Solution for Automated Hyperparameter Optimization Challenge

DAIR Lab, Peking University

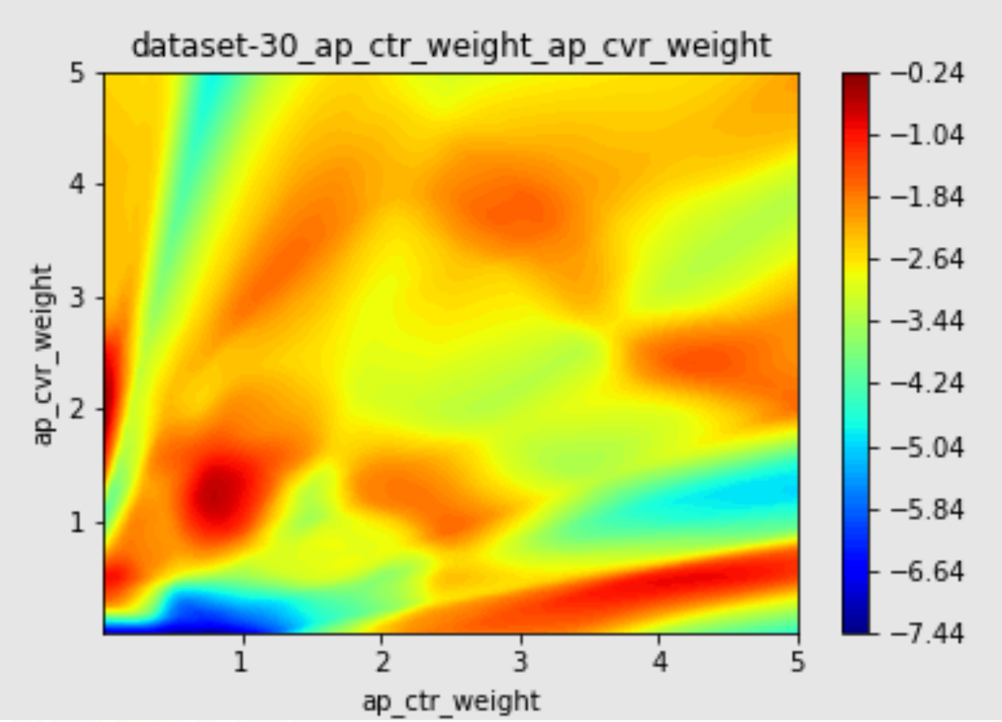
2021.10.30

北京大學
PEKING UNIVERSITY

PKU-DAIR

Preliminary

- Problem Analysis
  - Black-box optimization
  - All hyperparameters are numerical
  - The ground truth is continuous and stable



Bayesian Optimization (BO)

- Configuration – A choice of hyperparameters from their value ranges
- Observations – The collection of configurations and their performance
- Pipeline
  - Fit a probabilistic surrogate model based on observations
  - Choose the next configuration to evaluate based on the surrogate
  - Evaluate the configuration
  - Augment the observations

$[\mathbf{R}, \boldsymbol{\theta}_{inc}] \leftarrow$ *Initialize*$(\boldsymbol{\Theta}, \Pi)$;
repeat
  $[\mathcal{M}, t_{fit}] \leftarrow$ *FitModel*$(\mathbf{R})$;
  $[\vec{\boldsymbol{\Theta}}_{new}, t_{select}] \leftarrow$ *SelectConfigurations*$(\mathcal{M}, \boldsymbol{\theta}_{inc}, \boldsymbol{\Theta})$;
  $[\mathbf{R}, \boldsymbol{\theta}_{inc}] \leftarrow$ *Intensify*$(\vec{\boldsymbol{\Theta}}_{new}, \boldsymbol{\theta}_{inc}, \mathcal{M}, \mathbf{R}, t_{fit} + t_{select}, \Pi, \hat{c})$;
until *total time budget for configuration exhausted*;
return $\boldsymbol{\theta}_{inc}$;

Image: F., Hutter et al. Sequential Model-Based Optimization for General Algorithm Configuration, 2011

BO Components

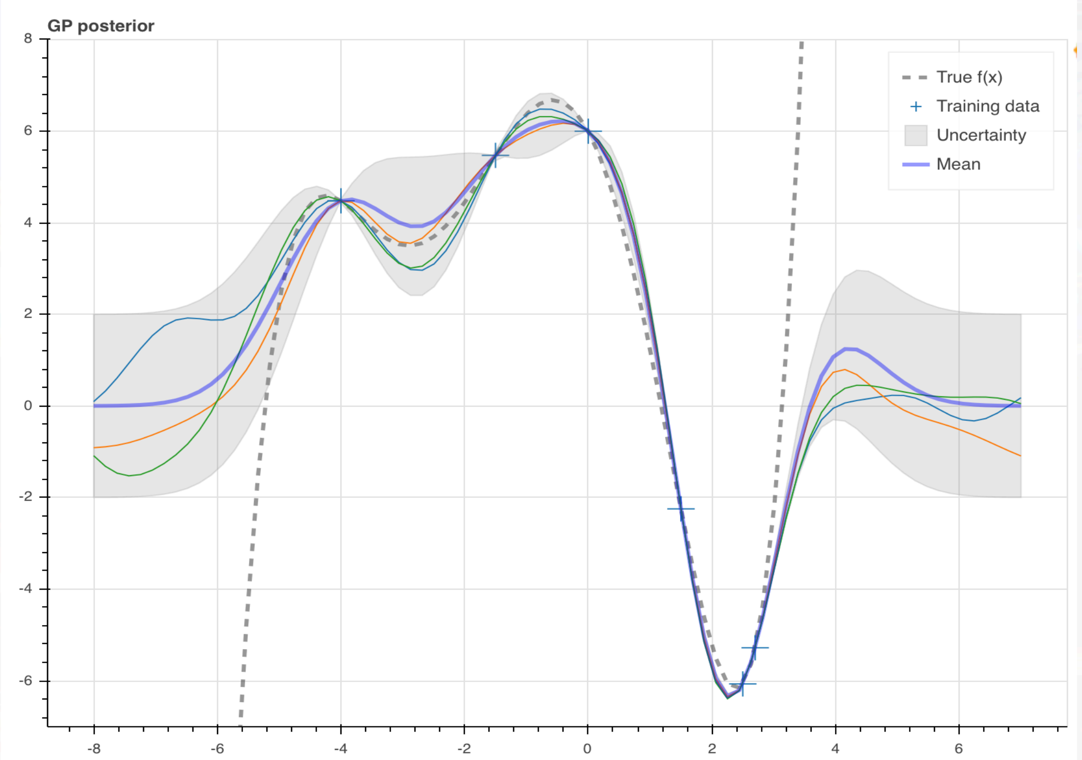
- Surrogate – Gaussian Process (GP)

- The posterior given x_* is a normal distribution,

$$\bar{y}_* = K_* K^{-1} \mathbf{y},$$
$$\mathrm{var}(y_*) = K_{**} - K_* K^{-1} K_*^{\mathrm{T}}.$$

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

$$K_* = \begin{bmatrix} k(x_*, x_1) & k(x_*, x_2) & \cdots & k(x_*, x_n) \end{bmatrix} \qquad K_{**} = k(x_*, x_*).$$



BO Components

- Acquisition Function – Expected Improvement (EI)

$$E(x) = E[max(f_{min} - f(x), 0)]$$

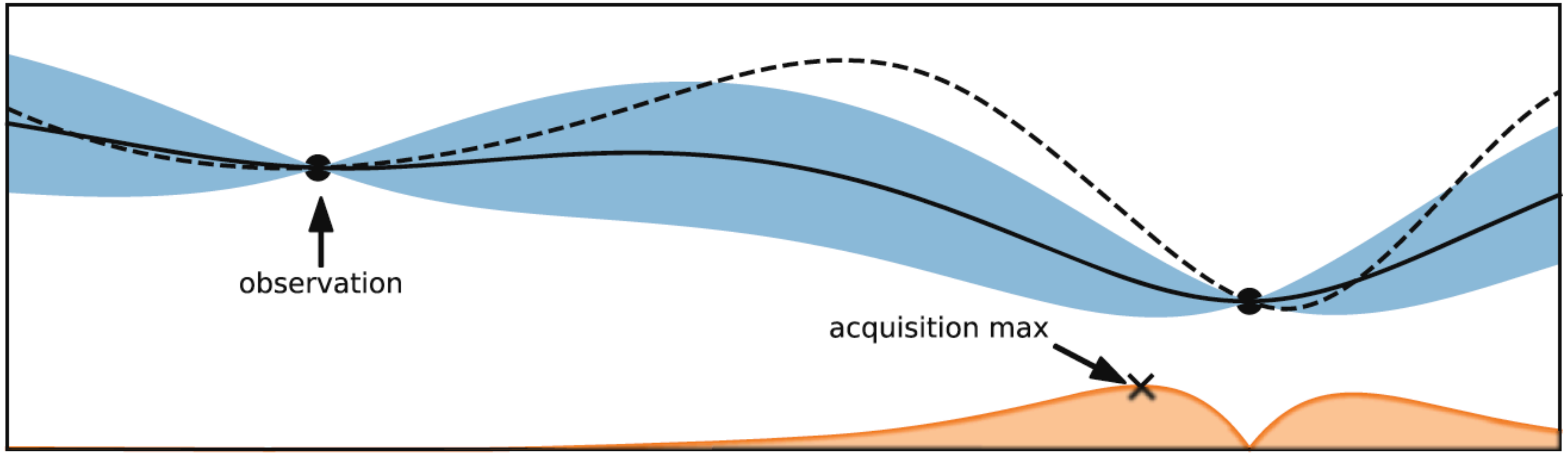


Image: https://zhuanlan.zhihu.com/p/143108146

BO Components

- Acquisition Optimizer – L-BFGS-B with local and random start
  - L-BFGS-B – A well-known quasi-Newton algorithm for optimization
  - Initial points – Select 10 start points with the best EI value via:
    - Monte Carlo sampling from the entire space
    - One-step mutation from the best observed configurations

Summary

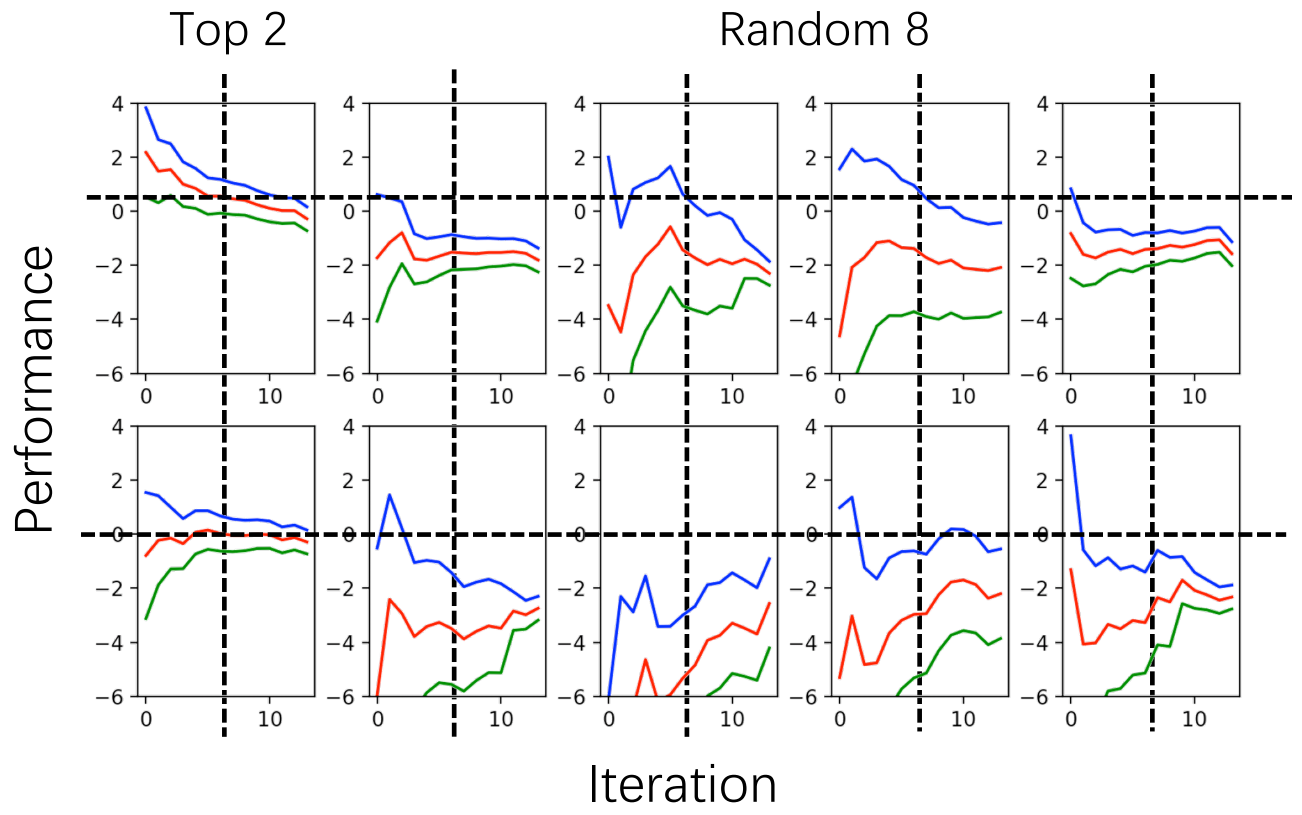
- Algorithm: Bayesian Optimization
- Surrogate: Gaussian Process
- Acquisition Function: EI
- Acquisition Optimizer: L-BFGS with local and random start

PEKING UNIVERSITY

PKU-DAIR

Final

- Less evaluation budget
- Multi-fidelity results with confidence bounds



Combine with BO

- At the 7-th iteration, early-stop a configuration if it is worse than half of the observed configurations

- Impute the early-stopped configuration with the median of observed performance

PKU-DAIR

Conclusion

- The preliminary solution has already been integrated into OpenBox

```python
from openbox import SyncBatchAdvisor, Observation

config_space = ...


def evaluate_configs(config):
    ...


advisor = SyncBatchAdvisor(
    config_space, batch_size=5, batch_strategy='default',
    initial_trials=10, init_strategy='random_explore_first',
    rand_prob=0.1, surrogate_type='gp', acq_type='ei',
    acq_optimizer_type='random_scipy', task_id='thpo'
)

for i in range(20):
    # ask
    config_list = advisor.get_suggestions()
    # evaluate
    reward_list = evaluate_configs(config_list)
    # tell
    for config, reward in zip(config_list, reward_list):
        objs = [-reward]  # OpenBox minimizes the objective
        observation = Observation(config=config, objs=objs)
        advisor.update_observation(observation)
```

OpenBox

北京大学 PEKING UNIVERSITY
PKU-DAIR

Q&A