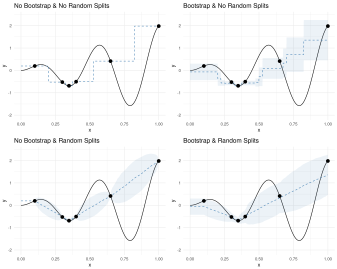
Optimization in Machine Learning

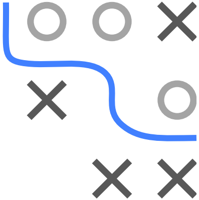
Bayesian Optimization
Important Surrogate Models



Learning goals

- Search space / input data peculiarities in black box problems
- Gaussian process
- Random forest

SURROGATE MODELS



Desiderata:

- Regression model (there are also classification approaches)
- Non-linear local model
- Accurate predictions (especially for small sample sizes)
- Often: uncertainty estimates
- Robust, works often well without human modeler intervention

Depending on the application:

- Can handle different types of inputs (numerical and categorical)
- Can handle dependencies (i.e., hierarchical input)

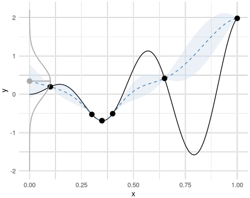
GAUSSIAN PROCESS

Posterior predictive distribution for test point $\mathbf{x} \in \mathcal{S}$:

$$Y(\mathbf{x}) \mid \mathbf{x}, \mathcal{D}^{[t]} \sim \mathcal{N}\left(\hat{f}(\mathbf{x}), \hat{s}^2(\mathbf{x})\right)$$

with

$$\begin{aligned} \hat{f}(\mathbf{x}) &= k(\mathbf{x})^\top \boldsymbol{K}^{-1} \boldsymbol{y} \\ \hat{s}^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x})^\top \boldsymbol{K}^{-1} k(\mathbf{x}) \end{aligned}$$



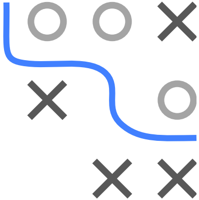
Kernel method, based on kernel / Gram matrix $\boldsymbol{K} := \left(k(\mathbf{x}^{[i]}, \mathbf{x}^{[j]})\right)_{i,j}$

GAUSSIAN PROCESS / 2

Example kernel functions:

- Radial basis function kernel (also known as Gauss kernel):
  $$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d(\mathbf{x},\mathbf{x}')^2}{2l^2}\right)$$
  - l length scale; $d(\cdot, \cdot)$ Euclidean distance
  - infinitely differentiable - very “smooth”
- Matérn kernels:
  $$k(\mathbf{x}, \mathbf{x}') = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\mathbf{x}, \mathbf{x}')\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(\mathbf{x}, \mathbf{x}')\right)$$
  - l length scale; $d(\cdot, \cdot)$ Euclidean distance; $K_{\nu}(\cdot)$ modified Bessel function; $\Gamma(\cdot)$ Gamma function
  - for $\nu = 3/2$ once differentiable, for $\nu = 5/2$ twice differentiable
  - Popular choice as a kernel function when using a GP as SM

GAUSSIAN PROCESS / 3



Pros:

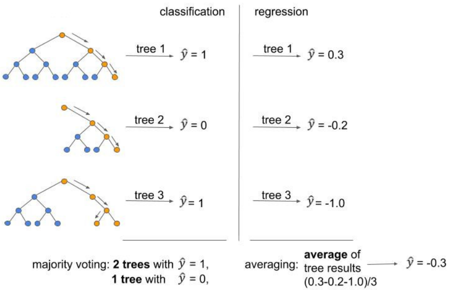
- Smooth, local, powerful estimator, also for small data
- GPs yield well-calibrated uncertainty estimates
- The posterior predictive distribution under a GP is normal

Cons:

- Vanilla GPs scale cubic in the number of data points
- Can natively only handle numeric features
  Mixed inputs / dependencies require special kernels
- GPs aren’t that robust; numerical problems can occur
- Can be sensitive to the choice of kernel and hyperparameters

RANDOM FOREST

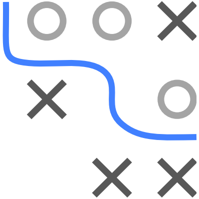
- Bagging ensemble
- Fit *B* decision trees on bootstrap samples
- Feature subsampling



“extratrees” / random splits:

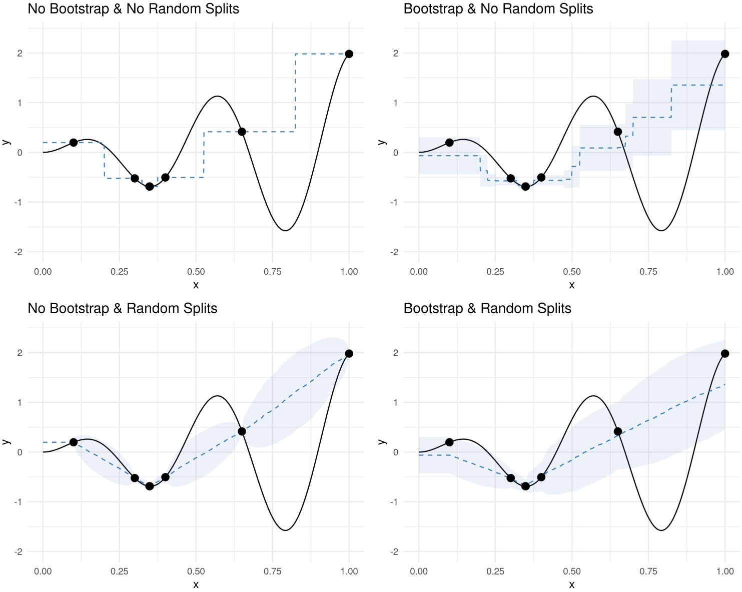
- Choose split location uniformly at random
- Results in a “smoother” mean prediction

RANDOM FOREST - MEAN AND VARIANCE



- Let $\hat{f}_b : \mathcal{S} \to \mathbb{R}$ be the mean prediction of a decision tree b (mean of all data points in the same node as observation $\mathbf{x} \in \mathcal{S}$)
- Let $\hat{s}_b^2 : \mathcal{S} \to \mathbb{R}$ be the variance prediction (variance of all data points in the same node as observation $\mathbf{x} \in \mathcal{S}$)
- Mean prediction of forest: $\hat{f} : \mathcal{S} \to \mathbb{R}$, $\mathbf{x} \mapsto \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\mathbf{x})$
- Variance prediction of forest: $\hat{s}^2 : \mathcal{S} \to \mathbb{R}$,
  $\mathbf{x} \mapsto \left(\frac{1}{B} \sum_{b=1}^{B} \hat{s}_b^2(\mathbf{x}) + \hat{f}_b(\mathbf{x})^2 \right) - \hat{f}(\mathbf{x})^2$
  (law of total variance assuming a mixture of B models)
- Alternative variance estimator:
  - (infinitesimal) Jackknife
- Variance prediction derived from randomness of individual trees
  - Bagging / boostrap samples
  - Features sampled at random
  - (randomized split locations in the case of “extratrees”)

RANDOM FOREST - DIFFERENT CHOICES



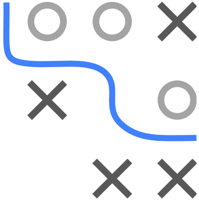
No Bootstrap & No Random Splits

Bootstrap & No Random Splits

No Bootstrap & Random Splits

Bootstrap & Random Splits

RANDOM FOREST



Pros:

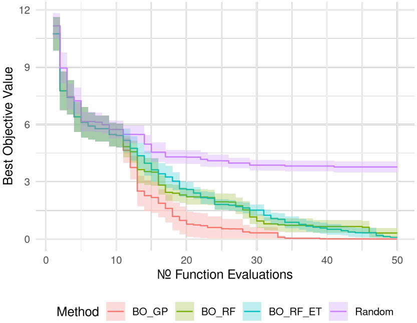
- Cheap(er) to train
- Scales well with the number of data points
- Scales well with the number of dimensions
- Can easily handle hierarchical mixed spaces. Either via imputation or directly respecting dependencies in the tree structure
- Robust

Cons:

- Suboptimal uncertainty estimates
- Not really Bayesian (no real posterior predictive distribution)
- Poor extrapolation

EXAMPLE

Minimize the 2D Ackley Function using BO_GP (GP with Matérn 3/2, EI), BO_RF (standard Random Forest, EI), BO_RF_ET (Random Forest with extratrees, EI) or a random search:



Strong BO_GP performance. BO_RF and BO_RF_ET not too bad either. BO_RF_ET maybe slightly better final performance than BO_RF.