Robust Surrogate Models in Bayesian Optimization

Julian Rodemann

Summer Retreat Department of Statistics

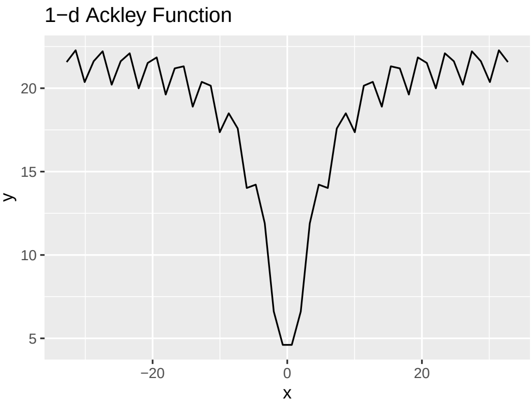
23.10.2021

Image credits: Sigillum: Wikimedia (cc). Logo: Wikimedia (cc). Both last accessed 21.10.2021.

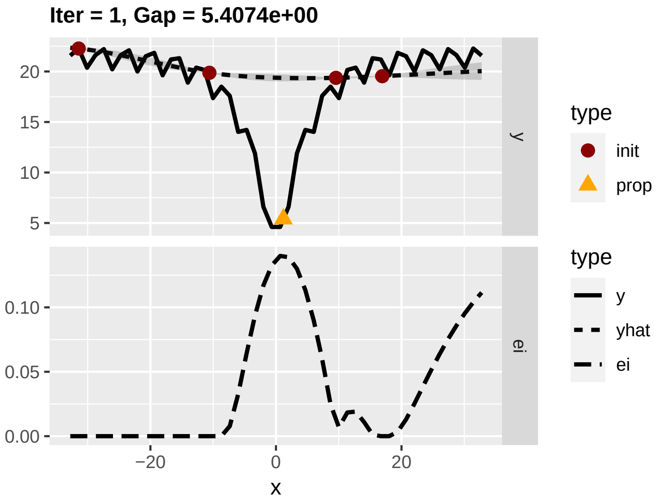
Agenda

Agenda

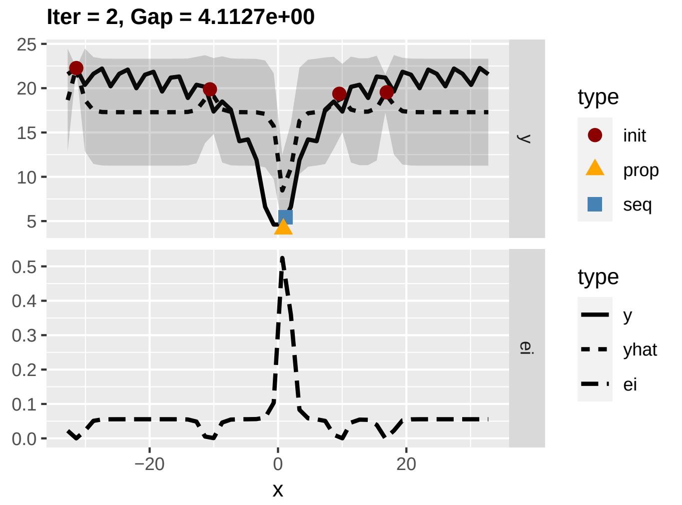
Everyone heard of Bayesian/model-based optimization before? skip intro



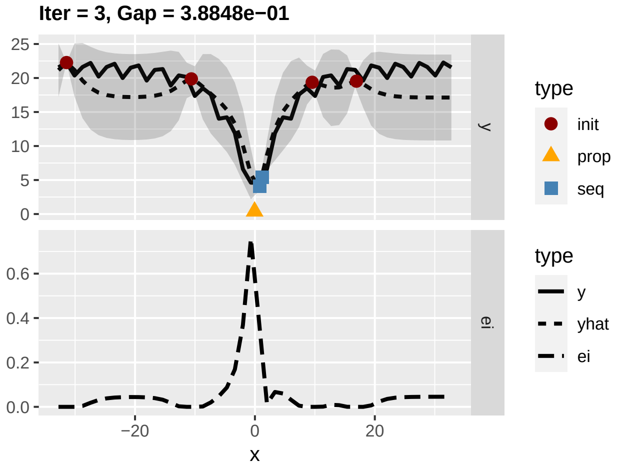
Note: If not otherwise stated, all figures are based on own computations using ggplot2 [Wickham, 2016], smoof [Bossek, 2017] and mlr(3)MBO [Bischl et al., 2017]



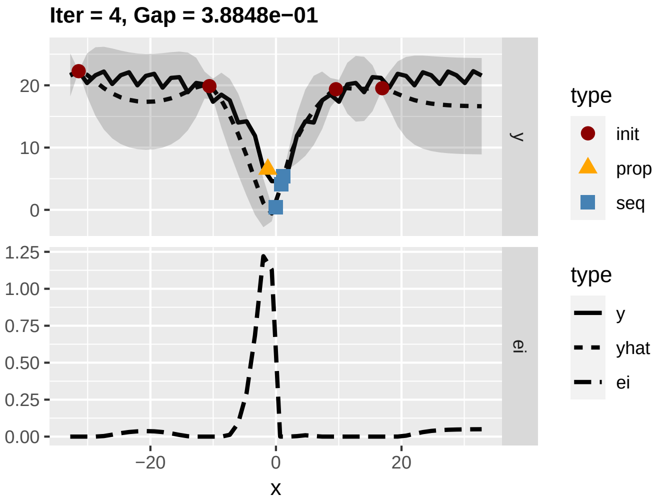
Iteration 1: Surrogate Model (top) and Acquisition Function (bottom)


Iter = 2, Gap = 4.1127e+00
25
20
15
10
5
0.5
0.4
0.3
0.2
0.1
0.0
−20
0
20
x
y
ei
type
init
prop
seq
type
y
yhat
ei

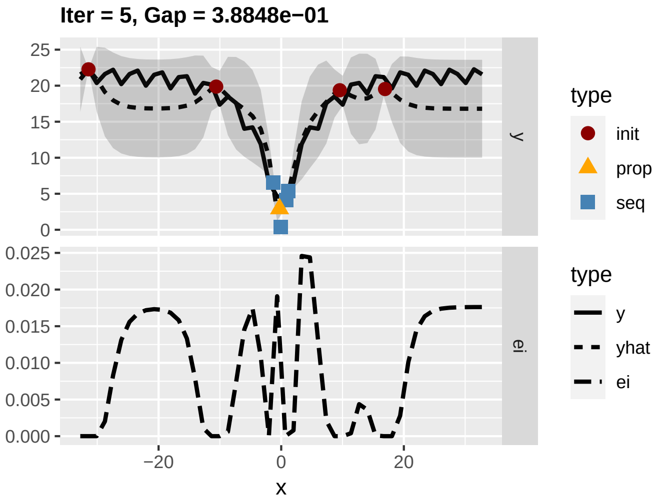
Iteration 2


Iter = 3, Gap = 3.8848e−01
25
20
15
10
5
0
y
0.6
0.4
0.2
0.0
ei
−20
0
20
x
type
init
prop
seq
type
y
yhat
ei

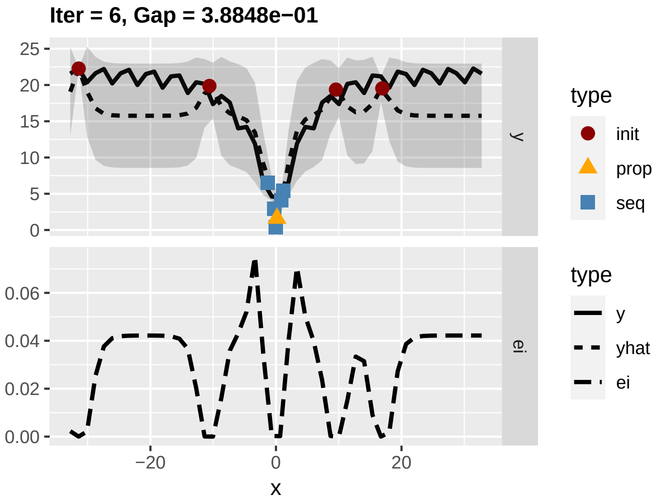
Iteration 3


Iter = 4, Gap = 3.8848e−01
20
10
0
y
1.25
1.00
0.75
0.50
0.25
0.00
ei
−20
0
20
x
type
init
prop
seq
type
y
yhat
ei

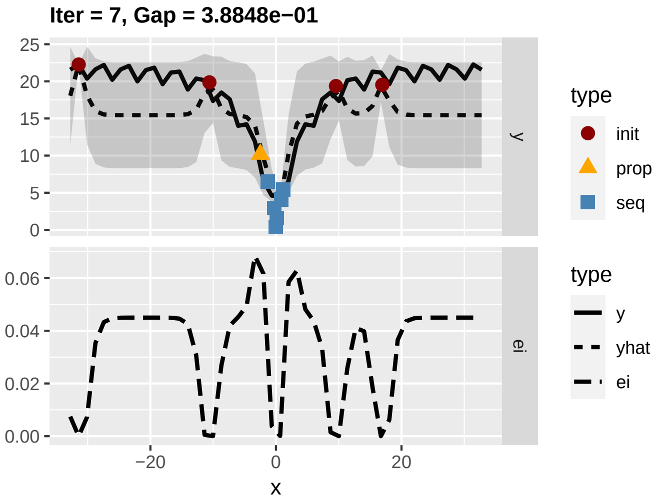
Iteration 4


Iter = 5, Gap = 3.8848e−01
25
20
15
10
5
0
y
0.025
0.020
0.015
0.010
0.005
0.000
ei
−20
0
20
x
type
init
prop
seq
type
y
yhat
ei

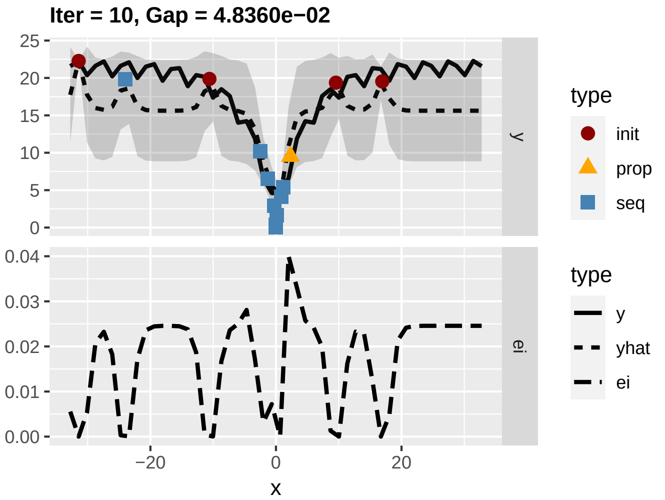
Iteration 5


Iter = 6, Gap = 3.8848e−01
25
20
15
10
5
0
y
0.06
0.04
0.02
0.00
ei
−20
0
20
x
type
init
prop
seq
type
y
yhat
ei

Iteration 6

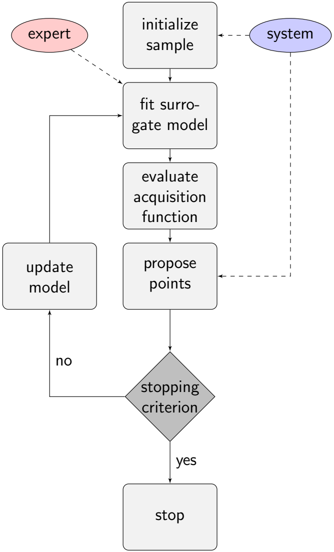

Iter = 7, Gap = 3.8848e−01
25
20
15
10
5
0
0.06
0.04
0.02
0.00
−20
0
20
x
y
ei
type
init
prop
seq
type
y
yhat
ei

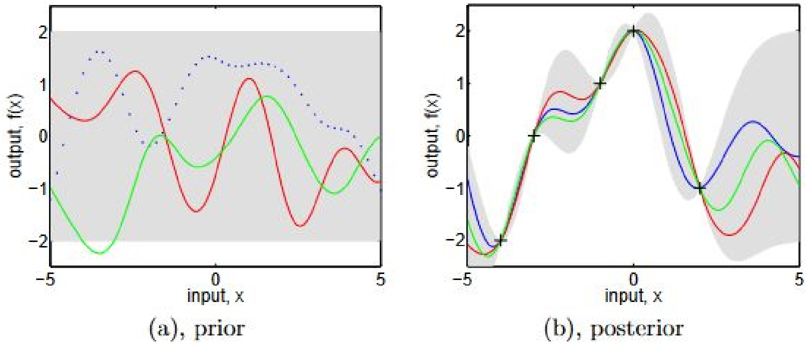
Iteration 7



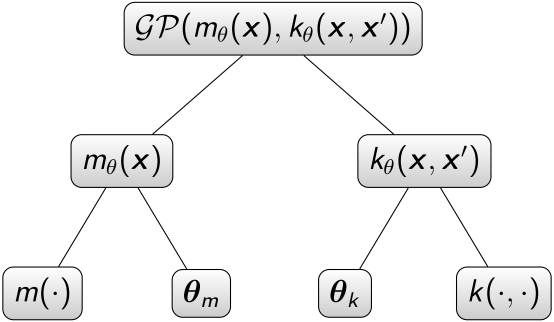
Iteration 10

Bayesian Optimization





Functional GP regression: Three functions drawn from prior (a) and posterior (b) GP. Image credits: [Rasmussen, 2003].



How to specify $m(\cdot)$, $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_k$ and $k(\cdot, \cdot)$ in absence of prior knowledge?

Prior near-ignorance models

- Idea: Use set of $\boldsymbol{\theta}_m$ instead of precise $\boldsymbol{\theta}_m$. Fully specify the other components.
- [Mangili, 2015] proposes imprecise Gaussian processes $\left\{\mathcal{GP}\left(Mh, k_{\boldsymbol{\theta}}(x,x') + \frac{1+M}{c}\right) : h = \pm 1, M \geq 0\right\}$, given a base kernel $k_{\boldsymbol{\theta}}(x,x')$ and a degree of imprecision $c > 0$.
  - $\rightarrow$ set of posteriors with upper and lower mean estimates $\underline{\mu}(x)_c$, $\overline{\mu}(x)_c$

Note: See [Benavoli and Zaffalon, 2015] for background of imprecise GPs and prior near-ignorance.

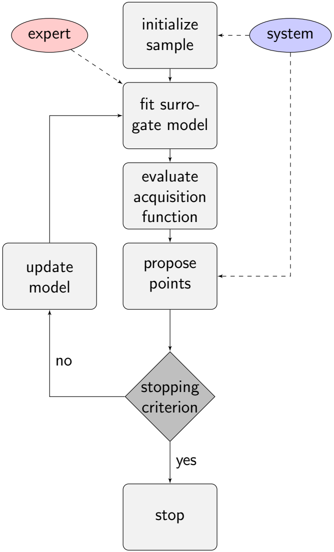
Agenda

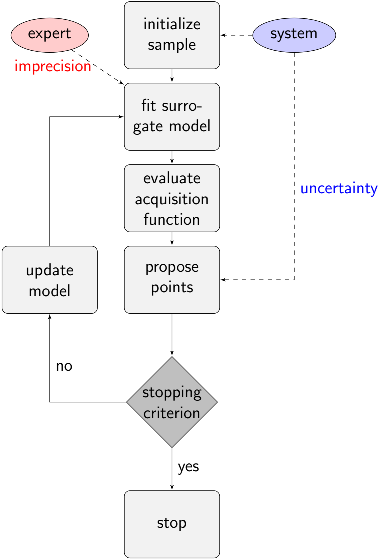
1. Bayesian Optimization
2. Gaussian Processes
3. Prior-Mean-Robust BO
   - Prior near-ignorance models
   - GLCB
4. Weighted ML Estimation of θ_m
5. Questions?
6. Questions!
7. References

GLCB (Generalized Lower Confidence Bound)

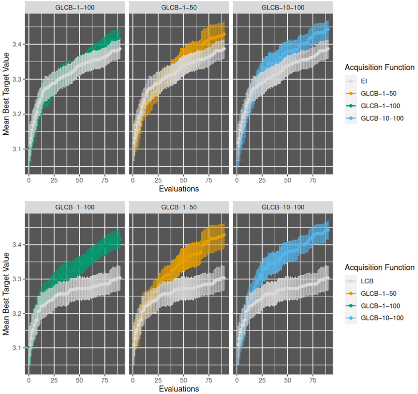
- $LCB(x) = -\widehat{\mu}(x) + \tau \cdot \sqrt{\widehat{\text{Var}}(\mu(x))}$
- $GLCB(x) = -\widehat{\mu}(x) + \tau \cdot \underbrace{\sqrt{\widehat{\text{Var}}(\mu(x))}}_{\text{“classical” uncertainty}} + \rho \cdot \underbrace{(\overline{\mu}(x)_c - \underline{\mu}(x)_c)}_{\text{prior-induced imprecision}}$
  - τ is the degree of risk-aversion
  - ρ is the degree of ambiguity aversion

Bayesian Optimization




expert
initialize sample
system
imprecision
fit surro- gate model
evaluate acquisition function
uncertainty
update model
propose points
no
stopping criterion
yes
stop

GLCB – Results



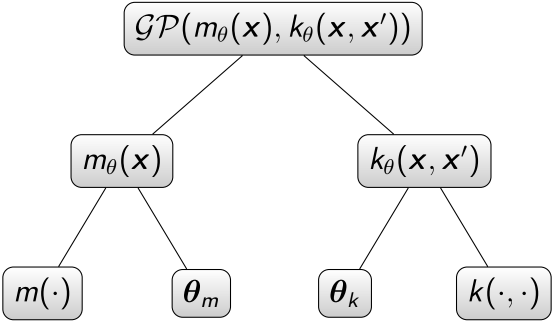
BO with GLCB on Graphene function. GLCB-1-50 means GLCB with $\rho = 1$, $c = 50$. Data source: [Wahab et al., 2020].

Note: If not otherwise stated, all figures are based on own computations using `ggplot2` [Wickham, 2016], `smoof` [Bossek, 2017] and `mlr(3)MBO` [Bischl et al., 2017]

Agenda

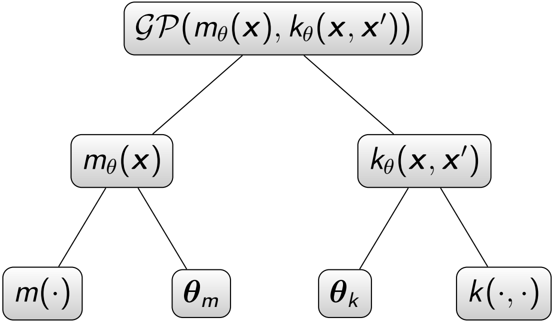
Agenda

Recall: Gaussian Processes – Prior Components



How to specify $m(\cdot)$, $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_k$ and $k(\cdot, \cdot)$
in absence of prior knowledge?

Recall: Gaussian Processes – Prior Components



How to specify $m(\cdot)$, $\boldsymbol{\theta}_m$, $\boldsymbol{\theta}_k$ and $k(\cdot, \cdot)$
in absence of prior knowledge?

Answer[1]**:** "Empirical Bayes", i.e. $\hat{\boldsymbol{\theta}}_m = \arg\max_{\boldsymbol{\theta}_m} \mathcal{L}(\boldsymbol{\theta}_m | \boldsymbol{X}_t)$, where $\boldsymbol{X}_t$ is the incumbent design of iteration t.

[1] given by most software libraries

Weighted ML Estimation of $\boldsymbol{\theta_m}$

- $\hat{\boldsymbol{\theta}}_m = \arg\max_{\boldsymbol{\theta}_m} \mathcal{L}(\boldsymbol{\theta}_m | \boldsymbol{X}_t)$ requires an iid. sample

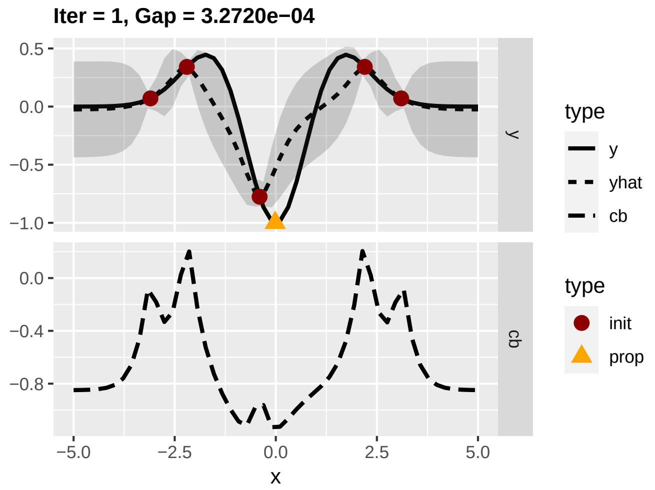
$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{X}_t) \overset{ind.}{=} \prod_{i=1}^{n} \ell(\boldsymbol{\theta}_m|\boldsymbol{x}_i) \overset{iden.}{=} \prod_{i=1}^{n} f(\boldsymbol{\theta}_m|\boldsymbol{x}_i),$$

- iid. assumption is fulfilled for initial sample
- ↯ in the course of the optimization, however, $\boldsymbol{X}_t$ becomes biased
- ↯ can slow down BO performance

Example: Mexican Hat Function

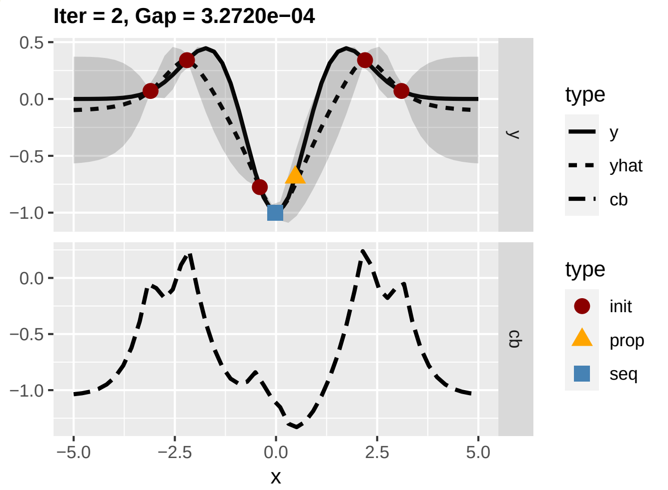
Image credits: Pixabay (cc license)

Example: Mexican Hat Function



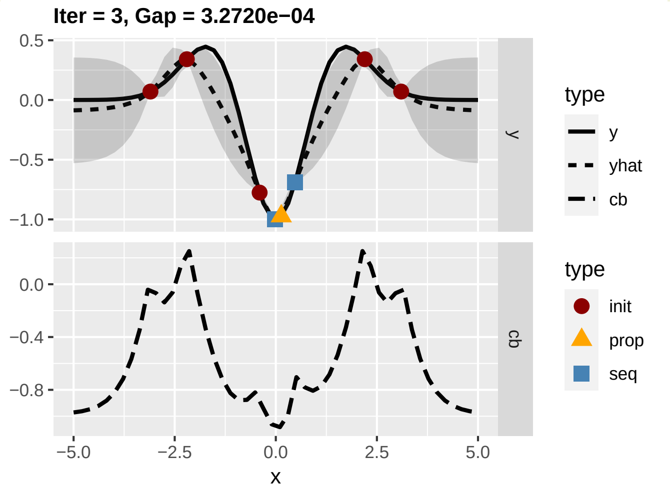
Iteration 1 of BO on Mexican Hat Function

Example: Mexican Hat Function



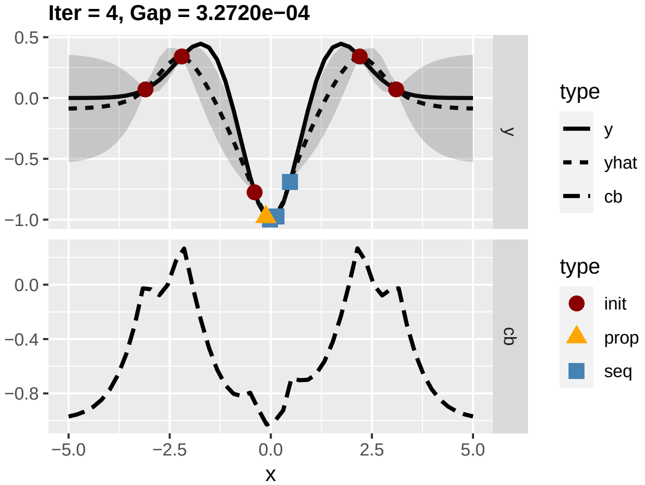
Iteration 2

Example: Mexican Hat Function

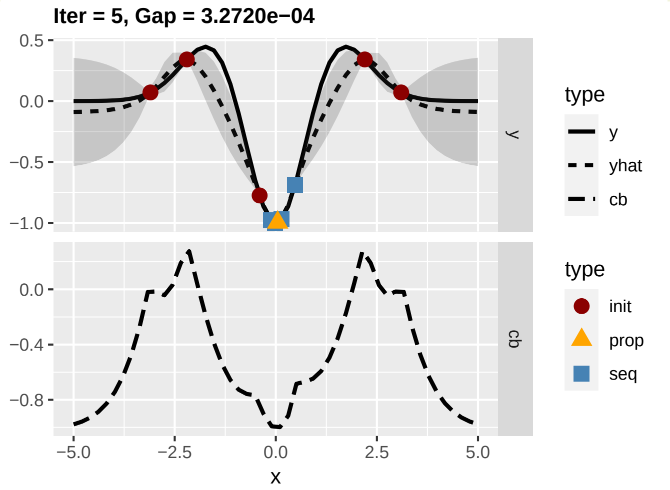


Iteration 3

Example: Mexican Hat Function



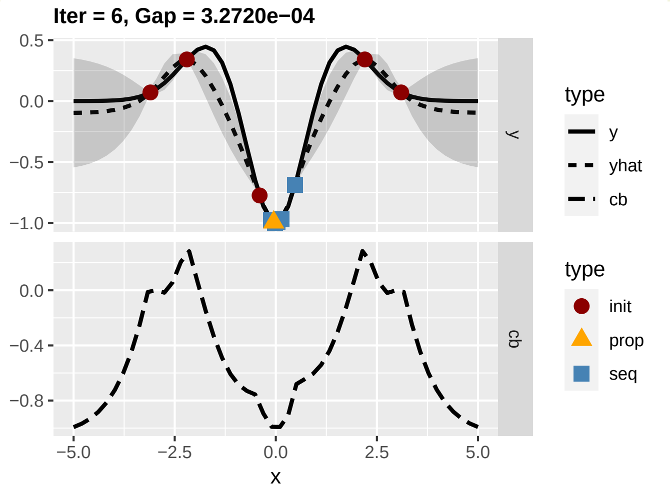
Iteration 4



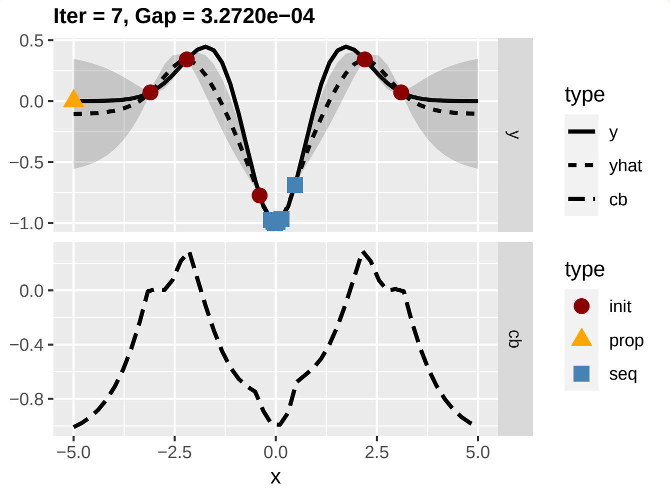
Iter = 5, Gap = 3.2720e−04
0.5
0.0
−0.5
−1.0
y
0.0
−0.4
−0.8
cb
−5.0
−2.5
0.0
2.5
5.0
x
type
y
yhat
cb
type
init
prop
seq

Iteration 5

Example: Mexican Hat Function



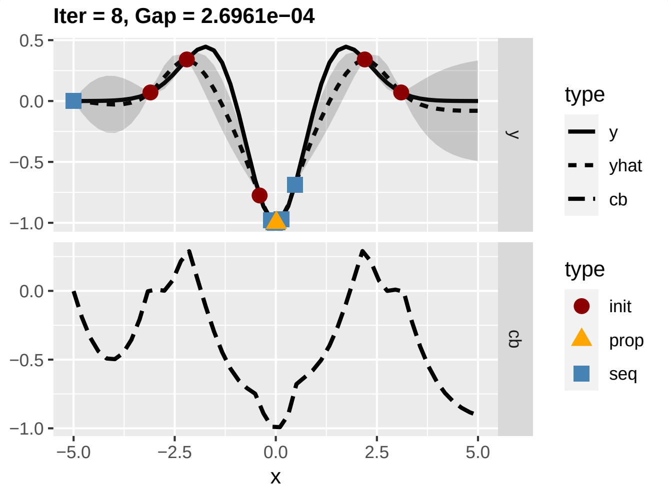
Iteration 6



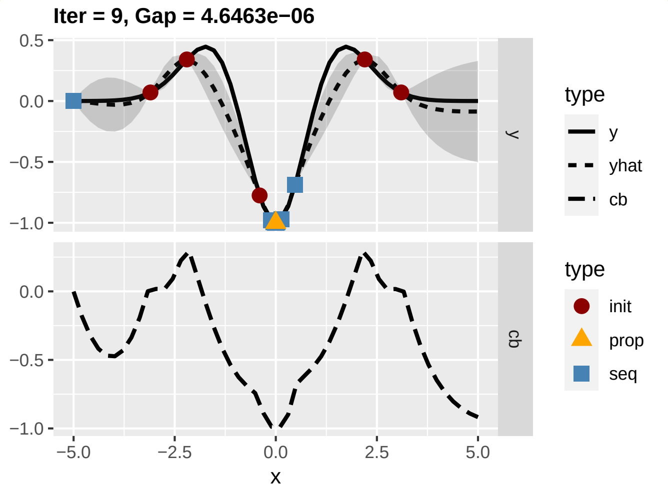
Iter = 7, Gap = 3.2720e−04
0.5
0.0
−0.5
−1.0
y
0.0
−0.4
−0.8
cb
−5.0
−2.5
0.0
2.5
5.0
x
type
y
yhat
cb
type
init
prop
seq

Iteration 7

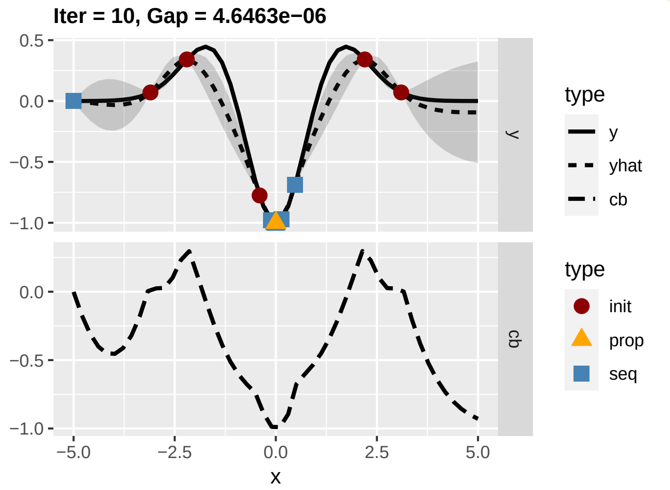
Example: Mexican Hat Function



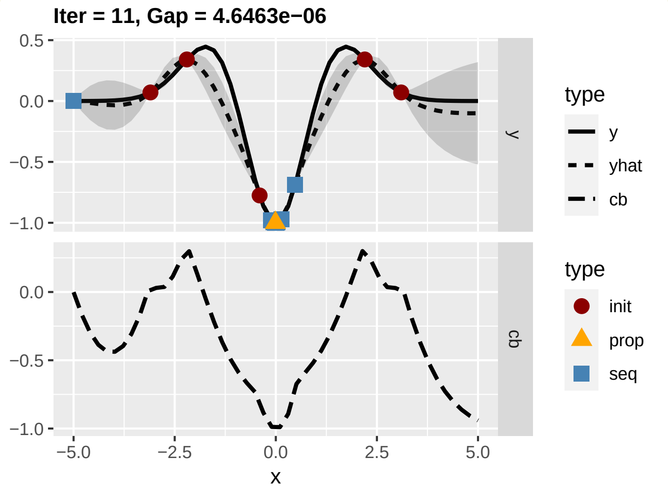
Iteration 8


Iter = 9, Gap = 4.6463e−06
0.5
0.0
−0.5
−1.0
y
cb
−5.0
−2.5
0.0
2.5
5.0
x
type
y
yhat
cb
type
init
prop
seq

Iteration 9

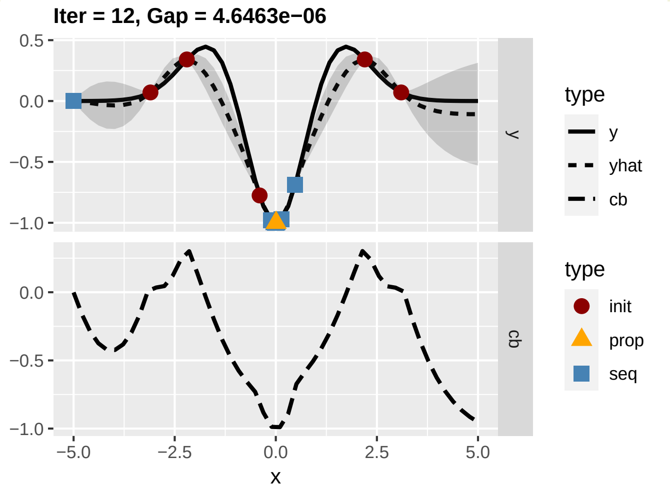


Iteration 10



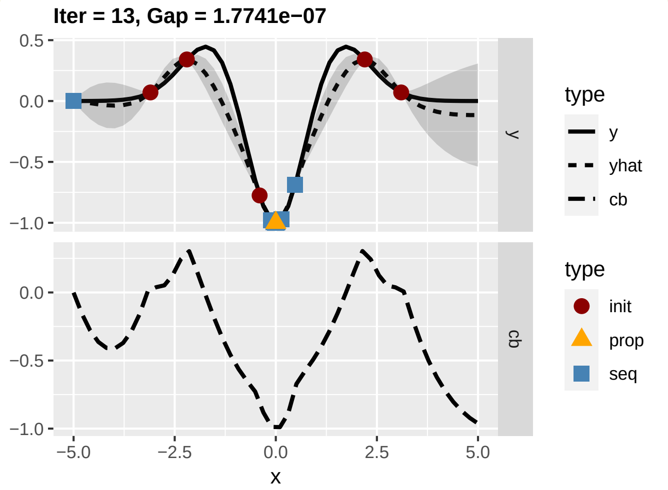
Iteration 11

Example: Mexican Hat Function



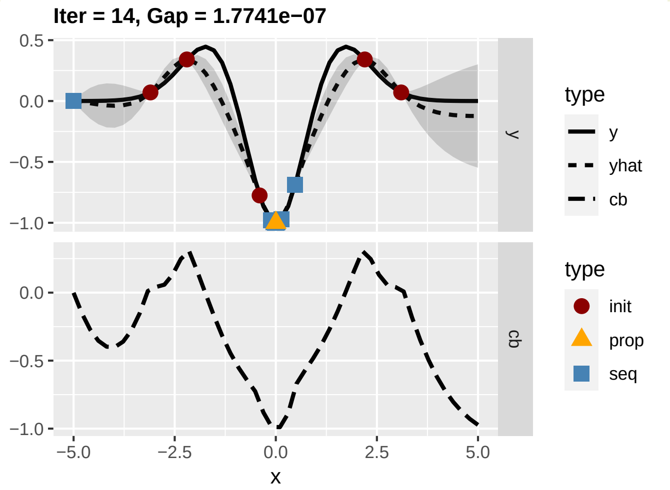
Iteration 12

Example: Mexican Hat Function

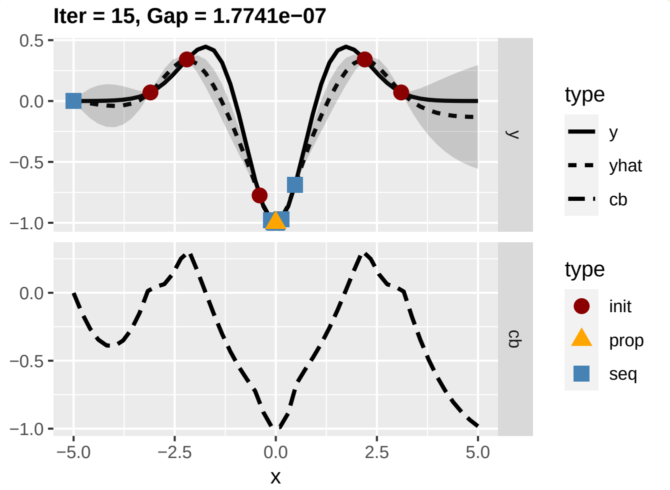


Iteration 13

Example: Mexican Hat Function

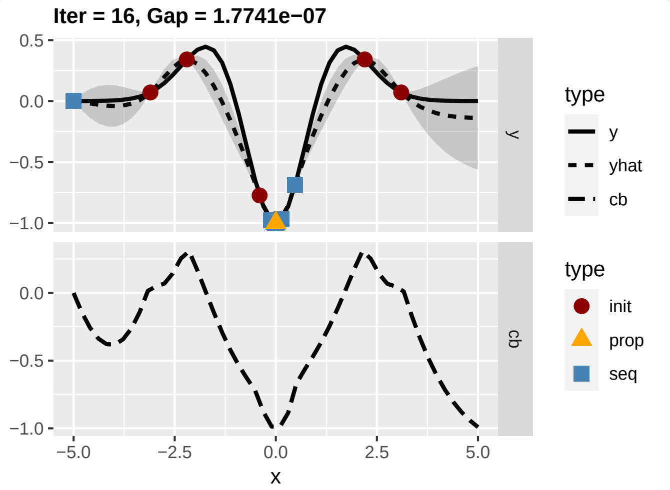


Iteration 14



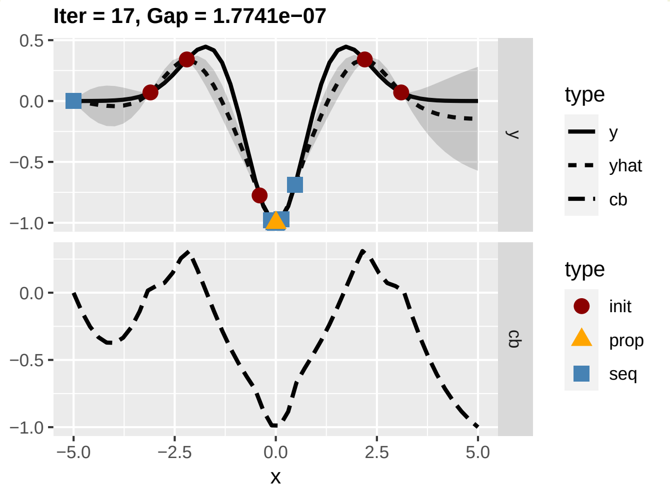
Iteration 15

Example: Mexican Hat Function

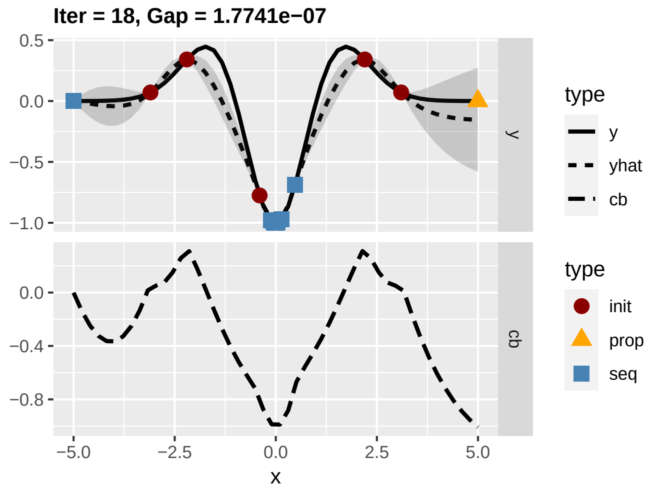


Iteration 16

Example: Mexican Hat Function



Iteration 17

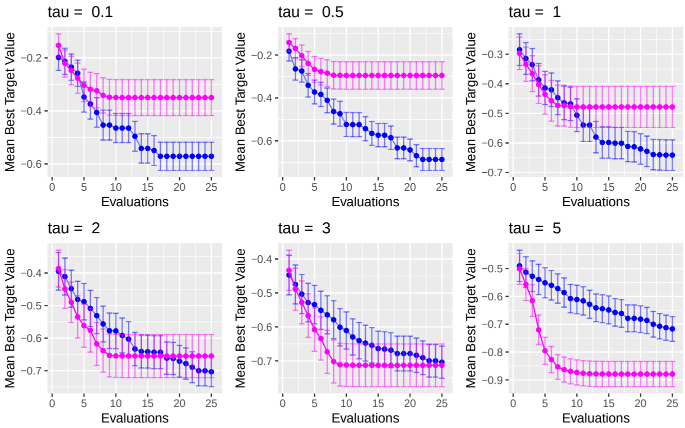


Iteration 18

Weighted ML Estimation

- Idea: Weight by potential gain of information at time of proposal
- Weights:
  - Use variance (standard error) estimation at proposed point
  - Compare to variances at n randomly sampled points
  - Use empirical distribution function $F(\bullet)$
  - The weight w_j of $\boldsymbol{x}_j$ then is $w_j = \frac{F(\boldsymbol{x}_j)}{\sum_i^n F(\boldsymbol{x}_i)}$.
- Estimation:
  - Draw $\boldsymbol{x}_j$ (with replacement) with probability w_j
  - $\hat{\boldsymbol{\theta}}_m(\boldsymbol{X}_t) = \arg\max_{\boldsymbol{\theta}_m} \mathcal{L}(\boldsymbol{\theta}_m|\boldsymbol{X}_t)$, where $\boldsymbol{X}_t$ is the design matrix of the so-generated sample

Weighted ML Estimation – Preliminary Results



Benchmarking of BO with weighted ML (blue) against classic unweighted ML (magenta) on Mexican hat function with varying τ in LCB. 200 BO runs with initial sample size 4. Error bars depict 95-%-CIs.

Agenda

Questions?

Do you have any questions?

- On what type of problems is weighted ML superior to standard ML?
  - Distance from local to global optima
  - Edges?
- Why inferior to standard ML on wiggly functions?
- Benchmark GLCB (and weighted ML?) against “integrated acquisition function” [Snoek et al., 2012], i.e. against (improper) uniform hyperpriors[2]

[2] This is the native bayesian way to represent uncertainty over the prior. However, note that such a uniform prior reflects *indifference* rather than *ignorance*.

Benavoli, A. and Zaffalon, M. (2015).
Prior near ignorance for inferences in the k-parameter exponential family.
Statistics, 49(5):1104–1140.

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J., and Lang, M. (2017).
mlrmbo: A modular framework for model-based optimization of expensive black-box functions.
arXiv preprint arXiv:1703.03373.

Bossek, J. (2017).
smoof: Single- and multi-objective optimization test functions.
The R Journal.

Mangili, F. (2015).
A prior near-ignorance Gaussian process model for nonparametric regression.
In *ISIPTA '15: Proceedings of the 9th International Symposium on Imprecise Probability: Theories and Applications*, pages 187–196.

Rasmussen, C. E. (2003).
Gaussian processes in machine learning.
In *Summer school on machine learning*, pages 63–71. Springer.

Snoek, J., Larochelle, H., and Adams, R. P. (2012).
Practical Bayesian optimization of machine learning algorithms.
In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2*, pages 2951–2959.

Wahab, H., Jain, V., Tyrrell, A. S., Seas, M. A., Kotthoff, L., and Johnson, P. A. (2020).
Machine-learning-assisted fabrication: Bayesian optimization of laser-induced graphene patterning using in-situ raman analysis.
Carbon, 167:609–619.

Wickham, H. (2016).
ggplot2: Elegant Graphics for Data Analysis.
Springer-Verlag New York.