.RandomState(42),  
    algo=tpe.suggest, # tpe  
    # algo=anneal.suggest,  
    # algo=rand.suggest,  
)
```

# Hyperopt – objective function

Objective function created by user



# Hyperopt – Sklearn

Alternative library to work with Scikit-learn classes

- <http://hyperopt.github.io/hyperopt-sklearn/>
- Thin documentation
- Haven't seen it widely adopted
- We won't cover it in the course

# Hyperopt – Search Space

Built-in module to create hyperparameter sample spaces - **hp**

- Samples Reals, Integers and Categories
- Extremely versatile, allows multiple distributions and combinations of them
- Accepts lists, tuples and dictionaries
- Accepts nested hyperparameter spaces

# Search Space Configuration

- Accepts lists, tuples and dictionaries

```
from hyperopt import hp

list_space = [
    hp.uniform('a', 0, 1),
    hp.loguniform('b', 0, 1)]
tuple_space = (
    hp.uniform('a', 0, 1),
    hp.loguniform('b', 0, 1))

dict_space = {
    'a': hp.uniform('a', 0, 1),
    'b': hp.loguniform('b', 0, 1)}
```



# Search Space – nested spaces

- Accepts nested hyperparameter spaces

```
nested_space = [  
    [ {'case': 1, 'a': hp.uniform('a', 0, 1)},  
      {'case': 2, 'b': hp.loguniform('b', 0, 1)} ],  
    'extra literal string',  
    hp.randint('r', 10) ]
```

# Hyperopt – Acquisition Function

## Built-in Acquisition functions

- Expected Improvement (EI)
- EI evaluated at binomial distributions of the input space for discrete and categorical hyperparameters
- EI evaluated with CMA-ES for the continuous hyperparameter space

# Hyperopt – Analysis: Trials

- Trials object allows you to store as much information as you like
- Built-in functions for plotting
  - Not very useful
  - No documentation

# Parallelization - MongoDB

- Allows search in parallel utilizing MongoDB
  - ✓ Reduced efficiency per (single) evaluation
  - ✓ Increased overall efficiency by saving time

```
from hyperopt import fmin
from hyperopt.mongo import MongoTrials
trials = MongoTrials('mongo://host:port/fmin_db/')
best = fmin(q, space, trials=trials)
```

# THANK YOU

[www.trainindata.com](http://www.trainindata.com)