Sequential hyperparameter optimization

MIPT

2023

Model selection: coherent inference

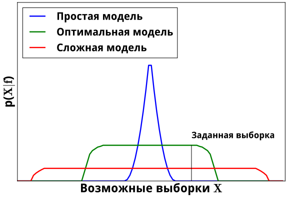
First level: select optimal parameters:

$$\mathbf{w} = \arg\max \frac{p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathfrak{D}|\mathbf{h})},$$

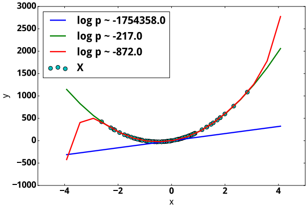
Second level: select optimal model (hyperparameters).

Evidence:

$$p(\mathfrak{D}|\mathbf{h}) = \int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})d\mathbf{w}.$$



Model selection scheme



Example: polynoms

Hyperparameters

Definition

Prior for parameters w and structure $\mathbf{\Gamma}$ of the model f is a distrubution $p(\mathsf{W}, \mathbf{\Gamma}|\mathsf{h}) : \mathbb{W} \times \Gamma \times \mathbb{H} \to \mathbb{R}^+$, where $\mathbb{W}$ is a parameter space, Γ is a structure space.

Definition

Hyperparameters $\mathsf{h} \in \mathbb{H}$ of the models are the parameters of $p(\mathsf{w}, \mathbf{\Gamma}|\mathsf{h})$ (parameters of prior f).

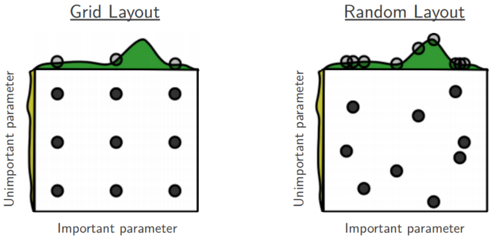
Basic methods of hyperparameter optimization

Variants:

- Grid search;
- random search.

Both methods suffer from curse of dimensionality.

The random search can be more effective if the hyperparameter space is degenerate.



Bergstra et al., 2012

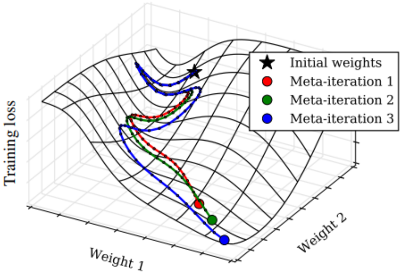
Gradient methods

Idea: Optimize hyperparameters using the full parameter optimization trajectory.

Pros:

- Hyperparameter optimization will consider the features of the parameter optimization.
- The complexity is linear on the number of hyperparameters.

Cons: the computational cost is very expensive.



Maclaurin et. al, 2015. Example.

SMBO

Algorithm Framework 1: Sequential Model-Based Optimization (SMBO)

$\mathbf{R}$ keeps track of all target algorithm runs performed so far and their performances (*i.e.*, SMBO's training data $\{([\boldsymbol{\theta}_1, \boldsymbol{x}_1], o_1), \ldots, ([\boldsymbol{\theta}_n, \boldsymbol{x}_n], o_n)\}$), $\mathcal{M}$ is SMBO's model, $\vec{\boldsymbol{\Theta}}_{new}$ is a list of promising configurations, and t_{fit} and t_{select} are the runtimes required to fit the model and select configurations, respectively.

Input : Target algorithm A with parameter configuration space $\boldsymbol{\Theta}$; instance set Π; cost metric $\hat{c}$

Output : Optimized (incumbent) parameter configuration, $\boldsymbol{\theta}_{inc}$

$[\mathbf{R}, \boldsymbol{\theta}_{inc}] \leftarrow \mathit{Initialize}(\boldsymbol{\Theta}, \Pi)$

repeat

  $[\mathcal{M}, t_{fit}] \leftarrow \mathit{FitModel}(\mathbf{R})$

  $[\vec{\boldsymbol{\Theta}}_{new}, t_{select}] \leftarrow \mathit{SelectConfigurations}(\mathcal{M}, \boldsymbol{\theta}_{inc}, \boldsymbol{\Theta})$

  $[\mathbf{R}, \boldsymbol{\theta}_{inc}] \leftarrow \mathit{Intensify}(\vec{\boldsymbol{\Theta}}_{new}, \boldsymbol{\theta}_{inc}, \mathcal{M}, \mathbf{R}, t_{fit} + t_{select}, \Pi, \hat{c})$

until *total time budget for configuration exhausted*

return $\boldsymbol{\theta}_{inc}$

TPE

Basic idea

- Sample multiple hyperparameter instances h_i
- Fit models f_i using h_i
- Select models from λ-quantile of model results Loss_λ and fit adaptive parzen estimator p_1
- Select reaming models and fit adaptive parzen estimator p_2
- Sample new hyperparameter h that maximizes Expected improvement: $\int_{\infty}^{\mathrm{Loss}_\lambda}(\mathrm{Loss}_\lambda - u)p(u|\mathrm{h})du$.

Gaussian process, definition (wiki)

- A random process f_t with continuous time is gaussian if and only if for each finite set of indices $t_1, \ldots, t_k$: $f_{t_1}, \ldots, f_{t_k}$ is a mutlivariative Gaussian variable.
- Each linear combination $f_{t_1}, \ldots, f_{t_k}$ is a univariative Gaussian.

Definition (simplified)

Define a Gaussian process $\mathfrak{GP}(m(x), k(x, x')$ to be a distribution on the set of functions that for each x, x': $\mathfrak{GP}(m(x), k(x, x')$ is a Gaussian distribution.

Example: regression

$$\mathsf{f} \sim \mathcal{N}(0, \mathsf{K}),$$

where K is a covariance matrix for X.

$$y \sim f + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

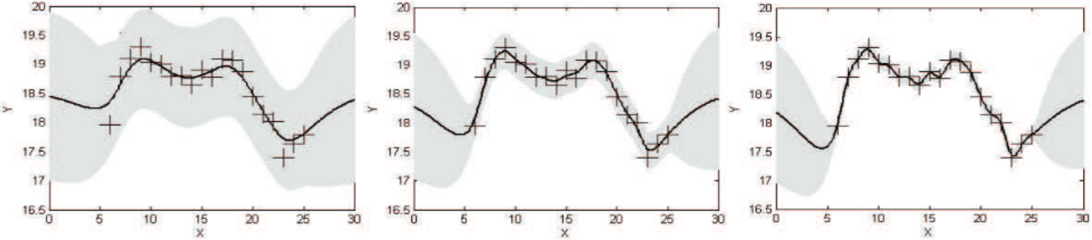
Then $\mathsf{y} \sim \mathcal{N}(0, \mathsf{K} + \sigma^2 \mathsf{I})$ is a prediction for new objects $\hat{\mathsf{y}}$:

$$\hat{\mathsf{y}} \sim \mathcal{N}(\mathsf{K}'^{\mathsf{T}}(\mathsf{K} + \sigma^2 \mathsf{I})^{-1}\mathsf{y}, \mathsf{K}'' - \mathsf{K}'^{\mathsf{T}}(\mathsf{K} + \sigma^2 \mathsf{I})^{-1}\mathsf{K}'.$$

Gaussian process: main features

- Nonparametric
  - defined via covariance function and noise level
  - optimization: via MLE
- Prior is defined for the function, not for the parameters
  - we can set prior on the parameters of Gaussian process, then we get GP with degenerate covariance matrix
- Prediction complexity: $O(N^3)$.

sigma^2 and prediction performance



(McDuff, 2010)

Covariance functions

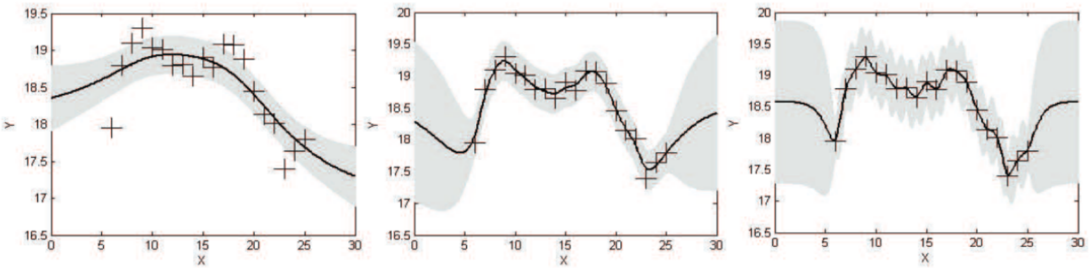
Requirements and properties:

- Symmetry: $K(x, x') = K(x, x')$;
- Positive semidefiniteness: $v^T K v \geq 0$;
- Stationarity: $K(x, x') = K(x + a, x' + a)$;
- Isotropy: dependence only on $||x - x'||$.

Covariance functions: examples

- Exponential: $K = \sigma_0^2 \exp\left(\frac{-(x-x')^2}{2\lambda}\right)$
- Linear: $\sigma_0 + xx'$
- Brownian: $\min(x, x')$
- Periodic: $\exp\left(\frac{-2\sin^2\left(\frac{x-x'}{2}\right)}{\lambda^2}\right)$
- Defined using NN

λ and prediction performance



(McDuff, 2010)

Matérn covariance

$$\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|x-x'|}{\lambda}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}|x-x'|}{\lambda}\right)$$

- K_ν is a modified Bessel function;
- Stationary and isotropic;
- With $\nu \to \infty$: exponential
- With finite ν: less smooth

Hyperparameter optimization

(Snoek et al., 2012)

$$\mathsf{f}(\mathsf{x},\mathsf{w},\mathsf{h}) = \mathsf{f}(\mathsf{w}(\mathsf{h}),\mathsf{h}|\mathsf{x})$$

The model f is a function from hyperparameters:

$$\mathsf{f} \sim \mathcal{GP}, \mathsf{y} \sim \mathsf{f} + \varepsilon$$

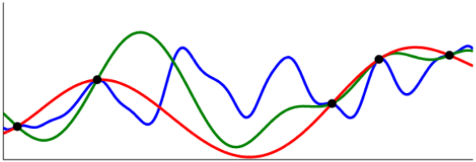
Using Matérn covariance with $\nu = 2.5$.

Next point to estimate

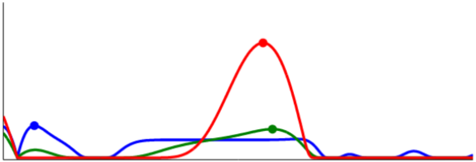
Next point selection is done using *Acquisition function*:

- Upper confidence level
- Probability of Improvement: $P(I(\mathrm{h} > 0), I(\mathrm{h}) = \max(L(\mathrm{h}) - L(\mathrm{h}^*))$
- Expected improvement $\int_{\infty}^{\mathrm{L}^*}(\mathrm{Loss}_{\lambda} - u)p(u|\mathrm{h})du$.

λ and hyperparameter optimization

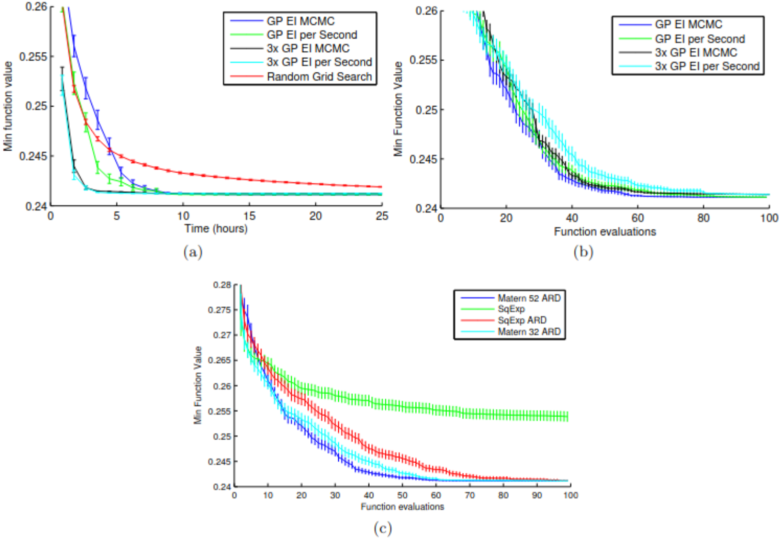


(a) Posterior samples under varying hyperparameters

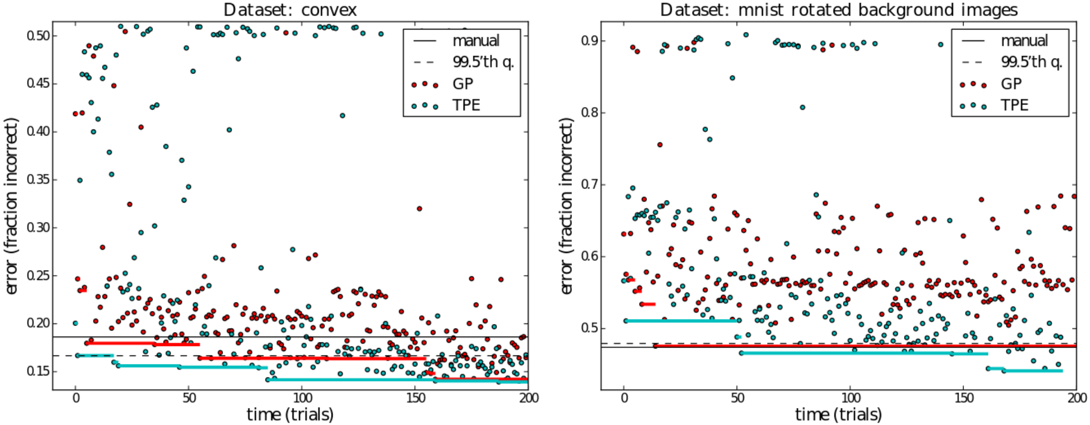


(b) Expected improvement under varying hyperparameters

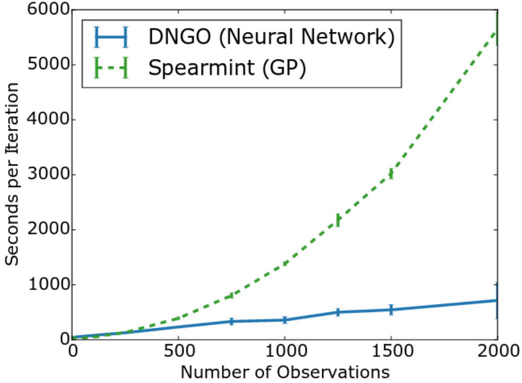
Hyperparameter optimization



TPE vs GP


Dataset: convex
error (fraction incorrect)
time (trials)
manual
99.5'th q.
GP
TPE
Dataset: mnist rotated background images
error (fraction incorrect)
time (trials)
manual
99.5'th q.
GP
TPE

GP: complexity challenge



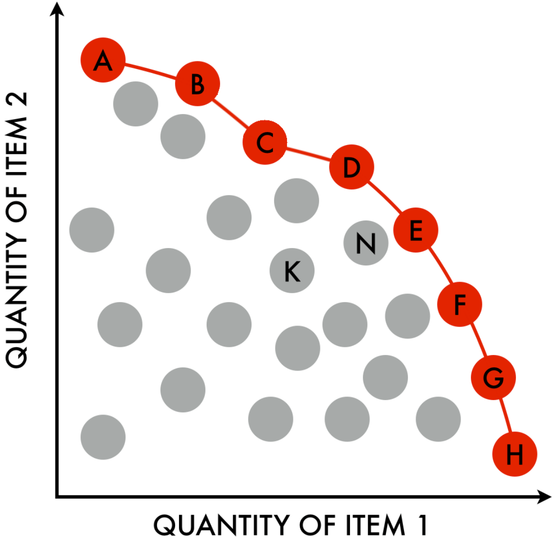
(Snoek, 2015)

Muilti-objective optimization

Can we use multiple criteria for optimization task?

Muilti-objective optimization

Can we use multiple criteria for optimization task?



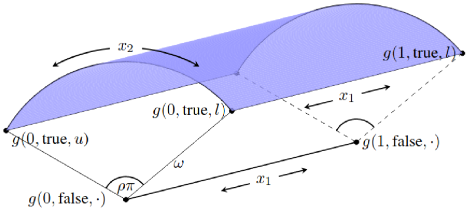
Covariance function for conditional parameters

NN selection problem:

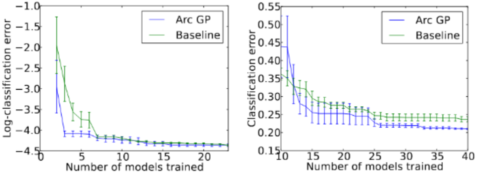
- Hidden layer neuron numbers: ok, we can set as a real-valued hyperparameter
- Layer number: ok, we can set as a real-valued hyperparameter
- **How to compare two architectures with different layer number?**
- **How to compare models with [100, 100] and [100] neurons?**

(Swersky et al., 2014): use a special covariance function!

Covariance function for conditional parameters: idea

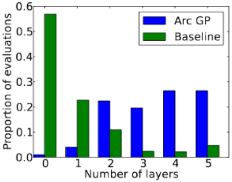


Covariance function for conditional parameters: results



(a) MNIST (b) CIFAR-10

Figure 2: Bayesian optimization results using the arc kernel.



Covariance function for conditional parameters: scheme

- If we are comparing two points for which the same parameters are relevant, the value of any unused parameters shouldn't matter,

$$k\big((x_1, \text{false}, x_2), (x_1', \text{false}, x_2')\big) = k\big((x_1, \text{false}, x_2''), (x_1', \text{false}, x_2''')\big),\ \forall x_2, x_2', x_2'', x_2''';\quad (1)$$

- The covariance between a point using both parameters and a point using only one should again only depend on their shared parameters,

$$k\big((x_1, \text{false}, x_2), (x_1', \text{true}, x_2')\big) = k\big((x_1, \text{false}, x_2''), (x_1', \text{true}, x_2''')\big),\ \forall x_2, x_2', x_2'', x_2'''.\quad (2)$$

Reference

- Bishop C. M. Pattern recognition //Machine learning. – 2006. – T. 128. – No. 9.
- Daniel McDuff, Gaussian Processes, https://courses.media.mit.edu/2010fall/mas622j/ProblemSets/slidesGP.pdf
- Chris Williams, Gaussian Processes for Machine Learning, https://www.newton.ac.uk/files/seminar/20070809140015001-150844.pdf
- Ed Snelson, utorial: Gaussian process models for machine learning, https://mlg.eng.cam.ac.uk/tutorials/06/es.pdf
- Snoek J., Larochelle H., Adams R. P. Practical bayesian optimization of machine learning algorithms //Advances in neural information processing systems. – 2012. – T. 25.
- Swersky K. et al. Raiders of the lost architecture: Kernels for Bayesian optimization in conditional parameter spaces //arXiv preprint arXiv:1409.4011. – 2014.
- Snoek J. et al. Scalable bayesian optimization using deep neural networks //International conference on machine learning. – PMLR, 2015. – C. 2171-2180.