Sensitivity Analysis with Shapley Effects: Computational Issues

Elmar Plischke
Joint Work with E. Borgonovo, G. Rabitti

Institut für Endlagerforschung
TU Clausthal
SAMO 2022, Mar 13, 2022, FSU

Outline

Shapley effects are attracting attention as sensitivity measures.
With the value function being the conditional variance, these effects account for the individual and higher order sensitivity effects.
One of the issues associated with their use is computational cost.
We present new algorithms [Plischke et al., 2021] that offer improvements for the computation of Shapley effects.

Shapley Values

Shapley Effects for Sensitivity Analyis

Example

Shapley Values: Concept from Game Theory

Attribute a fair share of the grand total to d individual players

- Coalition-worth value function $\mathrm{val} : 2^d \to \mathbb{R}_{\geq 0}$, 2^d: set of subsets of $\{1, \ldots, d\}$
- Coalition $\alpha \subset \{1, \ldots, d\}$ lists the active players, anti-coalition $\sim \alpha = \{1, \ldots, d\} \setminus \alpha$
- Marginal contribution of player i joining coalition α: $\mathrm{mar}(\alpha, i) = \mathrm{val}(\alpha \cup \{i\}) - \mathrm{val}(\alpha)$

Möbius inverses

Unique decomposition $\mathrm{val}(\alpha) = \sum_\beta \mathrm{mob}(\beta) u_\beta(\alpha)$
$u_\beta(\alpha) = \mathbf{1}(\beta \subset \alpha)$ (Unanimity game) codes subset inclusion
Weights: Möbius inverses / Harsanyi dividends. Implicitly defined by

$$\mathrm{val}(\alpha) = \sum_{\beta \subset \alpha} \mathrm{mob}(\beta).$$

This system of $2^k - 1$ linear equations can be solved by an inclusion-exclusion rule [Rota, 1964],

$$\mathrm{mob}(\alpha) = \sum_{\beta \subset \alpha} (-1)^{|\alpha|+|\beta|} \mathrm{val}(\beta).$$

Formulas for Shapley values

$$\mathrm{Sh}_i = \frac{1}{k} \sum_{\alpha : i \notin \alpha} \binom{k-1}{|\alpha|}^{-1} \mathrm{mar}(\alpha, i)$$

$$\mathrm{Sh}_i = \frac{1}{k} \sum_{\alpha : i \in \alpha} \binom{k-1}{|\alpha|-1}^{-1} (\mathrm{val}(\alpha) - \mathrm{val}(\sim \alpha))$$

$$\mathrm{Sh}_i = \sum_{\alpha : i \in \alpha} \frac{\mathrm{mob}(\alpha)}{|\alpha|}$$

All three formulas satisfy the axioms (efficiency, symmetry, linearity, null-player property) which uniquely describe the Shapley value [Shapley, 1953].

Computational pathways

- Deterministic: Compute all value functions and use a suitable formula for the Shapley value
- Probabilistic (random subsets may suffice): Interpret Shapley value as
  - weighted average over all marginals
  - average over weights on all paths connecting $(0,\ldots,0)$ with $(1,\ldots,1)$ in the d-dimensional hypercube.

Coalition α: all indices activated in the path before index i becomes active

Computing all combinations suffers from a curse of dimensionality

In this talk: Concentrate on deterministic methods

Shapley Values

Shapley Effects for Sensitivity Analyis

Example

Shapley Effects for Sensitivity Analyis

For use with sensitivity analyis [Owen, 2014, Owen and Prieur, 2017]: Given a deterministic simulator g: $x \mapsto y$, $\mathbb{R}^d \to \mathbb{R}$ and propagate uncertainty via $Y = g(X)$, X d-dim. random vector

- Players: input factors
- Grand total: variance
- Value function: $\text{val}(\alpha) = \mathbb{V}[\mathbb{E}[Y|X_\alpha]]$

Then, val is game with $\text{val}(\emptyset) = 0$, $\text{val}(\{1, \ldots, d\}) = \mathbb{V}[Y]$
Same setup as for variance-based sensitivity analysis: $\text{val}(\{i\})$ is (unnormalized) first order effect, $\text{mob}(\alpha)$ are higher order effects

An inclusion bound for Shapley effects

$S_i = \mathrm{val}(\{i\}) = \mathrm{mob}(\{i\})$ first order effect

$T_i = \sum_{i\in\alpha} \mathrm{mob}(\alpha) = \mathrm{mob}(\{i\}) + \sum_{j\neq i} \mathrm{mob}(\{i,j\}) + \dots$ total effect

An inclusion bound for Shapley effects

$S_i = \mathrm{val}(\{i\}) = \mathrm{mob}(\{i\})$ first order effect

$T_i = \sum_{i\in\alpha} \mathrm{mob}(\alpha) = \mathrm{mob}(\{i\}) + \sum_{j\neq i} \mathrm{mob}(\{i,j\}) + \dots$ total effect

Consider the mean

$$\frac{1}{2}(S_i + T_i) = \mathrm{mob}(\{i\}) + \frac{1}{2}\left(\sum_{j\neq i} \mathrm{mob}(\{i,j\}) + \dots\right)$$

And compare to the Shapley value (Möbius inverse formula)

$$\mathrm{Sh}_i = \mathrm{mob}(\{i\}) + \frac{1}{2}\sum_{j\neq i} \mathrm{mob}(\{i,j\}) + \frac{1}{3}\sum_{k\neq j\neq i} \mathrm{mob}(\{i,j,k\}) + \dots$$

Theorem

If $\mathrm{mob}(\cdot) \geq 0$ *for all index sets then* $S_i \leq \mathrm{Sh}_i \leq \frac{1}{2}(S_i + T_i)$

For independent inputs: $\mathrm{mob}(\cdot) \geq 0$

If only interactions up to order 2 are present in the model, then $\mathrm{Sh}_i = \frac{1}{2}(S_i + T_i)$

Computing Shapley effects: Improving available algorithms

[Song et al., 2016] use the path approach: Deterministic version requires evaluation of the value function along all paths for all factors, $d \cdot d!$

Refinements to improve performance:

1. use a pick-and-freeze design instead of a brute-force double loop;
2. use a duality result for obtaining two estimators at the same computational costs;
3. estimate conditional variances via Sobol'/Saltelli and Jansen formulas;
4. Use a quasi Monte-Carlo design for improved convergence compared to a crude Monte-Carlo design.
5. Traverse all permutations by implicitly generating them (Heap or Steinhaus-Johnson-Trotter)

Estimating the Value Function

Sobol' method / Pick-and-Freeze design with Quasi-Monte Carlo Sampling scheme.

$$\mathbb{V}[\mathbb{E}[Y|X_\alpha]] = \text{Cov}(Y, \mathbb{E}[Y|X_\alpha]) = \text{Cov}(Y, Y')$$

where Y and Y' are identically distributed and conditionally independent given X_α.
Sampling Y and Y' under input independence: pick-and-freeze $Y = g(X_\alpha : X_{\sim\alpha})$, $Y' = g(X_\alpha : X'_{\sim\alpha})$ where X and X' are iid.
Under input dependence: Inverse Knothe-Rosenblatt transformation is required (for Gaussian copulas, one can work with Cholesky decompositions)

Computing Shapley values, I

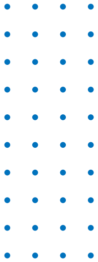
Use $\mathrm{Sh}_i = \frac{1}{k}\left(\sum_{\alpha:i\in\alpha}\binom{k-1}{|\alpha|-1}^{-1}(\mathrm{val}(\alpha)-\mathrm{val}(\sim\alpha))\right)$ [Myerson, 1991]

- Precompute the weights
- for all $\alpha \subset \{1,\ldots,d\}$
  - compute the value function $v = \mathrm{val}(\alpha)$,
  - for all factors i,
    - if $i \in \alpha$ add v with weight for $|\alpha|$ to Sh_i
    - if $i \notin \alpha$ subtract v with weight for $d-|\alpha|$ from Sh_i

Modification: Directly compute $\mathrm{val}(\alpha)-\mathrm{val}(\sim\alpha)$ from pick-and-freeze (efficient under independence)

Interlude: Probabilistic approach for Shapley effects

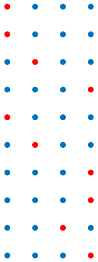
[Goda, 2021] variant of winding stairs design (crawling centipede, each leg moves on its own) to gain marginal contributions for all factors at once.
Pick-and-freeze sampling with randomized insertion



Inserting B sample (X') into A sample (X) :
Distance to A sample output yield different marginal contribution per realisation, add to Shapley values of different factors.
May introduce improvements discussed here in estimating the value function: Duality setting (distance to B sample), QMC.

Interlude: Probabilistic approach for Shapley effects

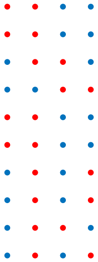
[Goda, 2021] variant of winding stairs design (crawling centipede, each leg moves on its own) to gain marginal contributions for all factors at once.
Pick-and-freeze sampling with randomized insertion



Inserting B sample (X') into A sample (X) :
Distance to A sample output yield different marginal contribution per realisation, add to Shapley values of different factors.
May introduce improvements discussed here in estimating the value function: Duality setting (distance to B sample), QMC.

Interlude: Probabilistic approach for Shapley effects

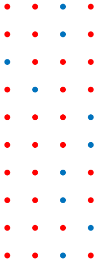
[Goda, 2021] variant of winding stairs design (crawling centipede, each leg moves on its own) to gain marginal contributions for all factors at once.
Pick-and-freeze sampling with randomized insertion



Inserting B sample (X') into A sample (X) :
Distance to A sample output yield different marginal contribution per realisation, add to Shapley values of different factors.
May introduce improvements discussed here in estimating the value function: Duality setting (distance to B sample), QMC.

Interlude: Probabilistic approach for Shapley effects

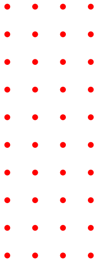
[Goda, 2021] variant of winding stairs design (crawling centipede, each leg moves on its own) to gain marginal contributions for all factors at once.
Pick-and-freeze sampling with randomized insertion



Inserting B sample (X') into A sample (X) :
Distance to A sample output yield different marginal contribution per realisation, add to Shapley values of different factors.
May introduce improvements discussed here in estimating the value function: Duality setting (distance to B sample), QMC.

Interlude: Probabilistic approach for Shapley effects

[Goda, 2021] variant of winding stairs design (crawling centipede, each leg moves on its own) to gain marginal contributions for all factors at once.
Pick-and-freeze sampling with randomized insertion



Inserting B sample (X') into A sample (X) :
Distance to A sample output yield different marginal contribution per realisation, add to Shapley values of different factors.
May introduce improvements discussed here in estimating the value function: Duality setting (distance to B sample), QMC.

Computing Shapley values, II

Using $\mathrm{Sh}_i = \sum_{\alpha: i \in \alpha} \frac{\mathrm{mob}(\alpha)}{|\alpha|}$ [Grabisch, 2006, Owen, 2014]: fast Möbius inverse needed

Computing Shapley values, II

Using $\mathrm{Sh}_i = \sum_{\alpha: i\in\alpha} \frac{\mathrm{mob}(\alpha)}{|\alpha|}$ [Grabisch, 2006, Owen, 2014]: fast Möbius inverse needed
Fast multiplication algorithms [Yates, 1937, Good, 1958]: iterated Kronecker products
$A \otimes^d A = (A \otimes^{d-1} A) \otimes (A \otimes^{d-1} A)$ with $A \otimes^0 A = A$.

Theorem (Post-publication result, not in [Plischke et al., 2021])

Let $v = (\mathrm{val}(\emptyset), \mathrm{val}(\{1\}), \mathrm{val}(\{2\}), \mathrm{val}(\{1,2\}), \mathrm{val}(\{3\}), \dots, \mathrm{val}(\{1,\dots,d\}))^T$ be a 2^d vector in natural order (binary coded). Möbius inverse is obtained by left-multiplication with the iterated Kronecker product of $A = \left(\begin{smallmatrix} 1 & 0 \\ -1 & 1 \end{smallmatrix}\right)$.

$A^{-1} = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ codes inclusion in 2^1: $\emptyset \subset \emptyset$, $\emptyset \subset \alpha$, $\alpha \subset \alpha$.

Shapley Values

Shapley Effects for Sensitivity Analyis

Example

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
0	31	44	75	0	56	44	100	val α
								Step 1
								Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
								Step 1
								Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44	0	44)					Step 1
								Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44	0	44)	(31	31	56	56)	Step 1
								Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
								Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0	31	56)					Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0	31	56)	(44	44	0	0)	Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0)	(31	56)	(44	44)	(0	0)	Step 2
								mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots\mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0)	(31	56)	(44	44)	(0	0)	Step 2
(0	31	44	0)					mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0)	(31	56)	(44	44)	(0	0)	Step 2
(0	31	44	0)	(0	25	0	0)	mob(α)
								$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0)	(31	56)	(44	44)	(0	0)	Step 2
0	31	44	0	0	25	0	0	mob(α)
	31	44	0	0	12.5	0	0	$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$								Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0)	(31	56)	(44	44)	(0	0)	Step 2
0	31	44	0	0	25	0	0	mob(α)
	31	44	0	0	12.5	0	0	$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$	31		0		12.5		0	Sh_1
$\sum$								Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0)	(31	56)	(44	44)	(0	0)	Step 2
0	31	44	0	0	25	0	0	mob(α)
	31	44	0	0	12.5	0	0	$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$	31		0		12.5		0	Sh_1
$\sum$		44	0			0	0	Sh_2
$\sum$								Sh_3

Illustrating Fast Möbius/Yates Transformation

(first,last) $\cdots \mapsto$ (all first, all last-first)

$\emptyset$	1	2	1,2	3	1,3	2,3	1,2,3	α
(0	31)	(44	75)	(0	56)	(44	100)	val α
(0	44)	(0	44)	(31	31)	(56	56)	Step 1
(0	0)	(31	56)	(44	44)	(0	0)	Step 2
0	31	44	0	0	25	0	0	mob(α)
	31	44	0	0	12.5	0	0	$\lvert\alpha\rvert^{-1}$ mob(α)
$\sum$	31		0		12.5		0	Sh_1
$\sum$		44	0			0	0	Sh_2
$\sum$				0	12.5	0	0	Sh_3

Conclusions

- Shapley values: Computation can be improved by using alternative representations
- Deterministic calculation for $d \leq 15$ are in computational reach, provided model simulation is at virtual no cost.
- For variance based value functions, additional computational advances are available
- Shapley-Owen effects for groups: only available via Möbius inverses formula (not discussed)

Open issues

- Currently, only Gaussian copula dependence structures (we are working on this)
- Alternative value functions: finite change, elementary effects
- Goda approach: Impact of Quasi Random Permutations

Thank You!

Questions, Comments

`mailto:elmar.plischke@tu-clausthal.de`

Preprints, Scripts, Stuff

`https://artefakte.rz-housing.tu-clausthal.de/epl/`

References I

Goda, T. (2021).
A simple algorithm for global sensitivity analysis with Shapley effects.
Reliability Engineering&System Safety, 213:107702.

Good, I. J. (1958).
The interaction algorithms and practical Fourier analysis.
Journal of the Royal Statistical Society, Series B, 20:361–372.
Addendum: 22:372-375, 1960.

Grabisch, M. (2006).
Capacities and games on lattices: A survey of results.
International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 14(4):371–392.

Myerson, R. B. (1991).
Game Theory: Analysis of Conflict.
Harvard University Press.

Owen, A. B. (2014).
Sobol' indices and Shapley values.
SIAM/ASA Journal on Uncertainty Quantification, 2(1):245–251.

References II

Owen, A. B. and Prieur, C. (2017).
On Shapley value for measuring importance of dependent inputs.
Technical report, arXiv.
arXiv:1610.02080.

Plischke, E., Rabitti, G., and Borgonovo, E. (2021).
Computing Shapley effects for sensitivity analysis.
SIAM/ASA Journal on Uncertainty Quantification, 9(4):1411–1437.

Rota, G.-C. (1964).
On the foundations of combinatorial theory I. Theory of Möbius functions.
Z. Wahrscheinlichkeitstheorie, 2:340–368.

Shapley, L. S. (1953).
A value for n-person games.
In Kuhn, H. W. and Tucker, A. W., editors, *Contributions to the Theory of Games. Volume II*, volume 28 of *Annals of Mathematics Studies*, pages 307–317. Princeton University Press, Princeton, NJ.

Song, E., Nelson, B. L., and Staum, J. (2016).
Shapley effects for global sensitivity analysis: Theory and computation.
SIAM/ASA Journal on Uncertainty Quantification, 4(1):1060–1083.

References III

Yates, F. (1937).
The design and analysis of factorial experiments.
Technical Communication 35, Imperial Bureau of Soil Science, Harpenden.