Lecture 5
Hyper-parameter Optimization

Lisheng Sun-Hosoya, Feb 4
Successes of Machine learning

NLP

Computer vision

Speech recognition

Games

... relies on **extensive** and **manual** tuning of algorithms and their hyperparameters
What is Hyper-parameter Optimization (HPO)?
This class!

TRAINING DATA

AutoML black box

Trained model

Query x

Answer y
The HPO Problem

Given

- Training / validation set: \( \mathbf{D}_{\text{train}}, \mathbf{D}_{\text{valid}} \sim \mathcal{D} \),
- Scoring functions \( J_1, J_2 \),
- Hyper-parameters \( \theta \in \Theta \)
- Trainable parameters \( \alpha \in \mathcal{A} \)

\( f_\theta \) is a predictive model such that:

\[
\hat{y}_{\text{valid}} = f_\theta(\mathbf{x}_{\text{valid}} | \arg\min_{\alpha \in \mathcal{A}} J_1(\mathbf{D}_{\text{train}}, \alpha))
\]

we want to

\[
\max_{\theta \in \Theta} J_2(f_\theta)
\]

Ex: \( J_1 = \text{MSE} \), \( J_2 = \text{AUL} / k\text{-fold CV estimator of } J_1 \)
The HPO S A R I

S: Configuration space $\Theta$

A: Choose a configuration point $\theta_t$

R: $J_2(\theta_t)$

I: J, other meta-information
   (dataset meta-features, comp. time, ...)
Challenges of HPO

- Bi-level, black-box optimization
- How to model the complex search space?
- Which sampling strategy?
- How to quickly evaluate a sampled configuration?
HPO Solution Taxonomy

AI Hyperparameter Selection Methods

Model-Free

- Experiential
  - Grid Search
    - Response Surface Methods
    - Design of Experiments

Model-Based

- Random Search
- Bayesian Optimization
  - Evolutionary Algorithms
  - Stochastic Approximation
  - Other Methods
Why Bayesian Optimization (for HPO)?

- Promising results
- Active research field
- Good basis for understanding other HPO methods

<table>
<thead>
<tr>
<th>Rd</th>
<th>Ended</th>
<th>AutoML Winners</th>
<th>&lt;R&gt;</th>
<th>&lt;S&gt;</th>
<th>Final Winners</th>
<th>&lt;R&gt;</th>
<th>&lt;S&gt;</th>
<th>UP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>02/14/15</td>
<td>1. ideal</td>
<td>1.40</td>
<td>0.8159</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2. abhi</td>
<td>3.60</td>
<td>0.7764</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3. aad</td>
<td>4.00</td>
<td>0.7714</td>
</tr>
<tr>
<td>1</td>
<td>02/15/15</td>
<td>1. aad</td>
<td>2.80</td>
<td>0.6401</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. jrl44</td>
<td>3.80</td>
<td>0.6226</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. tadej</td>
<td>4.20</td>
<td>0.6456</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>06/14/15</td>
<td>1. aad</td>
<td>2.20</td>
<td>0.7479</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. ideal</td>
<td>3.20</td>
<td>0.7324</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. amsl</td>
<td>4.60</td>
<td>0.7158</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>06/15/15</td>
<td>1. jrl44</td>
<td>1.80</td>
<td>0.4320</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. aad</td>
<td>3.40</td>
<td>0.3529</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. mat</td>
<td>4.40</td>
<td>0.3449</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>11/14/15</td>
<td>1. ideal</td>
<td>2.00</td>
<td>0.5180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. djaj</td>
<td>2.20</td>
<td>0.5142</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. aad</td>
<td>3.20</td>
<td>0.4977</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11/15/15</td>
<td>1. djaj</td>
<td>2.40</td>
<td>0.0901</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>02/19/16</td>
<td>1. aad</td>
<td>1.80</td>
<td>0.8071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. djaj</td>
<td>2.00</td>
<td>0.7912</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. ideal</td>
<td>3.80</td>
<td>0.7547</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>02/20/16</td>
<td>1. aad</td>
<td>2.20</td>
<td>0.3881</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. djaj</td>
<td>2.20</td>
<td>0.3841</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. macc</td>
<td>2.60</td>
<td>0.3815</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>05/1/16</td>
<td>1. aad</td>
<td>1.60</td>
<td>0.5238</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. ideal</td>
<td>3.60</td>
<td>0.4988</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. abhi</td>
<td>5.40</td>
<td>0.4911</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PU</td>
<td>05/1/16</td>
<td>1. aad</td>
<td>1.60</td>
<td>0.5282</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. djaj</td>
<td>2.60</td>
<td>0.5379</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. post</td>
<td>4.60</td>
<td>0.4150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bayesian Optimization: What and How
Bayesian Optimization: The intuition

- Goal: find the max as soon as possible
- Iterate:
  1: model the function
  2: try the best point according to my model
  3: update my model with my trials

source: adapted from https://en.wikipedia.org/wiki/Bayesian_optimization
BO: key components

True function: unknown

Model: prior/posterior

Infill sampling criteria: acquisition function

Source: adapted from https://en.wikipedia.org/wiki/Bayesian_optimization
Step 1&3:

The prior / posterior

The prior $p(f)$ captures our belief on $f(x)$, it gets updated with our observations $\{(x_i, f(x_i))\}$ to form the posterior $p(f|\text{obs.})$

Ex: Gaussian process (GP): $f(x) \sim GP(m(x), k(x, x'))$

- Distribution of random functions
- Fully determined by mean and covariance function
GP: the kernels

\[ \kappa = \exp \left( -\frac{||x-x'||^2}{2\sigma^2} \right) \]

\[ \kappa = \min(x,x') \]

\[ \kappa = (x' + c)^2 \]

\[ m(x) = 0 \]
Matern kernels

- Quality of GP depends on the kernel
- Good choice: Matern kernels  [Matern, 1960, Stein, 1999]

\[
k_{\nu=p+1/2}(r) = \exp \left( - \frac{\sqrt{2\nu r}}{\ell} \right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left( \frac{\sqrt{8\nu r}}{\ell} \right)^{p-i}.
\]

\[
\nu = 3/2 \text{ or } 5/2 \quad \rightarrow \quad \text{Matern 3/2 and 5/2 kernel}
\]

\[
k_{\nu=3/2}(r) = \left( 1 + \frac{\sqrt{3r}}{\ell} \right) \exp \left( - \frac{\sqrt{3r}}{\ell} \right),
\]

\[
k_{\nu=5/2}(r) = \left( 1 + \frac{\sqrt{5r}}{\ell} + \frac{5r^2}{3\ell^2} \right) \exp \left( - \frac{\sqrt{5r}}{\ell} \right)
\]
function samples with mean = 0 and matern 3/2 (5/2) kernels
[image source: https://pythonhosted.org/infpy/gps.html]
GP: modeling -> sampling -> predicting

Gaussian Process Prediction
[Image source: https://en.wikipedia.org/wiki/Gaussian_process]
GP prior / posterior in BO

In BO, we want to `fit’ the GP with observations
\[ \mathcal{D}_{1,...,t} = \{ \mathbf{x}_{1:t}, \mathbf{f}_{1:t} \} \]
And use the GP posterior to `predict’ \( f(x_{t+1}) \)
\[ p(f_{t+1}|\mathcal{D}_{1,...,t}, x_{t+1}) = \mathcal{N}(\mu_t(x_{t+1}), \sigma^2_t(x_{t+1})) \]
\[ \mu_t(x_{t+1}) = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{f}_{1:t} \]
\[ \sigma^2_t(x_{t+1}) = k(x_{t+1}, x_{t+1}) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} \]

\( \mathbf{k} \) = covariance between \( x_{t+1} \) and all previous samples \( \mathbf{x}_{1:t} \)
\( \mathbf{K} \) = covariance matrix of all previous samples \( \mathbf{x}_{1:t} \)
1D GP with 3 observations, the surrogate mean prediction of $f(x)$ given the data (black line), the variance (shaded area).

[image source: Brochu et al., 2010, A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning]
Step 2: The acquisition function

Determines the next query point, trading-off exploration - exploitation

Ex: Probability of improvement [Kushner et al. 1964]

$$PI(x) = Pr(f(x) \geq f(x^+))$$

$$= \Phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right)$$

- We want to find the pt. w. max. area above the best obs. f(x+)
- This corresponds to the max of PI

[Image source: Brochu et al., 2010]
Other priors

Problems with GP:

- Scales cubically in data points
- High dim. and categorical HP space: need to adapt the kernel

→ Frequentist solution: Random forest
  - a collection of regression trees
  - input: x;
  - output: ^f(x)
  - mean and variance over trees
Other acquisition functions

ParBayesianOptimization in Action (Round 1)

Expected improvement
[Mockus et al., 1978, Jones et al., 1998]

UCB-GP
[Srinivas et al. 2010]

PI
HPO algorithms using BO
Ex 1: Freeze-Thaw BO

[Swersky et al. 2014]

- Intuition:
  Maintains a set of “frozen” (partially completed but not being actively trained) models and uses an information-theoretic criterion to determine which ones to “thaw” and continue training

- Use BO for:
  - learning curve prediction → offers quick evaluations
  - HP space modeling
• GP for learning curve (b):
  - Exponential decay kernel
  - \( p(f_{t+1}|\mathcal{D}_{1:t}) \)

• GP for HP space (c):
  - Matern 5/2 kernel
  - \( p(f_{new}|\mathcal{D}_{1:t}, x_{new}) \)
Run some models

**Algorithm 1** Entropy Search Freeze-Thaw Bayesian Optimization

1: Given a basket \( \{(x, y)\}_{B_{\text{old}}} \cup \{(x)\}_{B_{\text{new}}} \)
2: \( a = (0, 0, \ldots, 0) \)
3: Compute \( P_{\text{min}} \) over the basket using Monte Carlo simulation and Equation 19.
4: for each point \( x_k \) in the basket do
5:     // \( n_{\text{fant}} \) is some specified number, e.g., 5.
6:     for \( i = 1 \ldots n_{\text{fant}} \) do
7:         if the point is old then
8:             Fantasize an observation \( y_{k+1} \) using Equation 20.
9:         end if
10:     if the point is new then
11:         Fantasize an observation \( y_1 \) using Equation 21.
12:     end if
13:     Conditioned on this observation, compute \( P_{\text{min}}^y \) over the basket using Monte Carlo simulation and Equation 19.
14: \( a(k) \leftarrow a(k) + \frac{H(P_{\text{min}}^y) - H(P_{\text{min}})}{n_{\text{fant}}} \) // information gain.
15: end for
16: end for
17: Select \( x_k \), where \( k = \arg\max_k a(k) \) as the next model to run.

Run next model
Ex 2: Auto-sklearn

- Intuition:
  Warm start the BO with meta-learning techniques, ensemble the top models.

- Use BO for:
  HP space modeling
Figure 1: Our improved approach to AutoML. We add two components to Bayesian hyperparameter optimization of an ML framework: meta-learning for initializing the Bayesian optimizer and automated ensemble construction from configurations evaluated during optimization.

**Meta-learning** [Brazdil et al., 2009]:
- characterize the dataset using meta-features,
- initialize BO with config. that performed well on old similar dataset

**BO subroutine**: SMAC [Hutter et al., 2011]
- Random Forest prior
- Expected improvement acquisition
- 1 fold quick evaluation
Today’s
Take-home messages
Take-home messages (1)

What you have learned:

- HPO: bi-level, black-box optimization problem

- Bayesian Optimization: a powerful solution w. 2 key ingredients:
  - a prior: to model the space
  - an acquisition function: to guide the sampling
Take-home messages (2)

What you can use in your projects:

- Auto-sklearn: open-source, active community
  https://automl.github.io/auto-sklearn/master/

- NNI: more than BO, good for deep learning models
  https://github.com/microsoft/nni

- Hyperopt:
  https://github.com/hyperopt/hyperopt
Take-home messages (3)

Open Question and research directions:

- **Benchmarks and Comparability**
  eg. [Black-box Optimization Benchmarking](#), AutoML and AutoDL challenges

- **Gradient-Based Optimization**
  eg. [Maclaurin et al., 2015](#), [Franceschi et al., 2017](#), [Pedregosa, 2016](#), etc.

- **Scalability and parallelization**
  [Bergstra et al., 2011](#), [Desautels et al., 2014](#), [Falkner et al., 2018](#), etc.

- **Towards meta-learning (coming lecture)**