# Bayesian Optimization for Hyperparameter Optimization

## The Alternative Optimization Approach of the Tree-Parzen Estimator (TPE)

Bernd Bischl  Frank Hutter  Lars Kotthoff
Marius Lindauer  Joaquin Vanschoren

# Overview of TPE [Bergstra et al. 2011]

- Standard Bayesian optimization models the probability $p(y \mid \boldsymbol{\lambda})$ of observations $y$ given configurations $\boldsymbol{\lambda}$
- Instead, TPE fits kernel density estimators (KDEs) $l(\boldsymbol{\lambda} \mid y \leq \gamma)$ and $g(\boldsymbol{\lambda} \mid y \leq \gamma)$
  - These KDEs are for "good configurations" (leading to objective function values below a threshold $\gamma$) and "bad configurations"
  - By default, $\gamma$ is set to the 15% quantile of the observations

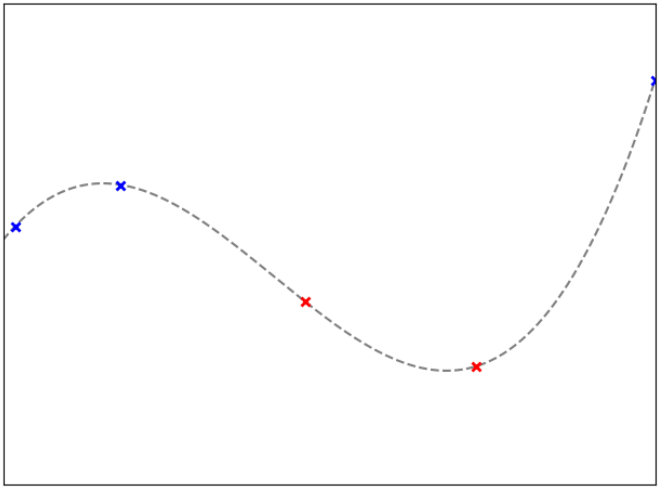
# Overview of TPE [Bergstra et al. 2011]

- Standard Bayesian optimization models the probability $p(y \mid \boldsymbol{\lambda})$ of observations $y$ given configurations $\boldsymbol{\lambda}$
- Instead, TPE fits kernel density estimators (KDEs) $l(\boldsymbol{\lambda} \mid y \leq \gamma)$ and $g(\boldsymbol{\lambda} \mid y \leq \gamma)$
  - These KDEs are for "good configurations" (leading to objective function values below a threshold $\gamma$) and "bad configurations"
  - By default, $\gamma$ is set to the 15% quantile of the observations
- Optimizing $l(\boldsymbol{\lambda})/g(\boldsymbol{\lambda})$ is equivalent to optimizing standard expected improvement in Bayesian optimization [Bergstra et al. 2011]

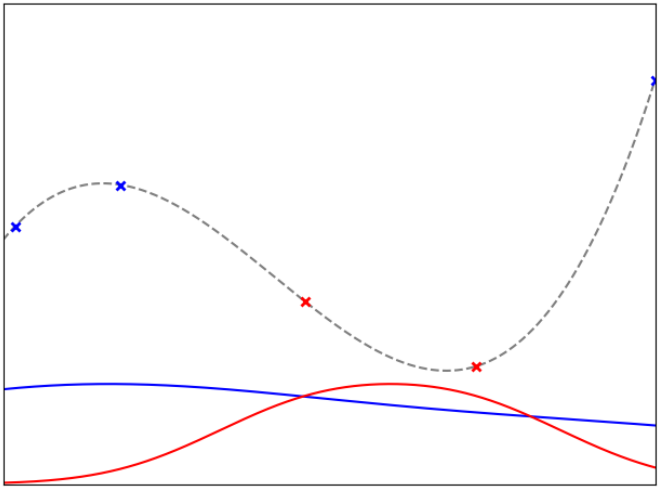
# Overview of TPE [Bergstra et al. 2011]

- Standard Bayesian optimization models the probability $p(y \mid \boldsymbol{\lambda})$ of observations $y$ given configurations $\boldsymbol{\lambda}$
- Instead, TPE fits kernel density estimators (KDEs) $l(\boldsymbol{\lambda} \mid y \leq \gamma)$ and $g(\boldsymbol{\lambda} \mid y \leq \gamma)$
  - These KDEs are for "good configurations" (leading to objective function values below a threshold $\gamma$) and "bad configurations"
  - By default, $\gamma$ is set to the 15% quantile of the observations
- Optimizing $l(\boldsymbol{\lambda})/g(\boldsymbol{\lambda})$ is equivalent to optimizing standard expected improvement in Bayesian optimization [Bergstra et al. 2011]
- Why is the technique called TPE?
  - The used KDEs are Parzen estimators
  - TPE can handle tree-structured search spaces

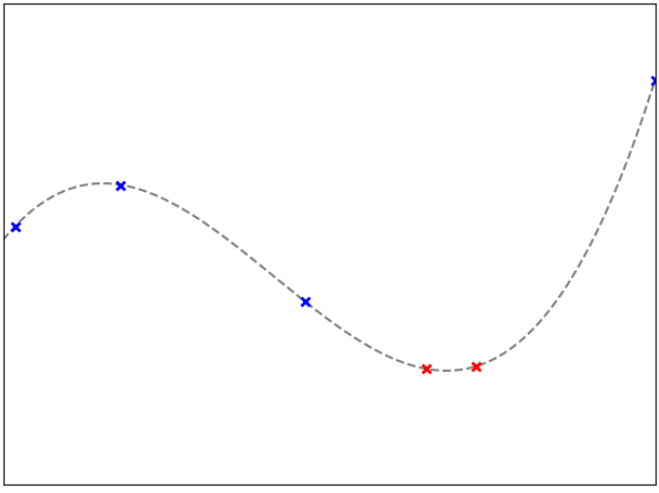
# TPE Example



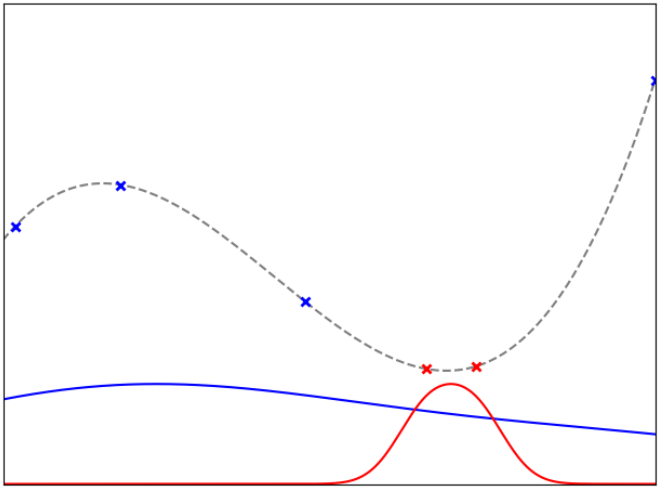
# TPE Example



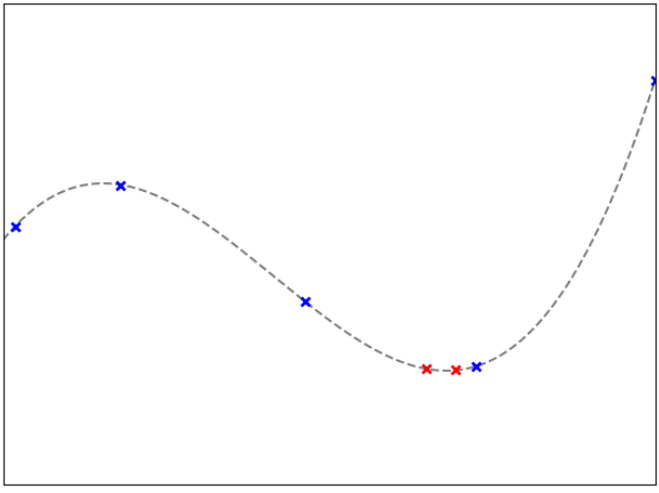
# TPE Example



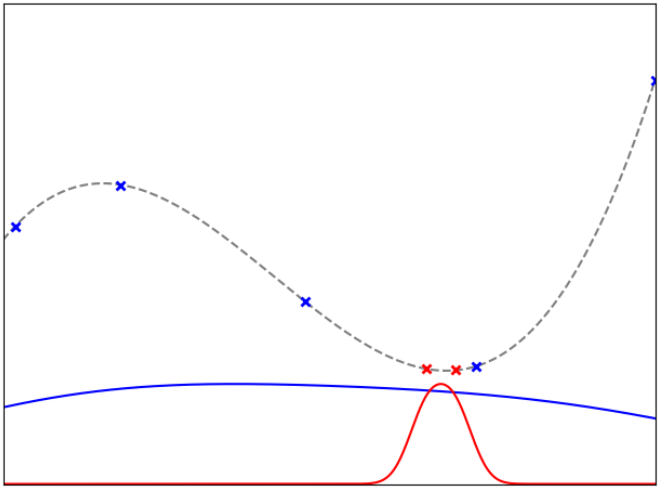
# TPE Example



# TPE Example



# TPE Example



# TPE Pseudocode

**TPE loop**

**Require:** Search space $\mathbf{\Lambda}$, cost function $c$, percentile $\gamma$, maximal number of function evaluations $T$

**Result :** Best observed configuration $\boldsymbol{\lambda}$ according to $\mathcal{D}^{(T)}$

$\mathcal{D}^{(0)} \leftarrow \varnothing$

**for** $t = 1$ **to** $T$ **do**

  $\mathcal{D}_{\text{good}}, \mathcal{D}_{\text{bad}} \leftarrow$ split $\mathcal{D}^{(t-1)}$ according to quantile $\gamma$

  $l(\boldsymbol{\lambda})$, $g(\boldsymbol{\lambda}) \leftarrow$ fit KDE on $\mathcal{D}_{\text{good}}$, $\mathcal{D}_{\text{bad}}$ respectively

  $\mathbf{\Lambda}_{\text{cand}} \leftarrow$ draw samples from $l$;

  Select next query point: $\boldsymbol{\lambda}^{(t)} \in \arg\max_{\boldsymbol{\lambda} \in \mathbf{\Lambda}_{\text{cand}}} l(\boldsymbol{\lambda}) / g(\boldsymbol{\lambda})$

  Query $c(\boldsymbol{\lambda}^{(t)})$

  $\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t-1)} \cup \{\langle \boldsymbol{\lambda}^{(t)}, c(\boldsymbol{\lambda}^{(t)}) \rangle\}$

**end**

# Further Details

Remarks:

- TPE models $p(\boldsymbol{\lambda}|c(\boldsymbol{\lambda}))$
  - we can multiply it with a prior to add expert knowledge

# Further Details

Remarks:

- TPE models $p(\boldsymbol{\lambda}|c(\boldsymbol{\lambda}))$
  - we can multiply it with a prior to add expert knowledge
- Performance of TPE depends on:
  - setting of $\gamma$ to trade-off exploration and exploitation
  - bandwidth of the KDEs

# Further Details

Remarks:

- TPE models $p(\boldsymbol{\lambda}|c(\boldsymbol{\lambda}))$
  - we can multiply it with a prior to add expert knowledge
- Performance of TPE depends on:
  - setting of $\gamma$ to trade-off exploration and exploitation
  - bandwidth of the KDEs
- A successful tool implementing TPE is Hyperopt [Bergstra et al.]

# Summary

## Advantages

- Computationally efficient: $O(Nd)$
- Parallelizable
- Robust
- Can handle complex search spaces with priors

# Summary

## Advantages

- Computationally efficient: $O(Nd)$
- Parallelizable
- Robust
- Can handle complex search spaces with priors

## Disadvantages

- Less sample-efficient than GPs

# Questions to Answer for Yourself / Discuss with Friends

- Disussion. Is TPE really Bayesian optimization?
- Disussion. How does $\gamma$ impact the optimization procedure?
- Disussion. Derive that optimizing $l(\boldsymbol{\lambda})/g(\boldsymbol{\lambda})$ is equivalent to optimizing expected improvement.