BORE: Bayesian Optimization by Density-Ratio Estimation

- 作者：Louis C. Tiao, Aaron Klein, Matthias Seeger, et al
- 机构：University of Sydney, Amazon, NVIDIA

- 会议：ICML2021
- 地址：https://arxiv.org/abs/2102.09009
- 代码：https://github.com/ltiao/bore

论文主要内容

摘要

Bayesian optimization是广泛使用的BBO方法，BO依据acquisition function的explore-exploit trade-off 准则，其中acquisition function主要来自一个概率surrogate model。但EI（流行的acquisition function）的analytical tractability阻碍了bayes optimization的效率和适用性。文章将EI的计算问题转化成二分类问题，构建了类概率估计与密度比估计的联系、EI与密度比的联系。通过规避tractability限制，能够带来model表达力、多用性、可扩展性等优势。

贡献

1. 列出TPE在（Density-Ratio Estimation）DRE上的缺陷
2. 将计算EI的问题转化成**概率分类问题**（simple but powerful）
3. 将**二分类概率**估计和**密度比**估计建立联系

研究内容

Motivation

- Expected improvement (EI)

$$\mathbf{x}_{N+1}=\arg\max_{\mathbf{x}\in\mathcal{X}}\alpha\left(\mathbf{x};\mathcal{D}_N\right)$$

$$\alpha\left(\mathbf{x};\mathcal{D}_N,\tau\right):=\mathbb{E}_{p(y|\mathbf{x},\mathcal{D}_N)}[\max(\tau-y,0)]$$

  - 主流、简单有效
  - Compute and optimize

- 当model posterior predict是Gaussian的时候 => closed-form expression （限制了model族）
- 一般情况，只能在model族表达力和求解能力之间折中

- surrogate model for **acquisition function**：构建代理模型最终还是为了acquisition function的采样
- 传统EI formulation:

  设

$$p\left(y \mid \mathbf{x}, \mathcal{D}_N\right)=\mathcal{N}\left(y \mid \mu(\mathbf{x}), \sigma^2(\mathbf{x})\right)$$

  有

$$\alpha\left(\mathbf{x} ; \mathcal{D}_N, \tau\right)=\sigma(\mathbf{x}) \cdot\left[\frac{\tau-\mu(\mathbf{x})}{\sigma(\mathbf{x})} \cdot \Psi\left(\frac{\tau-\mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)\right)+\psi\left(\frac{\tau-\mu(\mathbf{x})}{\sigma(\mathbf{x})}\right))]$$

Formulation

γ-relativedensity-ratio:

$$r_\gamma(\mathbf{x}):=\frac{\ell(\mathbf{x})}{\gamma \ell(\mathbf{x})+(1-\gamma) g(\mathbf{x})}$$

EI的计算和求解问题转化为：

$$\alpha\left(\mathbf{x} ; \mathcal{D}_N, \Phi^{-1}(\gamma)\right) \propto r_\gamma(\mathbf{x})$$

$$\mathbf{x}_{\star}=\underset{\mathbf{x} \in \mathcal{X}}{\arg \max } r_0(\mathbf{x})$$

陷阱

1. Singularities： $\gamma=0$ 导致 $\ell(x)$ 没有mass => **DRE**
2. Vapnik’s principle：避免在求解的**中间步骤**中引入一个更genral problem(density estimation)=>**DRE**
3. Kernel bandwidth：KDE难以使用一个**fixed bandwith**适应高-低密度区域
4. Error sensitivity：估计 $\ell(x)$ 和 $g(x)$ 再求比例 （=> 直接估计density-ratio **DRE**）
5. Curse of dimensionality：KDE（=> **DRE**）
6. Optimization：除了estimation，还需要方便关于inputs x 优化

方法

定义

$$\pi(\mathbf{x})=p(z=1 \mid \mathbf{x})$$

则

$$r_\gamma(\mathbf{x}) = \gamma^{-1}\pi(\mathbf{x})$$

现在ploblem变成了找一个classify $\pi(x)$.

$$\mathcal{L}(\boldsymbol{\theta}) := -\frac{1}{N}\left(\sum_{n=1}^{N} z_n \log \pi_{\boldsymbol{\theta}}(\mathbf{x}_n) + (1-z_n)\log(1-\pi_{\boldsymbol{\theta}}(\mathbf{x}_n))\right)$$

*总结：优化**EI** <= 密度比估计 <= 找二分类器* $\pi_{\theta^*}(x)$

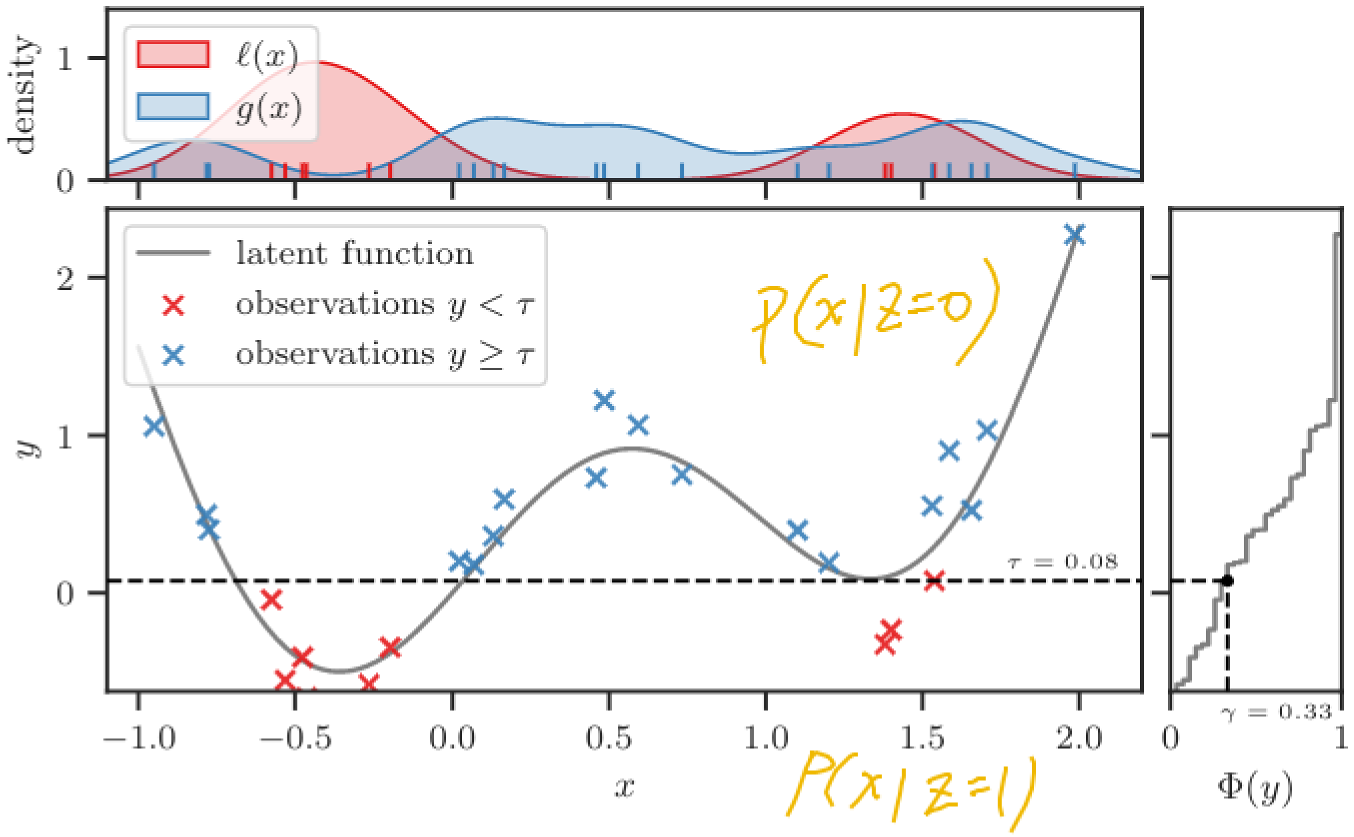
Algorithm 1: Bayesian optimization by density-ratio estimation (BORE).

Input: blackbox $f : \mathcal{X} \to \mathbb{R}$, proportion $\gamma \in (0, 1)$, probabilistic classifier $\pi_{\boldsymbol{\theta}} : \mathcal{X} \to [0, 1]$.

while *under budget* **do**

- $\tau \leftarrow \Phi^{-1}(\gamma)$ // compute γ-th quantile of $\{y_n\}_{n=1}^N$
- $z_n \leftarrow \mathbb{I}[y_n \le \tau]$ for $n = 1, \dots, N$ // assign labels
- $\tilde{\mathcal{D}}_N \leftarrow \{(\mathbf{x}_n, z_n)\}_{n=1}^N$ // construct auxiliary dataset
- /* update classifier by optimizing parameters $\boldsymbol{\theta}$ wrt log loss */
- $\boldsymbol{\theta}_\star \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ // depends on $\tilde{\mathcal{D}}_N$, see eq. 9
- /* suggest candidate by optimizing input $\mathbf{x}$ wrt classifier */
- $\mathbf{x}_N \leftarrow \arg\max_{\mathbf{x}\in\mathcal{X}} \pi_{\boldsymbol{\theta}_\star}(\mathbf{x})$ // see eq. 10
- $y_N \leftarrow f(\mathbf{x}_N)$ // evaluate blackbox function
- $\mathcal{D}_N \leftarrow \mathcal{D}_{N-1} \cup \{(\mathbf{x}_N, y_N)\}$ // update dataset
- $N \leftarrow N + 1$

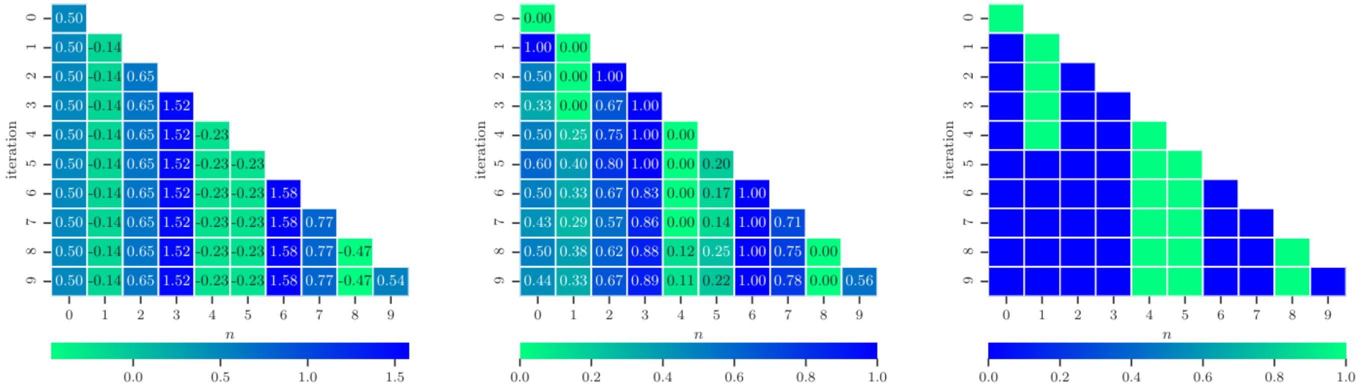
end



优势：

- 可以使用几乎任何SOTA的 classification method
- 强大的model family可以处理non-linear, non-stationary, and heteroscedastic phenomena **frequently encountered in practice**

实验结果



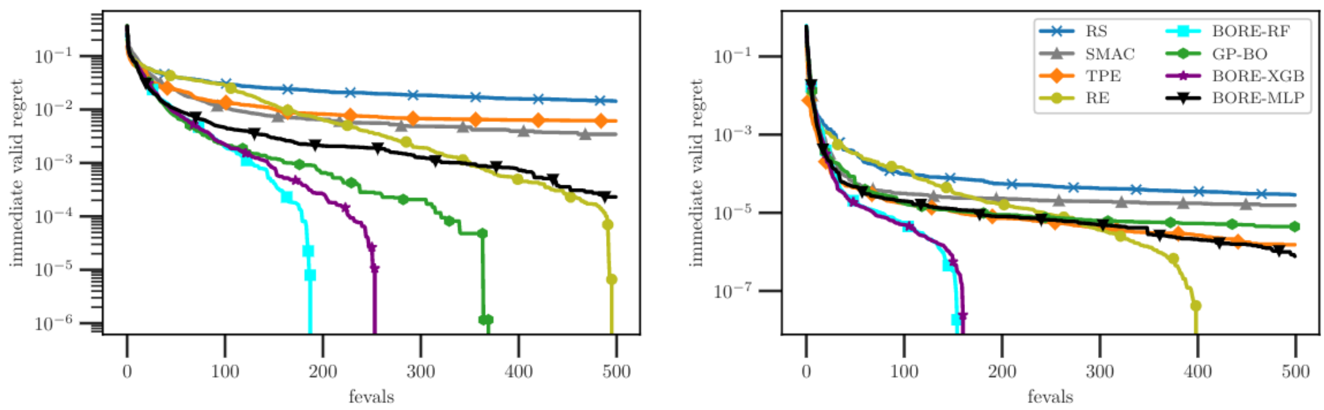
(a) Continuous targets y_n (b) Empirical distribution $\Phi(y_n)$ (c) Binary labels z_n

- Class imbalance： γ
- Label changes across iterations.（exploration）

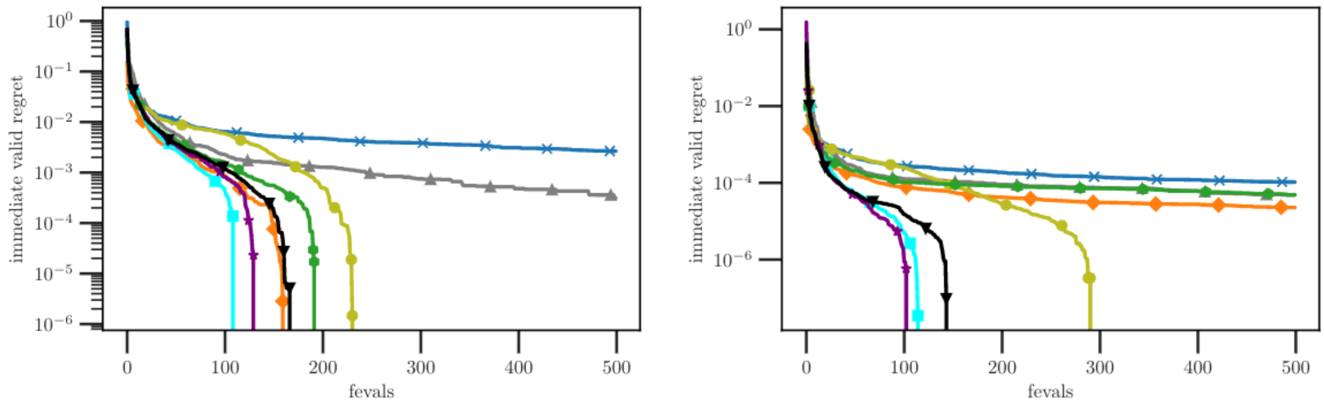
Classify 可以parameter reuse（online learning)

TASK

HBOBench (categorical and ordinal)



(a) PROTEIN (b) NAVAL

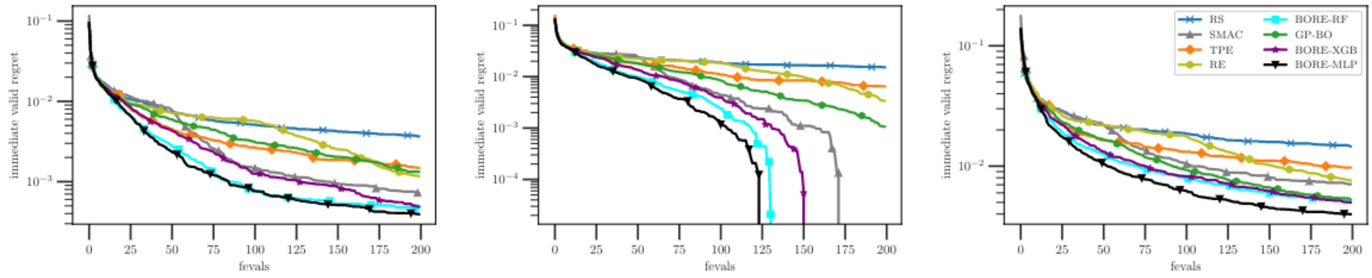


(c) PARKINSONS (d) SLICE

Figure 3. Immediate regret over function evaluations on the HPOBench neural network tuning problems ($D = 9$).

收敛速度领先1-2百个evaluation

NASBench201（pure categorical input）



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet-16

Figure 4. Immediate regret over function evaluations on the NASBench201 neural architecture search problems ($D = 6$).

Robot arm pushing (require large number of function evaluation)

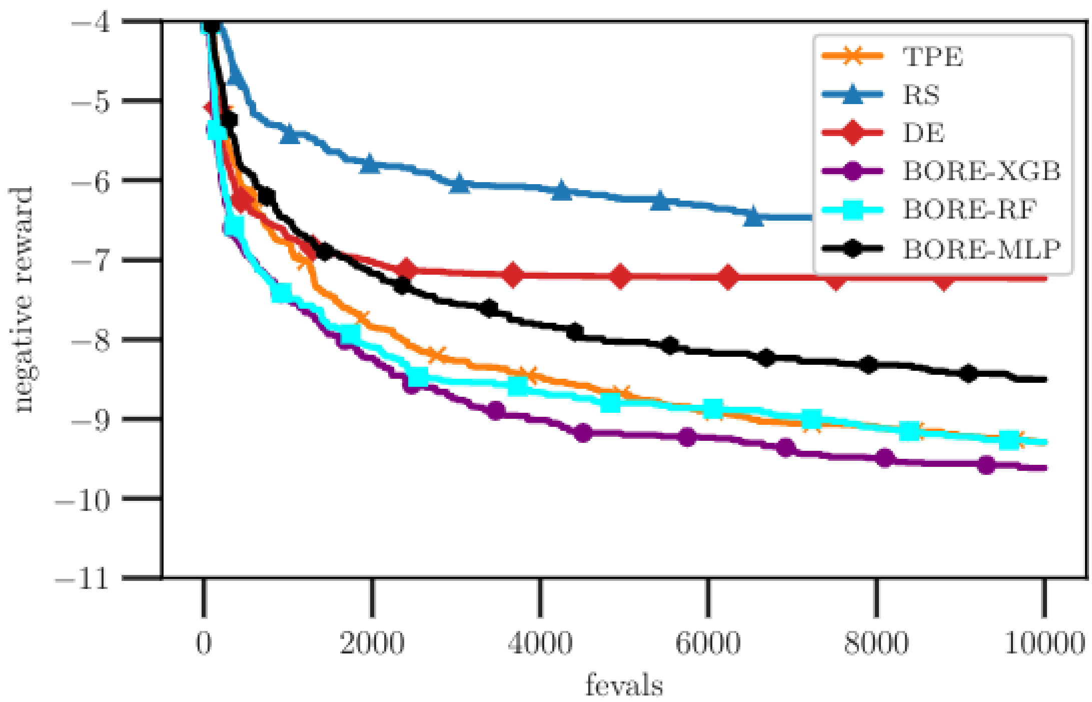
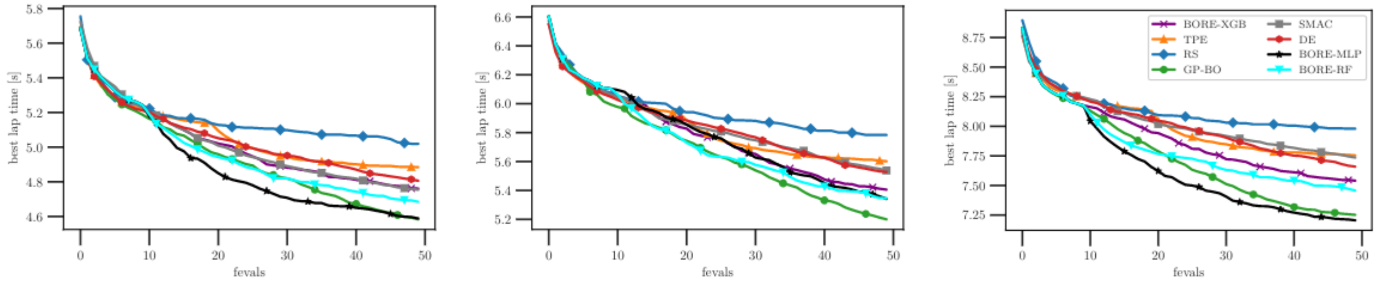


Figure 5. Negative reward over function evaluations on the Robot Pushing task ($D = 14$).

Racing line optimization (function 连续、光滑、⩽ 20 dim)



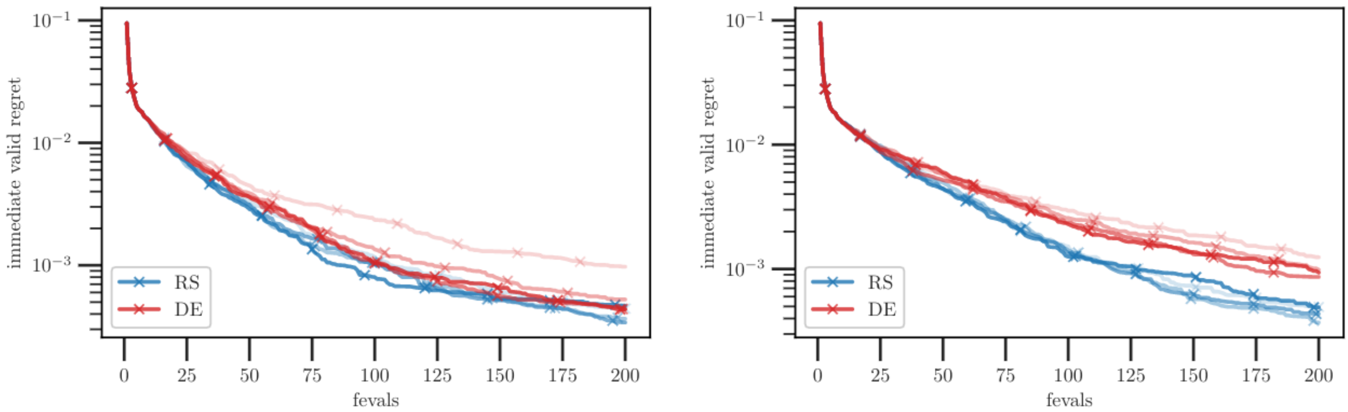
(a) UC BERKELEY ($D = 12$) (b) ETH ZÜRICH A ($D = 20$) (c) ETH ZÜRICH B ($D = 21$)

Figure 6. Best lap times (in seconds) over function evaluations in the racing line optimization problem on various racetracks.

BORE 一致表现更好，除了GP-BO

消融实验

Maximizing the acquisition function



(a) BORE-RF on CIFAR-10 (b) BORE-XGB on CIFAR-10

Figure 10. A comparison of various acquisition optimization strategies on the NASBench201 problem.

- Random search (RS)比Differential evolution (DE)表现好一点
- DE的evaluation budgets高一点效果好，但RS则不是

Calibration

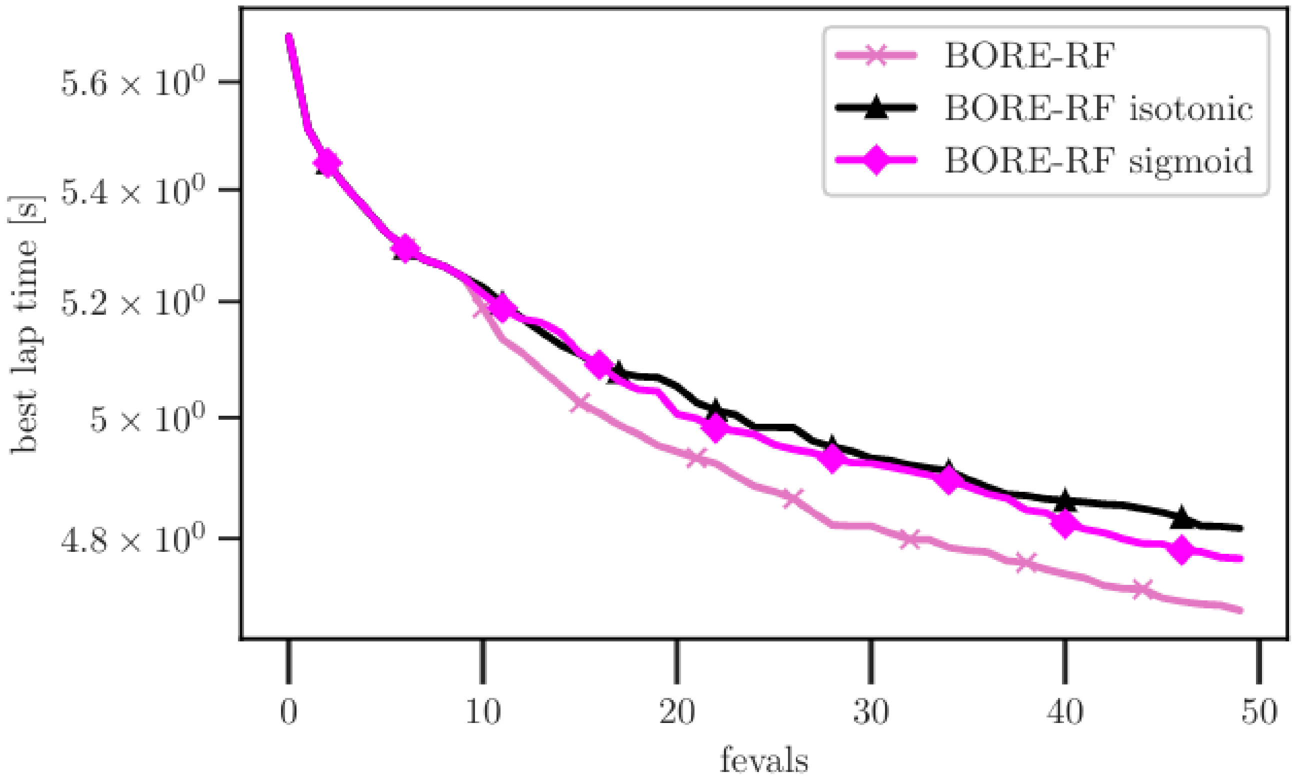
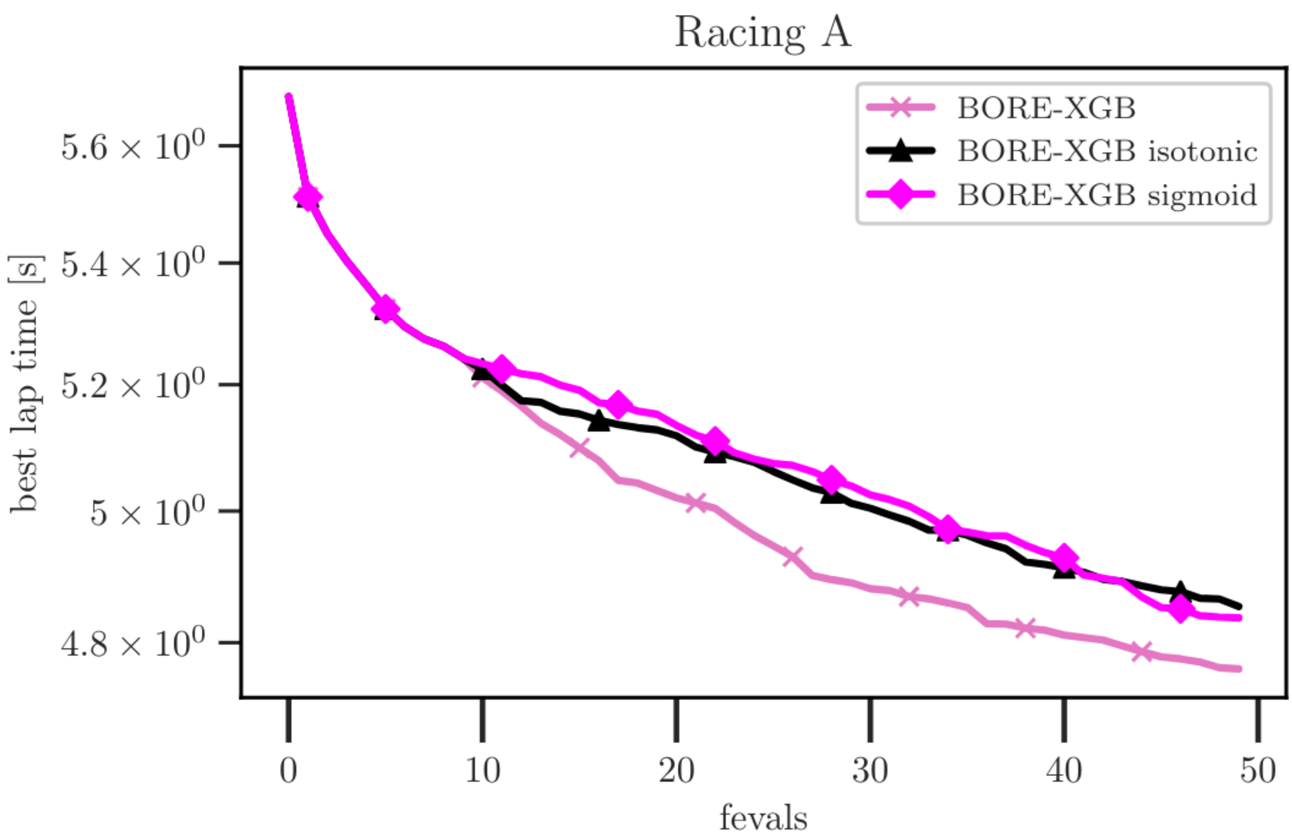
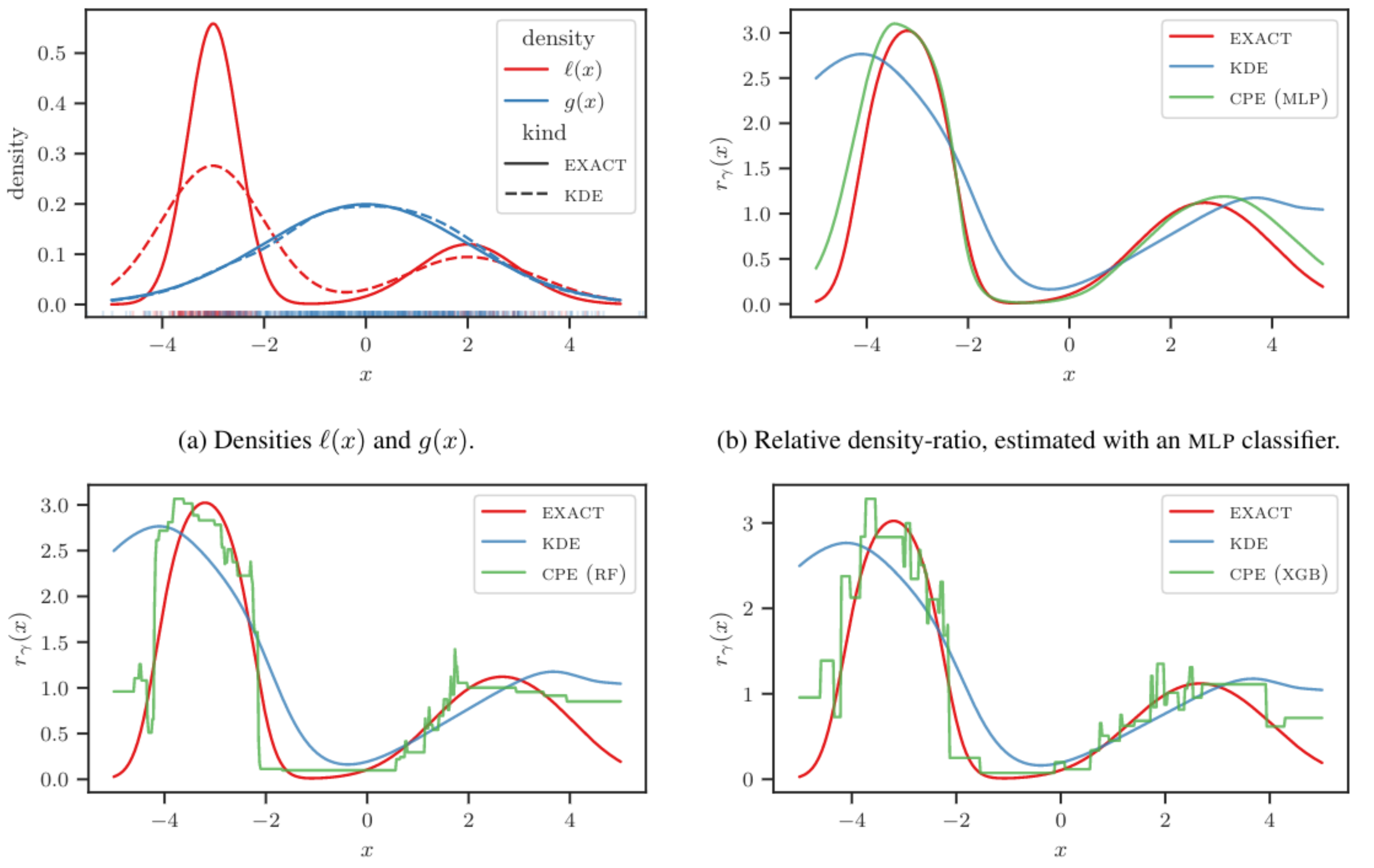


Figure 7. Effects of calibrating RFs in the BORE-RF variant. Results of racing line optimization on the UC BERKELEY track.



calibration方法容易导致过拟合现象（BO考虑怎么在少的evaluation下到达全局最优）。本质原因是classify的训练集是evaluation function的结果，就导致训练classify的**数据集小**

KDE vs DRE



(a) Densities $\ell(x)$ and $g(x)$.

(b) Relative density-ratio, estimated with an MLP classifier.

(c) Relative density-ratio, estimated with a RF classifier.

(d) Relative density-ratio, estimated with an XGBOOST classifier.

Figure 12. Synthetic toy example with (mixtures of) Gaussians.

KDE专注于估计概率密度的具体值，而DER直接估计密度比。在KDE中轻微的error就会导致比值巨大变化