Federico Croppi

Explaining Sequential Model-Based Optimization

Supervisors: Giuseppe Casalicchio, Julia Herbinger, Julia Moosbauer

20.10.2021

Agenda

1. Introduction
2. Our solution
3. Application examples
4. Discussion and conclusion
5. References
6. Appendix

Agenda

1. Introduction
2. Our solution
3. Application examples
4. Discussion and conclusion
5. References
6. Appendix

Motivation - aim of the project

- **Research gap**: the AF's choice in SMBO is not transparent $\rightarrow$ **why** are specific parameter values chosen in each iteration?
- **Idea**:
  1. distribute the desirability/utility of a proposal among its parameters
  2. further, split their desirability into mean and uncertainty effect
- **Solution**: Shapley value mainly because
  1. in a few minutes...
- **Main contribution**: provide a tool, based on SV, ready to use for SMBO users to explain the proposals of the AF (`ShapleyMBO`)
- **Goal**: increase transparency of SMBO

- HPO as an independent research field has various HPI that could be adapted to our purposes

  1. Local Parameter Importance [1]: $contr_{LPI}(j) = \frac{Var_{a \in \Theta_j}\, \hat{g}(\boldsymbol{\theta}[\theta_j = a])}{\sum_{l \in P} Var_{b \in \Theta_l}\, \hat{g}(\boldsymbol{\theta}[\theta_l = b])}$
     Problem: $contr_{LPI}(j) \in \mathbb{R}_0^+$ ($\lightning$EETO)

  2. Ablation Analysis [2]: $contr_{AA}(j) = \hat{g}(\boldsymbol{\theta}^s) - \hat{g}(\boldsymbol{\theta}^s[\theta_j^s = \theta_j^t])$
     Problem: no consideration of interactions $\rightarrow$ an "avenue for further work is to make support for complex parameter interdependencies more flexible, for example [...] **to allow sets of parameters without conditional relationships to be modified in the same ablation round**" [2, p.456].

- Assume grand coalition P, contribution function v and game (P, v). Further, assume that $\Pi(P)$ is the set of all permutations of P with $\pi \in \Pi(P)$, and $Pre_\pi(j)$ is the coalition consisting of the predecessors of player j.

$$
\begin{aligned}
contr_{SV}(j) = \phi_j(v) &= \frac{1}{p!} \sum_{\pi \in \Pi(P)} v(Pre_\pi(j) \cup \{j\}) - v(Pre_\pi(j)) \\
&= \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!\,(p-1-|S|)!}{p!} \left[v(S \cup j) - v(S) \right]
\end{aligned}
$$

\+ allows also negative contributions ($contr_{SV}(j) \in \mathbb{R}$)

\+ incorporate interactions and interacting features are equally remunerated for the worth the coalition

Agenda

1. Introduction
2. Our solution
3. Application examples
4. Discussion and conclusion
5. References
6. Appendix

- the AF is nothing but a transformed surrogate model
- **Turning point:** CB criterion together with the Linearity axiom are the perfect solution[1]:

$$\phi(cb) = \phi(m - \lambda \cdot se) = \phi(m) - \lambda \cdot \phi(se)$$

Up today we only had information on configuration level. Now, with the SV and the CB we can dig deeper and $\Rightarrow$

1. assess the overall desirability of the chosen parameters
2. understand **why** parameters are desirable, bringing to light previously hidden aspects of the EETO

[1] assuming *min* problems CB becomes LCB

Our solution - ShapleyMBO

Algorithm 1 `ShapleyMBO`

Require: SMBO result object *mbo*, iteration of interest t, sample size K

get explicand from *mbo*: $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}_t^{new}$
sample $1000 \cdot p$ points $\boldsymbol{Z}$ from $\boldsymbol{\Theta}$ to approximate the space
compute $\hat{\phi}(m) = (\hat{\phi}_1(m), \dots, \hat{\phi}_p(m))$:
    get SM from *mbo*: $\hat{f}_m = \hat{f}_t^{mean}$
    explain $\tilde{\boldsymbol{\theta}}$ with `iml::Shapley()` using $\boldsymbol{Z}$, $\hat{f}_m$ and K
compute $\hat{\phi}(se) = (\hat{\phi}_1(se), \dots, \hat{\phi}_p(se))$:
    get SM from *mbo*: $\hat{f}_{se} = \hat{f}_t^{uncertainty}$
    explain $\tilde{\boldsymbol{\theta}}$ with `iml::Shapley()` using $\boldsymbol{Z}$, $\hat{f}_{se}$ and K
compute $\hat{\phi}(cb)$ with linearity axiom:
$\hat{\phi}(cb) = \hat{\phi}(m) - \lambda\hat{\phi}(se)$

- built on {`mlrMBO`} and {`iml`}
- supports: Single-objective, Single-point, Min-problems, {`mlrMBO`} built-in infill criteria (*), decomposition only for LCB (using `contribution = T`)
- plot results with `plotShapleyMBO` here

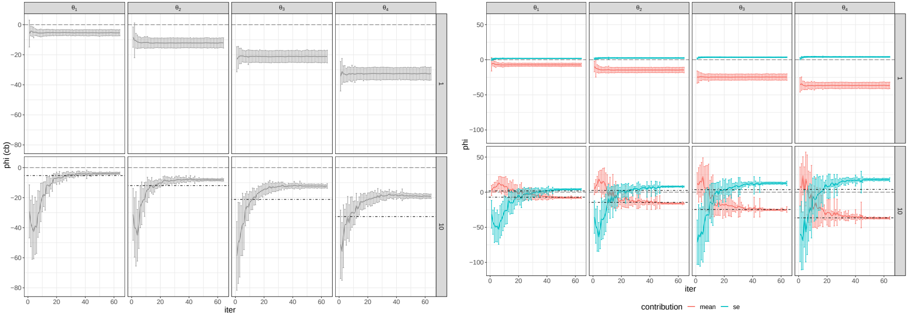
- can be found with `checkSampleSize` (later)
- recreate global AF search, sampling population for SV estimation, generates $\mathbb{E}[v(\boldsymbol{\Theta})]$ GS
- same Shapley objects (except for prediction function), otherwise recreate wrong *cb* contributions (achieved with seeds setting)
- create custom prediction function in {`iml`}, that predicts uncertainty
- implemented by internal function `computePhiCb`

1. Introduction
2. Our solution
3. Application examples
4. Discussion and conclusion
5. References
6. Appendix

1. Application on Hyper-Ellipsoid: Test function with known analytical description
   - Does `ShapleyMBO` deliver consistent results?
   - Does `ShapleyMBO` react to different LCB settings?
2. Application on MLP: real tuning example
   - show how users might benefit from `ShapleyMBO`

(*) LCB is minimized $\rightarrow \hat{cb} < \bar{cb}$. For a better intuition, when $\phi_j(cb) < 0$, we say that the contribution is **positive**. The analogous logic is applied to m and se contributions.

Hyper-Ellipsoid: results - desirability paths I



$$\mathcal{P}(cb) = (\hat{m} - \bar{m}) + \lambda(\bar{se} - \hat{se})$$

Hyper-Ellipsoid: results - desirability paths II

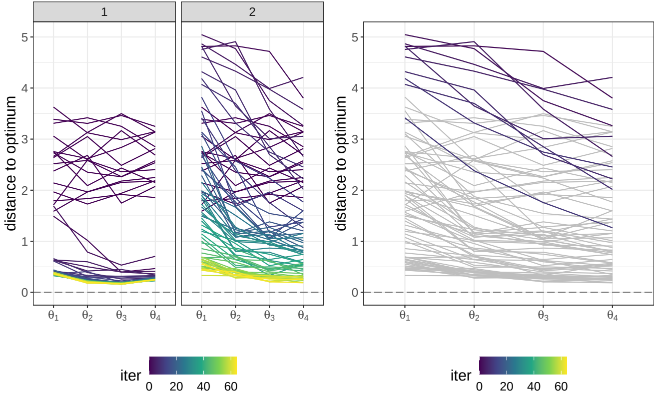


Figure: Design paths for both λ (left $\lambda = 1$, right $\lambda = 10$). For each parameter the average distance of the actual parameter value to its optimum $\theta_j^* = 0$ is displayed. Right Plot displays $= 10$ with only 10 iterations highlighted

Hyper-Ellipsoid: results - desirability paths III

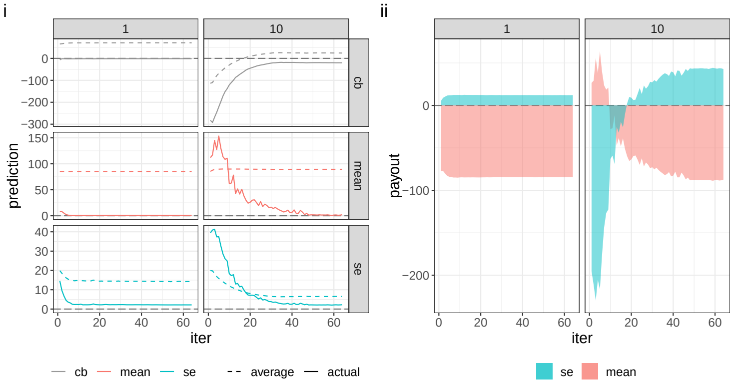


Figure: Payout paths for the Hyper-Ellipsoid optimization: (i) individually (*se* not scaled, (ii) *m* and *se* together, centered around their average prediction, *se* scaled

$$\mathcal{P}(cb) = (\hat{m} - \bar{m}) + \lambda(\bar{se} - \hat{se})$$

- **Idea**: find a sufficiently high sample size by greedy forward search
- contributions come with an efficiency error $\Delta^K_{eff}(v)$
- for $K \uparrow \Rightarrow \Delta^K_{eff}(v) \downarrow$
- **Approximations problematic** when ranking changes after redistributing $\Delta^K_{eff}(v)$
- **Solution**: error "position" is unclear, but ...
- it is sufficient to test $\Delta^K_{eff}(v)$ against the smallest contributions' distance between two parameters $\delta^K(v)$ (threshold) to check K
  1. if ranking **does not change** using the smallest distance → ranking **can not change** among multiple parameters either
- **Rule**: if $\Delta^K_{eff}(v) < \delta^K(v) \Rightarrow$ ranking can not change after correction $\Rightarrow K$ is sufficiently high

MLP: excursus - `checkSampleSize`

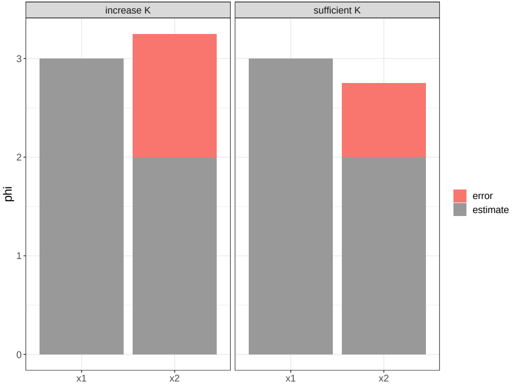
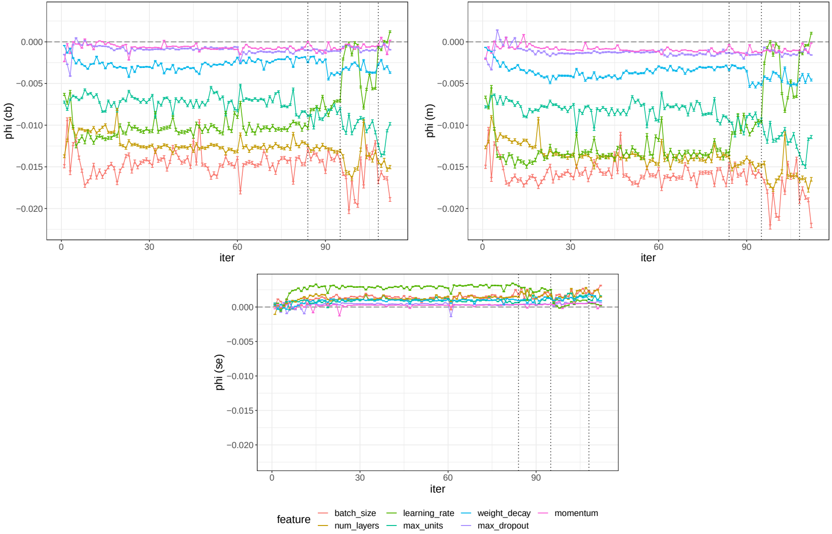
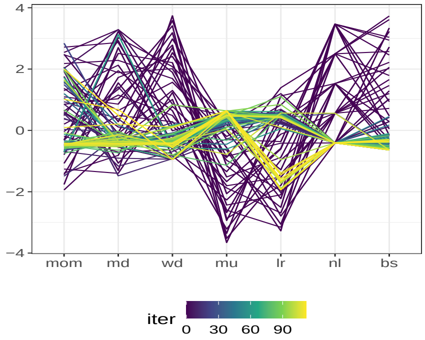
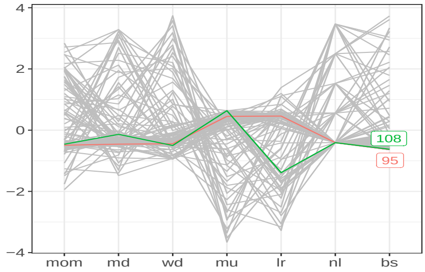


Figure: Example `checkSampleSize`

MLP: results - desirability paths I







iter	lr	$\bar{lr}$	error	$\hat{m}$	$\hat{se}$	$\hat{\phi}^{rel}_{lr}(cb)$	$\hat{\phi}^{rel}_{lr}(m)$	$\hat{\phi}^{rel}_{lr}(se)$
95	0.0079	0.007	0.2222	0.2226 (3.6)	0.0003 (8.0)	0.15	0.17	0.27
108	0.0041	0.007	0.2225	0.2229 (12.5)	0.0011 (66.1)	0.01	0.02	0.06

Agenda

1. Introduction
2. Our solution
3. Application examples
4. Discussion and conclusion
5. References
6. Appendix

Sampling strategy determines the average prediction and hence, indirectly, also the contributions (ShapleyMBO)

Global sampling

- \+ **recreates the global AF search**
- \+ stable desirability paths
- − mean contributions positively and se contributions negatively biased

⇒

1. awareness of GS effects → **globally** undesirable parameters
2. use additional materials for plausibility checks and interpretation of results
3. implement local sampling to further investigate results

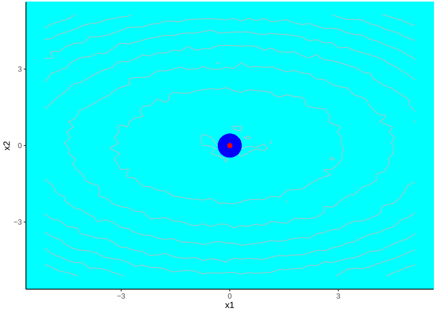


Figure: explicand (red), local (blue), global (cyan)

- Both analyses suggests that the SV is a valid method and `ShapleyMBO` is a diagnostic tool of great potential
- This thesis only set the first milestone and there are several improvements in which we should invest resources
  1. more exhaustive validation phase
  2. implement other sampling strategies
  3. implement the BS
  4. more research on mean contributions as HPI
  5. extend decomposition to other AF, e.g. the EI

$$EI(\boldsymbol{\theta}) = (\Psi^{min} - m(\boldsymbol{\theta}))\Phi\left(\frac{\Psi^{min} - m(\boldsymbol{\theta})}{se(\boldsymbol{\theta})}\right) + se(\boldsymbol{\theta})\phi\left(\frac{\Psi^{min} - m(\boldsymbol{\theta})}{se(\boldsymbol{\theta})}\right)$$

We sincerely hope that with this project we contributed, although infinitesimally, to the improvement of the SMBO transparency.

Thank you for your attention.

Still some time left?

- used to display the results of `ShapleyMBO`
- supports various options:
  1. single (bar-plot) and multiple iterations (line-plot, so called *desirability paths*)
  2. uncertainty estimates using CI intervals $CI_{1-\alpha} = [\hat{\phi}_j \pm t_{(1-\frac{\alpha}{2}, K-1)} \frac{\hat{\sigma}_j}{\sqrt{K}}]$, with $\alpha = \{0.01, 0.05, 0.1\}$
  3. various plot combinations via `decomp` argument, each plot can also be used/saved/modified individually {`patchwork`}
  4. `contribution = `{`T, F`}

Back to ShapleyMBO

plotShapleyMBO - single iteration output

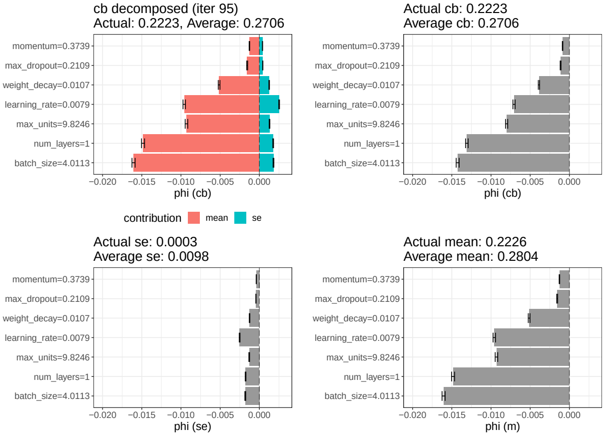


Figure: This figure shows how the complete output of `plotShapleyMBO` for single iteration looks. Here, we show the results of `ShapleyMBO` in the MLP application in iteration 95.

introduced by Sundararajan and Najmi [4]

$$v(S) = \hat{f}(\tilde{\boldsymbol{\theta}}_S; \boldsymbol{\theta}'_{P \setminus S}) \Rightarrow v(\emptyset) = \hat{f}(\boldsymbol{\theta}')$$

+ complementary results interpretation
+ no independence assumption

− AF choice explained relative to one configuration only
− baseline choice can be complicated, but

⇒ special case of SMBO for HPO

1. default baseline provided by libraries
2. adaptive baseline using incumbent configuration

Discussion - mean contributions

- Idea: We could use the mean contributions as a HPI metric
- Problem: results strongly on the quality of the surrogate models
- The aim of the project was to explain the choices of AF, hence using the actual SM is the best way to do that. But, using the SM to generalize importance scored is dangerous
- Use mean contributions as HPI with care!

	bs	*md*	*mu*	*nl*	*lr*	*mom*	*wd*
$\hat{\phi}(m)_{sm_{95}}$	-0.016	-0.002	-0.009	-0.015	-0.010	-0.001	-0.005
$\hat{\phi}(m)_{sm_{113}}$	-0.015	-0.002	-0.012	-0.016	-0.007	-0.001	-0.005

Table: Comparison of the mean contributions of the best-predicted proposal (iteration 95) in the MLP tuning example using the actual surrogate model (iteration 95) and the final surrogate model.

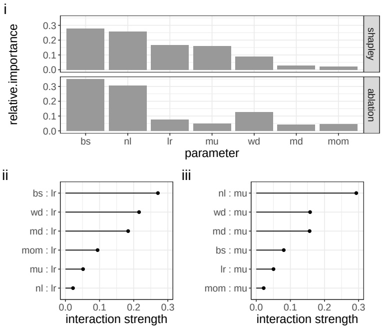


Figure: Relative *m* contribution of the parameters for configuration proposed in iteration 95 (plot i). To understand the difference in the results the interaction strength of learning rate (plot ii) and max units (plot iii) with other parameters is computed using Friedman's H-Statistic (interaction between two features).

	bs	*nl*	*lr*	*mu*	*wd*	*md*	*mom*
relative SV	0.278	0.257	0.166	0.161	0.089	0.027	0.022
relative AA	0.348	0.307	0.076	0.050	0.128	0.044	0.047

1. Introduction
2. Our solution
3. Application examples
4. Discussion and conclusion
5. References
6. Appendix

[1] A. Biedenkapp, J. Marben, M. Lindauer, and F. Hutter. Cave: Configuration assessment, visualization and evaluation. In *International Conference on Learning and Intelligent Optimization*, pages 115–130. Springer, 2018.

[2] C. Fawcett and H. H. Hoos. Analysing differences between algorithm configurations through ablation. *Journal of Heuristics*, 22(4):431–458, 2016.

[3] V. Picheny, T. Wagner, and D. Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3):607–626, 2013.

[4] M. Sundararajan and A. Najmi. The many shapley values for model explanation. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9269–9278, Virtual, 13–18 Jul 2020. PMLR.

Agenda

1. Introduction
2. Our solution
3. Application examples
4. Discussion and conclusion
5. References
6. Appendix

Algorithm 2 SMBO basic procedure I

create an initial design $\mathcal{D} = \{(\boldsymbol{\theta}^{(i)}, \Psi^{(i)})\}_{i=1}^{n_{init}}$
while termination criterion is not fulfilled **do**
  fit a surrogate model $\hat{f}$ on design $\mathcal{D}$
  propose $\boldsymbol{\theta}^{new} =_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} u(\boldsymbol{\theta}|\mathcal{D})$
  evaluate Ψ on $\boldsymbol{\theta}^{new}$ and **update** $\mathcal{D} \leftarrow \mathcal{D} \cup (\boldsymbol{\theta}^{new}, \Psi(\boldsymbol{\theta}^{new}))$
end while

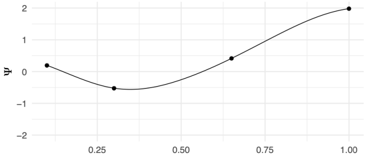
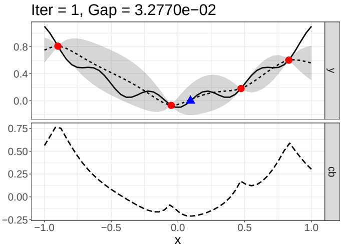
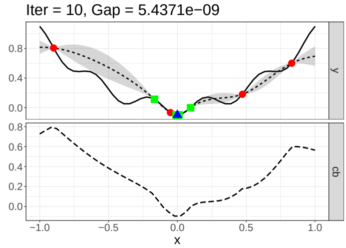


Figure: surrogate model fit, credits: Fortgeschrittene Computerintensive Methoden - lecture slides, Bischl B. and Moosbauer J. (2019)

SMBO - basic procedure II





Suppose u is minimized and predicted with a RF model, source and target are respectively $\boldsymbol{\theta}^s = (0.5, 0.5, 0.5)^T$ and $\boldsymbol{\theta}^t = (0, 0, 0)^T$.

$$u = \theta_1 + \theta_2 \cdot \theta_3 + \epsilon$$

$$\theta_1, \theta_2, \theta_3 \overset{i.i.d}{\sim} \mathcal{U}(0, 1),\ \epsilon \sim \mathcal{N}(0, 0.05^2),$$

We expect similar contributions for θ_2 and θ_3, and yet

parameter	aa round 1	relative aa	relative sv
θ_1	0.483	0.66	0.66
θ_2	0.262	0.31	0.21
θ_3	0.258	0.03	0.17

$\Rightarrow$ unfair contribution of θ_3, which has a very similar, unconditional effect in the first round

- $contr_{SV} \in \mathbb{R}$, better to explain EETO
- $contr_{SV}$ considers all possible interactions

parameter	aa round 1	relative aa	relative sv
θ_1	0.483	0.66	0.66
θ_2	0.262	0.31	0.21
θ_3	0.258	0.03	0.17

$\Rightarrow$ fair contribution of θ_3!

- the SV is a better method for the purposes of our project, **in addition**...

get back to ▸▸ SV theory

Dummy player: If $v(S \cup \{j\}) - v(S) = v(j)$ for player j and all $S \subseteq P \backslash \{j\}$, then $\phi_j(v) = v(j)$.

Efficieny: $\sum_{j=1}^{p} \phi_j(v) = v(P) - v(\emptyset)$

Linearity: Given two games (P, v_1) and (P, v_2), for $a, b \in \mathbb{R}$ it holds

$$\phi_j(av_1 + bv_2) = a\phi_j(v_1) + b\phi_j(v_2)$$

Symmetry: If $v(S \cup \{j\}) = v(S \cup \{l\})$ for players j, l and every $S \subseteq P \backslash \{j, l\}$, then $\phi_j(v) = \phi_l(v)$

Algorithm 3 Estimation of the CES

Require: explicand $\tilde{\boldsymbol{\theta}}$, feature index j, model $\hat{f}$ and sample size K

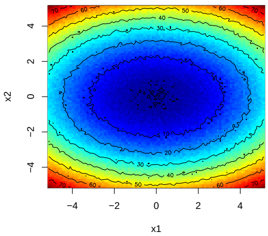
for $k = 1 \rightarrow K$ **do**
    sample (at random and with replacement) an instance $\boldsymbol{z} \in \boldsymbol{\Theta}$
    sample (at random and with replacement) an order $\pi \in \Pi(P)$
    order $\tilde{\boldsymbol{\theta}}$ and $\boldsymbol{z}$ according to π
        $\tilde{\boldsymbol{\theta}}_\pi = (\tilde{\theta}_{(1)}, \ldots, \tilde{\theta}_{(p)})$
        $\boldsymbol{z}_\pi = (z_{(1)}, \ldots, z_{(p)})$
    construct two new instances
        $\tilde{\boldsymbol{\theta}}_{+j} = (\tilde{\theta}_{(1)}, \ldots, \tilde{\theta}_{(j-1)}, \tilde{\theta}_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
        $\tilde{\boldsymbol{\theta}}_{-j} = (\tilde{\theta}_{(1)}, \ldots, \tilde{\theta}_{(j-1)}, z_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
    $\hat{\phi}_j^k(v) = \hat{f}(\tilde{\boldsymbol{\theta}}_{+j}) - \hat{f}(\tilde{\boldsymbol{\theta}}_{-j})$
end for
$\hat{\phi}_j(v) = \frac{1}{K} \sum_{k=1}^{K} \hat{\phi}_j^k(v)$

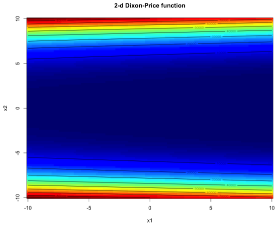
In both analyses, where possible we use default SMBO setting

hyperparameter	value
min objective function	TRUE
noisy objective	TRUE
initial design	size $4p$ sampled with maximin LHS
surrogate model	GP regression
kernel function	$\frac{3}{2}$-Matérn
acquisition function	LCB
min acquisition function(*)	TRUE
infill optimizer	focussearch $(n_r = 3, n_i = 5, n_p = 1000)$
termination condition	max. evaluations $20p$[3, p.614]

Why the 4p Hyper-Ellipsoid

- scalable dimensions: $4p$ is not too low and not too high
- intuitive functional form: clear *mean* contribution
- same parameter's domain: clear *se* contributions
- convex, clear optimum region and smooth: algorithmic paths somehow predictable (smooth convergence), useful for expectations on *cb* contributions



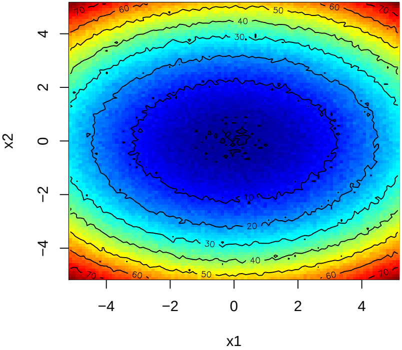


get back to Hyper-Ellipsoid

Hyper-Ellipsoid - experimental setup

$$f(\boldsymbol{\theta}) = \sum_{j=1}^{4} j \cdot \theta_j^2 \ , \ \theta_j \in [-5.12, 5.12] \text{ for}$$

$$\boldsymbol{\theta}^* = (0,0,0,0)^T \text{ and } f(\boldsymbol{\theta}^*) = 0.$$



- $\epsilon \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$ observation noise added, where $\sigma = 0.05 \cdot \hat{\sigma}_{HE}$ [3, p.613]
- $\lambda \in \{1, 10\}$
- 30 runs for each λ
- $K = 1000$

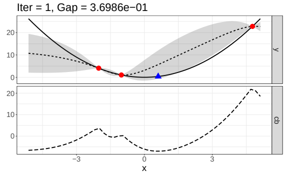
- Although (i) contributions depends on proposed values and (ii) each dimensions might be explored and exploited differently, we tried to formulate general expectations
- For $\lambda = \{1, 10\}$: similar $\phi(m)$, $\phi(se)$ stronger for $= 10$

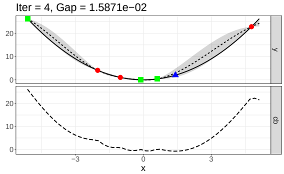
$$f(\boldsymbol{\theta}) = 1\theta_1^2 + 2\theta_2^2 + 3\theta_3^2 + 4\theta_4^2$$

$$\boldsymbol{\theta}^* = (0, 0, 0, 0)^T$$

$$\theta_j \in [-5.12, 5.12]$$

- $\phi(m)$: distance to θ^*, higher parameters more important
- $\phi(se)$: equally important, conditional on dimension exploration
- $\phi(cb)$: convex f, smooth convergence $\rightarrow$ initially $\phi(se)$, then $\phi(m)$





Purpose: compare "similar" points to understand the effect of λ

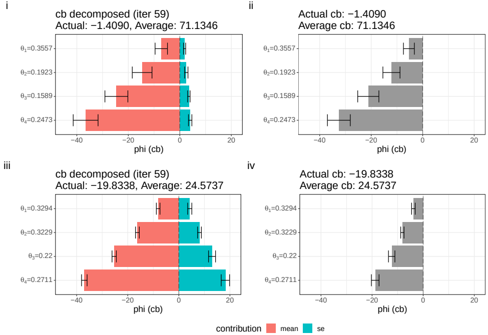
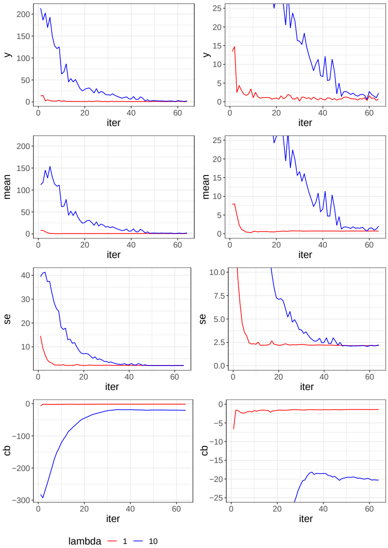


Figure: results in iteration 59 for $\lambda = 1$ on top, for $\lambda = 10$ at the bottom.

	y	$\hat{cb}$	$\bar{cb}$	$\mathcal{P}(cb)$	$\hat{m}$	$\bar{m}$	$\mathcal{P}(m)$	$\hat{se}$	$\bar{se}$	$\mathcal{P}(se)$
$\lambda = 1$	0.34	-1.41	71.13	-72.54	0.73	85.38	-84.64	2.14	14.25	12.11
$\lambda = 10$	0.66	-19.83	24.57	-44.40	0.72	89.47	-88.75	2.06	6.49	44.3


y
mean
se
cb
iter
lambda
1
10

Hyper-Ellipsoid: results - other material

		θ_1	θ_2	θ_3	θ_4
$\lambda = 1$	$\hat{\phi}(cb)$	-5.35 (2.12)	-12.14 (3.33)	-21.12 (4.11)	-32.56 (4.44)
	$\hat{\phi}(m)$	-7.20 (2.4)	-14.64 (3.83)	-24.63 (4.46)	-36.58 (4.83)
	$\hat{\phi}(se)$	1.85 (0.42)	2.50 (0.66)	3.51 (0.47)	4.02 (0.67)
$\lambda = 10$	$\hat{\phi}(cb)$	-3.82 (0.7)	-8.14 (0.65)	-12.32 (1.26)	-18.77 (1.5)
	$\hat{\phi}(m)$	-8.09 (0.71)	-16.22 (0.75)	-25.31 (0.84)	-36.99 (1.09)
	$\hat{\phi}(se)$	4.27 (0.81)	8.08 (0.74)	12.99 (1.37)	18.22 (1.64)

Table: Contributions in iteration 59 for both λ

	with *id*	θ_1	θ_2	θ_3	θ_4
$\lambda = 1$	yes	0.95	0.99	1.01	0.98
$[1-58]$	no	0.24	0.15	0.08	0.07
$\lambda = 10$	yes	1.29	1.26	1.15	1.07
$[1-58]$	no	1.40	1.31	1.11	0.97

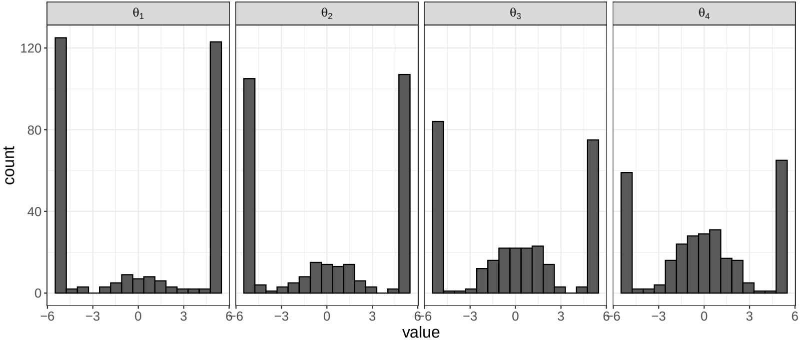
Table: Exploration of each dimension up to iteration 58 (iteration 1 to 58 are included). Column *with id* indicates if the initial design was included or not. The exploration is measured as the standard deviation of the average distance of the configurations from its optimal value $\theta_j^* = 0$.

		θ_1	θ_2	θ_3	θ_4
$\lambda = 1$	$\hat{\phi}(cb)$	-5.29 (2.22)	-11.97 (3.57)	-21.6 (4.41)	-32.59 (4.41)
$[1-64]$	$\hat{\phi}(m)$	-7.10 (2.53)	-14.44 (4.09)	-24.66 (4.54)	-36.57 (4.81)
	$\hat{\phi}(se)$	1.81 (0.48)	2.47 (0.68)	3.50 (0.52)	3.98 (0.70)
$\lambda = 10$	$\hat{\phi}(cb)$	-33.04 (16.06)	-30.35 (14.83)	-37.68 (16.62)	-38.88(14.21)
$[1-10]$	$\hat{\phi}(m)$	8.43 (9.06)	10.97 (15.62)	9.05 (27.14)	2.42 (34.65)
	$\hat{\phi}(se)$	-41.47 (21.48)	-41.33 (26.12)	46.73 (37.29)	41.3 (39.38)
$\lambda = 10$	$\hat{\phi}(cb)$	-3.81(0.77)	-8.12(0.8)	-12.31(1.31)	-19.01(1.54)
$[55-64]$	$\hat{\phi}(m)$	-7.79 (1.16)	-16.12 (1.01)	-25.09 (1.47)	-39.96 (1.27)
	$\hat{\phi}(se)$	3.98 (1.06)	8.00 (0.89)	12.78 (1.64)	17.94 (1.76)

Table: Contributions averaged over multiple iterations in the process. Range in brackets below λ indicates the iterations included.

back to desirability paths

Hyper-Ellipsoid: results - other material

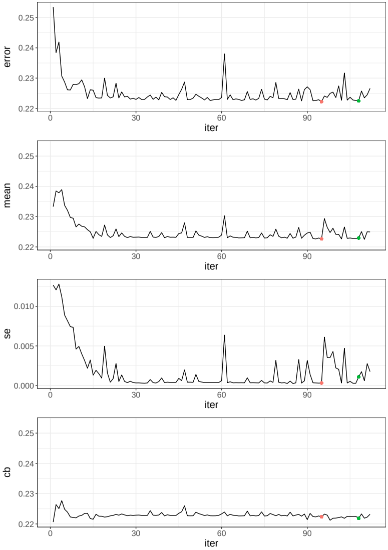


MLP: background information

- real application example: tuning of a multilayer perceptron for a speech recognition classification task using SMBO with LCB and $\lambda = 1$ (*min* validation error)

HP	Notation	Type	Lower	Upper	Trafo
batch size	*bs*	numeric	$\log_2 16$	$\log_2 512$	2^x
max dropout	*md*	numeric	0	1	
max units	*mu*	numeric	$\log_2 64$	$\log_2 1024$	2^x
number of layers	*nl*	integer	1	5	
learning rate	*lr*	numeric	0	0.01	
momentum	*mom*	numeric	0.1	1	
weight decay	*wd*	numeric	0	0.1	

- SV estimation with $K = 20000$ found with `checkSampleSize` among $\{100, 1000, 5000, 10000, 15000, 20000\}$

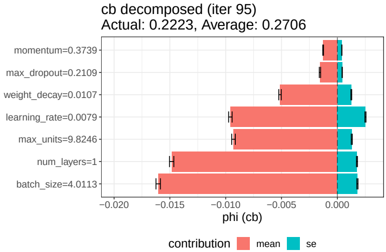


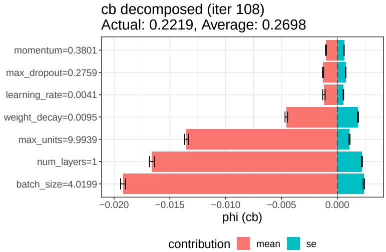
error
0.25
0.24
0.23
0.22
0
30
60
90
iter
mean
0.25
0.24
0.23
0.22
0
30
60
90
iter
se
0.010
0.005
0.000
0
30
60
90
iter
cb
0.25
0.24
0.23
0.22
0
30
60
90
iter

	bs	*nl*	*mu*	*lr*	*wd*	*md*	*mom*
$\hat{\phi}(cb)$	-0.014	-0.013	-0.008	-0.007	-0.004	-0.001	-0.001
$\hat{\phi}(m)$	-0.016	-0.015	-0.009	-0.010	-0.005	-0.002	-0.001
$\hat{\phi}(se)$	0.002	0.002	0.001	0.003	0.001	0.000	0.000

Table: Contributions of the parameters in iteration 95 for the MLP optimization problem.Payout $\mathcal{P}$ for cb, m and se are respectively $-0.048, -0.058, 0.09$.

MLP: results - other material



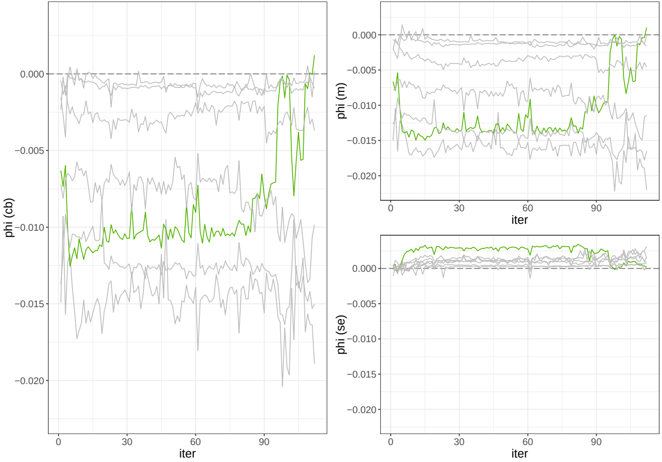


Algorithm 4 `checkSampleSize`

Require: `ShapleyMBO` results for $\tilde{\boldsymbol{\theta}}_t$ with size K, models $\hat{f}^v$ with $v = \{cb, m, se\}$

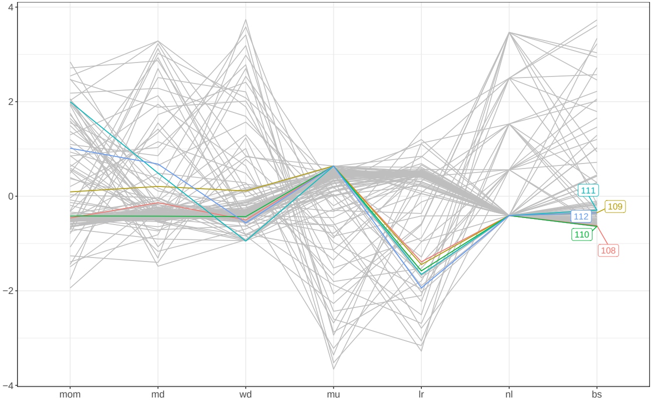
for w in v **do**
  compute payout $\mathcal{P}(w)$, error $\Delta_{eff}^K(w)$ and threshold $\delta^K(w)$
  if $\Delta_{eff}^K(w) < \delta^K(w)$ **then**
    $K_w = \mathtt{T}$
  end if
  if $\Delta_{eff}^K(w) \geq \delta^K(w)$ **then**
    $K_w = \mathtt{F}$
  end if
end for
if $(K_{cb}, K_m, K_{se}) = (\mathtt{T}, \mathtt{T}, \mathtt{T})$ **then**
  K is high enough
else if $(K_{cb}, K_m, K_{se}) \neq (\mathtt{T}, \mathtt{T}, \mathtt{T})$ **then**
  K should be increased

MLP: *lr* optional plots

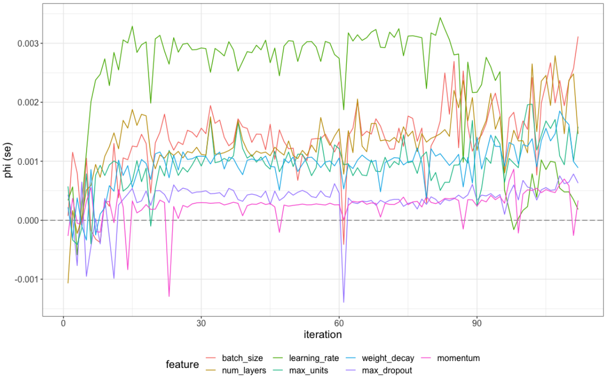


back to ▸▸ lr

MLP: *lr* optional plots



MLP: *lr* optional plots



Why `iml` package?

- umbrella package for many IML tools $\rightarrow$ homogenous results
- "in House" package
- Big potential for future work: IML tools provided show that {`iml`} package can be adapted to the interpretation of BO processes
  $\rightarrow$ Apply `PredictorAf` object to other IML tools

PredictorAf to analyse the AF

- before measuring the SV the {`iml`} package requires a `Predictor` object, which basically contains the data and the ML model
- we created a new object called `PredictorAf`, which inherits from the `Predictor` class and can be used in combination with `Shapley`
- How?
  1. `data` → sampling population
  2. `pred.fun` → acf.fun
  3. some infill criteria (e.g. EI) need design points → new `field` called `design` (correspond to `data` in `Predictor`)
  4. some additional fields
- adjustments from our Cosulting Project