Pairwise Interaction Network

Mario V. Wüthrich
Department of Mathematics
ETH zürich

Joint work with: Michael Mayer (Mobiliar Insurance), Ronald Richman (insureAI), Salvatore Scognamiglio (University of Naples Parthenope)

Actuarial Data Science Après-Midi – SAV

October 15, 2025

AI Tools for Actuaries (work in progress)

Authors: Mario V. Wüthrich, Ronald Richman, Benjamin Avanzi, Mathias Lindholm, Marco Maggi, Michael Mayer, Jürg Schelldorfer, Salvatore Scognamiglio

About This Project

This project aims to empower the actuarial profession with modern machine learning and AI tools. We provide comprehensive teaching materials that consist of lecture notes (technical document) building the theoretical foundation of this initiative. Each chapter of these lecture notes is supported by notebooks and slides which give teaching material, practical guidance and applied examples. Moreover, hands-on exercises in both R and Python are provided in additional notebooks.

Lecture Notes (Technical Document)

Lecture Notes

Notebooks, Slides and Code

Chapter 1: Introduction and Preliminaries

Notebook PDF Slides

https://aitools4actuaries.com/

- **Pairwise Interaction Network**

Actuarial pricing – regression modeling

- **Actuarial pricing.** Find the (unknown) regression function $\boldsymbol{X} \mapsto \mu(\boldsymbol{X})$ that describes the conditionally expected claim

$$\mu(\boldsymbol{X}) = \mathbb{E}\left[Y \middle| \boldsymbol{X}\right],$$

  where $\boldsymbol{X}$ are the covariates (features) characterizing the claim (response) Y.

- **Practical solution.** Select a class $\mathcal{M} = \{\mu\}$ of candidate regression models, a strictly consistent loss function L for mean estimation, and solve for a given i.i.d. learning sample $(Y_i, \boldsymbol{X}_i)_{i=1}^n$

$$\widehat{\mu} \in \operatorname*{arg\,min}_{\mu \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} L(Y_i, \mu(\boldsymbol{X}_i)).$$

- **Commonly used model classes.** Generalized linear models (GLMs), gradient boosting machines (GBMs) and feed-forward neural networks (FNNs).

Feed-forward neural networks

- **FNN model class.** One usually fixes the FNN architecture (hyper-parameters):
  - ⋆ depth of network;
  - ⋆ number of units in the hidden layers;
  - ⋆ activation functions;
  - ⋆ output activation (link function);
  - ⋆ further modeling features like normalization layers, drop-out layers, etc.

- Based on these hyper-parameter choices, one receives a parametrized model class

$$\mathcal{M} = \{\boldsymbol{z}_\vartheta\}_\vartheta, \qquad \text{of FNNs } \boldsymbol{z}_\vartheta \text{ with parameters (weights) } \vartheta.$$

- **Model training** focuses on finding an optimal parameter ϑ.

- Actuarial regression problems are predominantly based on **tabular input data** $\boldsymbol{X}$.

- Plain-vanilla FNN training might struggle on tabular input data, especially (but not only) if there are many high-cardinality categorical covariates.

Actuarial regression problems

- Many actuarial problems are characterized by a low signal-to-noise ratio.

- The main issue lies in the model training, struggling to discriminate fine-grained systematic structure from pure randomness on (small) finite samples.

- Attempts to adapt network architectures better to tabular input data:

  ⋆ TabTransformer of Huang et al. (2020);
  ⋆ Feature tokenizer (FT) transformer of Gorishniy et al. (2021) and Brauer (2024);
  ⋆ Credibility transformer of Richman et al. (2025);
  ⋆ Piecewise linear encoding (PLE) of Gorishniy et al. (2022);
  ⋆ In-context learning (ICL) credibility transformer of Padayachy et al. (2025).

  Many of these methods extract information via a classify (CLS) token, introduced by Devlin et al. (2019) to pre-train BERT (bidirectional encoder representations from transformers) in the context of language processing.

- We take one step back here: inspired by regression trees and GBMs, we only allow for **binary interactions** among the inputs by correspondingly partitioning the covariate space – this will still bear some similarity to transformers.

Tree-like pairwise interaction network

- We introduce tree-like pairwise interaction networks (PINs).
- It mimics a continuous version of covariate space splitting.
- These splits consider two covariate components at a time, thus, allowing to capture binary interactions; classical regression trees are based on splitting only along one covariate at a time, which does not directly capture interactions.
- PIN is closely related to generalized additive models with binary interactions (GA^{2}Ms); see Lou et al. (2013) and Wood (2006).
- Pros:
  - ⋆ Excellent predictive performance on tabular actuarial data.
  - ⋆ Efficient computations of SHapley Additive exPlanations (SHAP).
  - ⋆ Can be interpreted as a graph neural network (GNN) allowing to benefit from the GNN toolbox.
  - ⋆ Allows for variable selection (implemented but unpublished...).
- Cons:
  - ⋆ Scales badly in the input dimension.
  - ⋆ Higher order interactions are not captured and boosting is not straightforward.

Feature tokenizer

- It has become common practice to map tabular input data to 2D tensors; Gorishniy et al. (2021), Brauer (2024) and Richman et al. (2025).

- Map the tabular input data $\boldsymbol{X} = (X_1, \ldots, X_q)$ to a 2D input tensor

$$\boldsymbol{X} \;\mapsto\; \boldsymbol{\phi} = \Big[\phi_1 = \phi_1(X_1),\, \ldots,\, \phi_q = \phi_q(X_q)\Big] \;\in\; \mathbb{R}^{d\times q}.$$

- The embedding dimension $d \in \mathbb{N}$ is a hyper-parameter selected by the modeler.

- **Categorical covariates:** *entity embedding.* For $X_j \in \mathcal{X}_j := \{1, \ldots, n_j\}$ with n_j levels, consider

$$\phi_j : \mathcal{X}_j \to \mathbb{R}^d, \qquad X_j \;\mapsto\; \phi_j(X_j) = \sum_{x=1}^{n_j} \boldsymbol{w}_{j,x}\, \mathbb{1}_{\{X_j = x\}},$$

  with embedding matrix $[\boldsymbol{w}_{j,1}, \ldots, \boldsymbol{w}_{j,n_j}] \in \mathbb{R}^{d\times n_j}$ – these parameters are learned during network training (proximity means similarity in risk behavior).

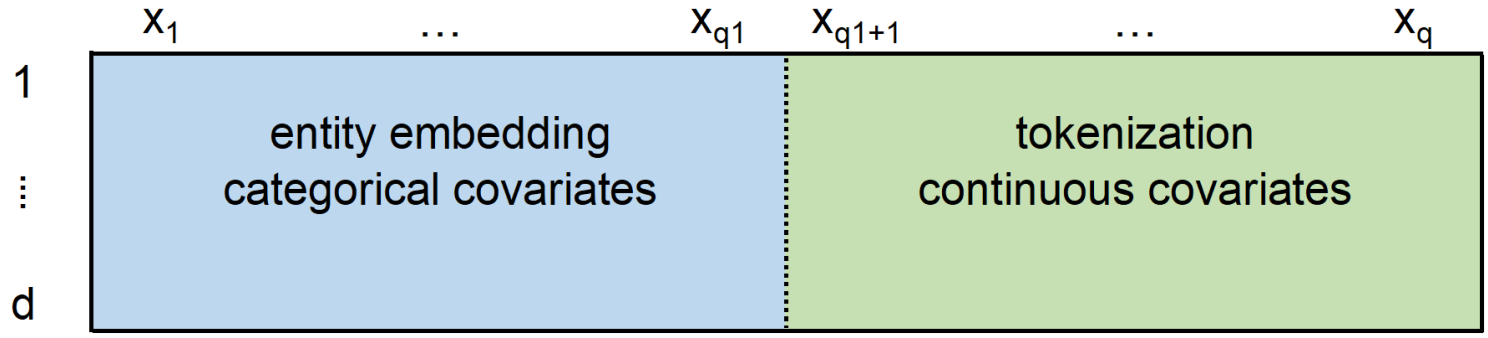
- **Continuous covariates:** *FNN embedding.* For $X_j \in \mathbb{R}$, consider a FNN

$$\phi_j^{\mathrm{FNN1}} : \mathbb{R} \to \mathbb{R}^d, \qquad X_j \;\mapsto\; \phi_j(X_j) = \phi_j^{\mathrm{FNN1}}(X_j),$$

  with ϕ_j^{FNN1} a standard FNN – its parameters are learned during network training (every continuous covariate X_j has its own FNN ϕ_j^{FNN1}).

- This results in the 2D input tensor

$$\boldsymbol{X} \;\mapsto\; \boldsymbol{\phi} = [\phi_1, \ldots, \phi_q] = [\phi_1(X_1), \ldots, \phi_q(X_q)] \in \mathbb{R}^{d \times q}.$$



- Covariates do not interact yet, this is similar to tranformers and attention layers.

Pairwise interaction token

- Select for the **shared interaction network** a deep FNN $\boldsymbol{z}_\theta^{\mathrm{FNN2}}$ providing

$$(\phi_j, \phi_k, \boldsymbol{e}_{j,k}) \;\mapsto\; \boldsymbol{z}_\theta^{\mathrm{FNN2}}(\phi_j, \phi_k, \boldsymbol{e}_{j,k}) \;\in\; \mathbb{R},$$

  with:

  - ⋆ this network $\boldsymbol{z}_\theta^{\mathrm{FNN2}}$ models the pairwise interaction between **all** pairs (ϕ_j, ϕ_k);
  - ⋆ the network parameter θ is **shared** across **all** pairs;
  - ⋆ to allow for different behaviors in the interactions between the pairs, we add a learnable **interaction token** $\boldsymbol{e}_{j,k} \in \mathbb{R}^{d_0}$;
  - ⋆ unlike the CLS token of Devlin et al. (2019) in BERT, our interaction token is not used to extract information, but it is used to encode different interaction behavior.

- This shared interaction network shares similarity with the **(self-)attention layer** of Vaswani et al. (2017). However, instead of computing attention weights that are applied to so-called values, we let the covariates directly form the outputs, and the interaction token may pronounce the effect of a given covariate pair.

Pairwise interaction layer

- Define the **pairwise interaction units** by

$$h_{j,k}(\boldsymbol{X}) = \sigma_{\text{hard}}\Big(\boldsymbol{z}_\theta^{\text{FNN2}}\left(\phi_j(X_j), \phi_k(X_k), \boldsymbol{e}_{j,k}\right)\Big), \qquad \text{for } 1 \le j \le k \le q,$$

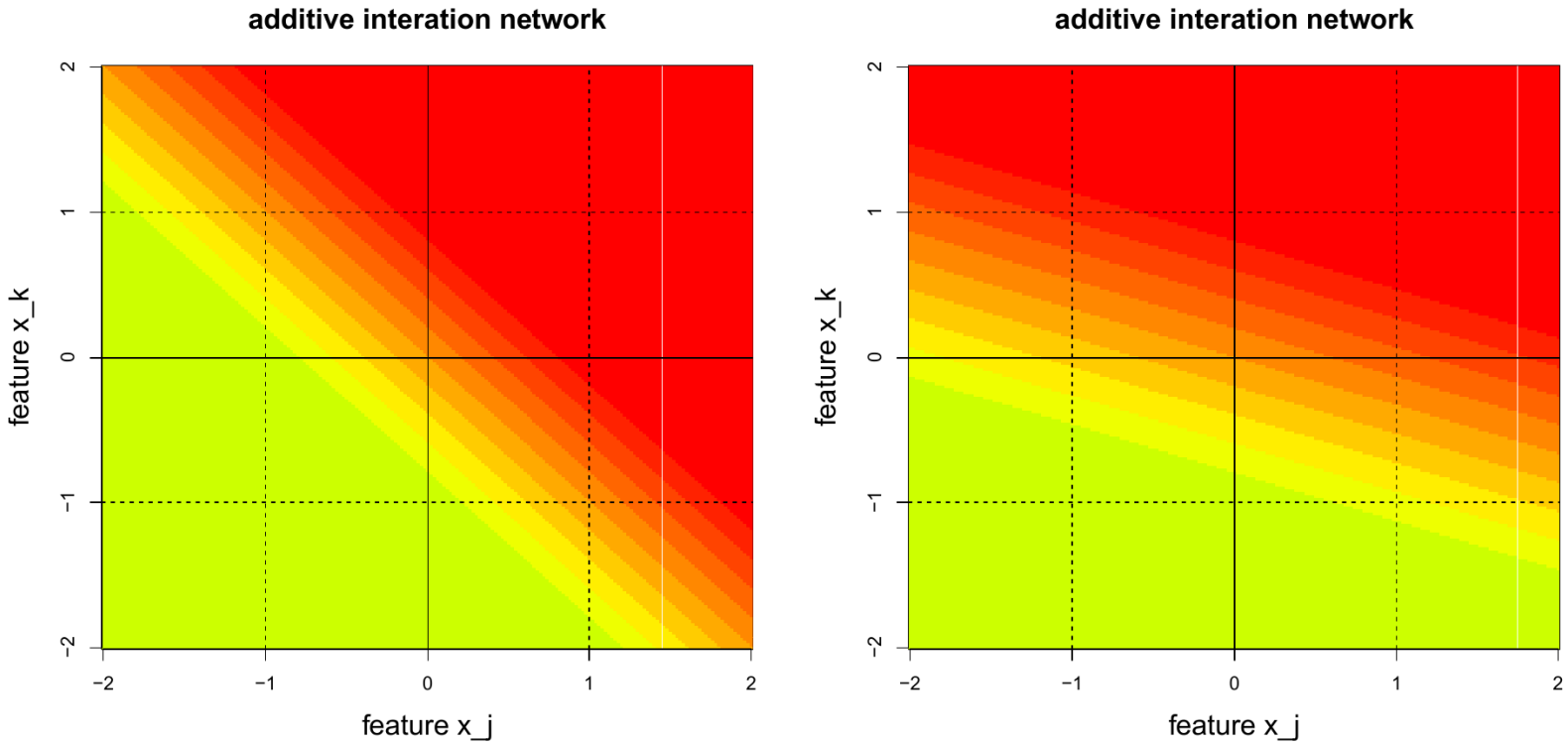
  for hard sigmoid activation $\sigma_{\text{hard}}(x) = \max(0,\, \min(1,\, (1+x)/2)) \in [0,1]$.

- The **pairwise interaction layer** for input $\boldsymbol{X} \in \mathcal{X}$ is defined by the upper-right triangular 2D tensor

$$(h_{j,k}(\boldsymbol{X}))_{1\le j\le k\le q} = \begin{pmatrix} h_{1,1}(\boldsymbol{X}) & h_{1,2}(\boldsymbol{X}) & \cdots & h_{1,q}(\boldsymbol{X}) \\ \text{n/a} & h_{2,2}(\boldsymbol{X}) & \cdots & h_{2,q}(\boldsymbol{X}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{n/a} & \text{n/a} & \cdots & h_{q,q}(\boldsymbol{X}) \end{pmatrix},$$

  where the lower-left part is left undefined (indicated by n/a).

Example: pairwise interaction layer



- The plots give two examples of (linear) pairwise interaction units $h_{j,k}(\boldsymbol{X})$.
- Essentially, we receive a continuous and generalized version of classical regression tree splits which can only be vertical or horizontal.

Pairwise interaction network

- The **pairwise interaction network** (PIN) is defined by $\mu_{\mathrm{PIN}} : \mathcal{X} \to \mathbb{R}$

$$\boldsymbol{X} \;\mapsto\; \mu_{\mathrm{PIN}}(\boldsymbol{X}) = g\left(\sum_{1 \le j \le k \le q} w_{j,k}\, h_{j,k}(\boldsymbol{X}) + b\right),$$

  with learnable output weights $(w_{j,k})_{1 \le j \le k \le q}$, bias $b \in \mathbb{R}$ and output activation (inverse link) function g.

- Structural similarities with GA^{2}Ms and GBMs are obvious.

- This PIN can be interpreted as GNN with message function

$$\mathcal{M}(\phi_j, \phi_k, \boldsymbol{e}_{j,k}) = \sigma_{\mathrm{hard}}\left(\boldsymbol{z}_\theta^{\mathrm{FNN2}}(\phi_j, \phi_k, \boldsymbol{e}_{j,k})\right),$$

  with nodes (vertices) $(\phi_j)_{j \in V}$ and edges $(\boldsymbol{e}_{j,k})_{(j,k) \in E}$.

Example: French MTPL claims frequency data

- A standard actuarial benchmark dataset is the French motor third party liability (MTPL) claims frequency data of Dutang et al. (2018).

- We use the identical data cleaning and learning-test sample partition as in W.–Merz (2023).[1] This makes the results comparable to various studies in the actuarial literature.

Characteristic	**Learning set**	**Test set**
Number of insurance policies	610'206	67'801
Total exposure (years)	322'857	35'943
Number of claims	23'738	2'645
Average frequency (per exposure)	7.36%	7.35%
Covariate description		
Categorical (3): VehBrand, Region, VehGas (binary)		
Continuous (6): Area, VehPower, VehAge, DrivAge, BonusMalus, Density		
Response: ClaimNb (claim count)		
Exposure: Exposure (in yearly units)		

[1] https://aitools4actuaries.com/

PIN architecture and model training

- We select the following PIN architecture:

Module	# **Weights**
• **Embedding dimension** $d = 10$**:**	
Categorical features (2) with $n_j = 11, 22$	330
Continuous features (7) FNN depth 2 with 20 units	1'750
• **Pairwise interaction layer (FNN depth** 3**):**	
Interaction tokens $\boldsymbol{e}_{j,k}$ $(9 \cdot 10/2)$ with $d_0 = 10$	450
1st FNN layer with 30 units	930
2nd FNN layer with 20 units	620
3rd layer with hard sigmoid activation	21
• **Output:**	
Output weights $(w_{j,k})_{j \leq k}$ including bias b	46
Total	4'147

- Model training is done with stochastic gradient descent (SGD) using the Poisson deviance loss, the Adam optimizer, a batch size of 128, and early stopping.

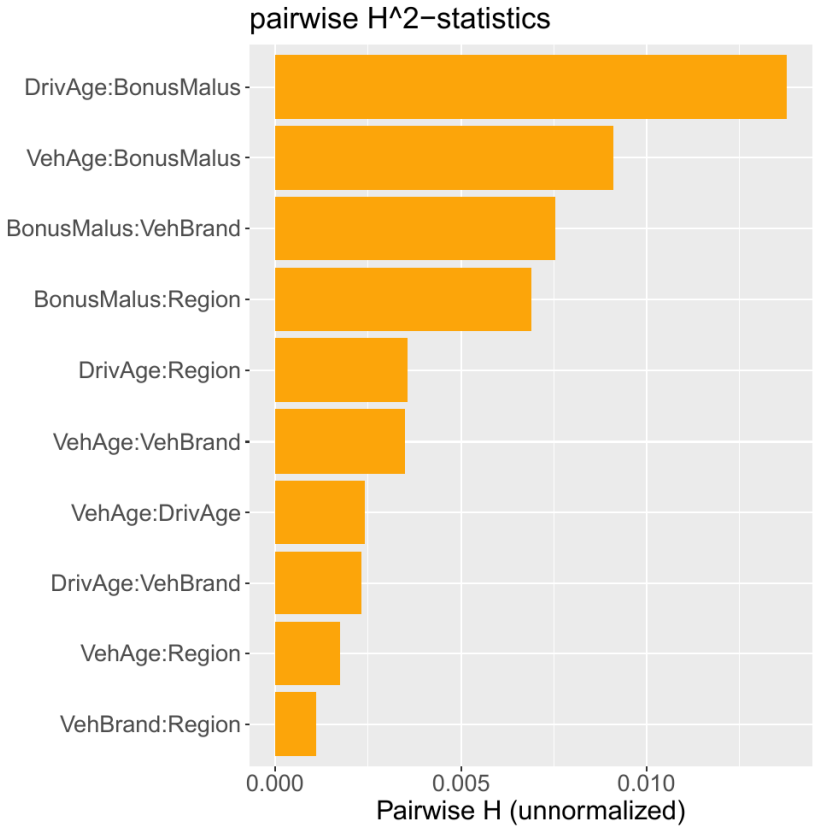
PIN results

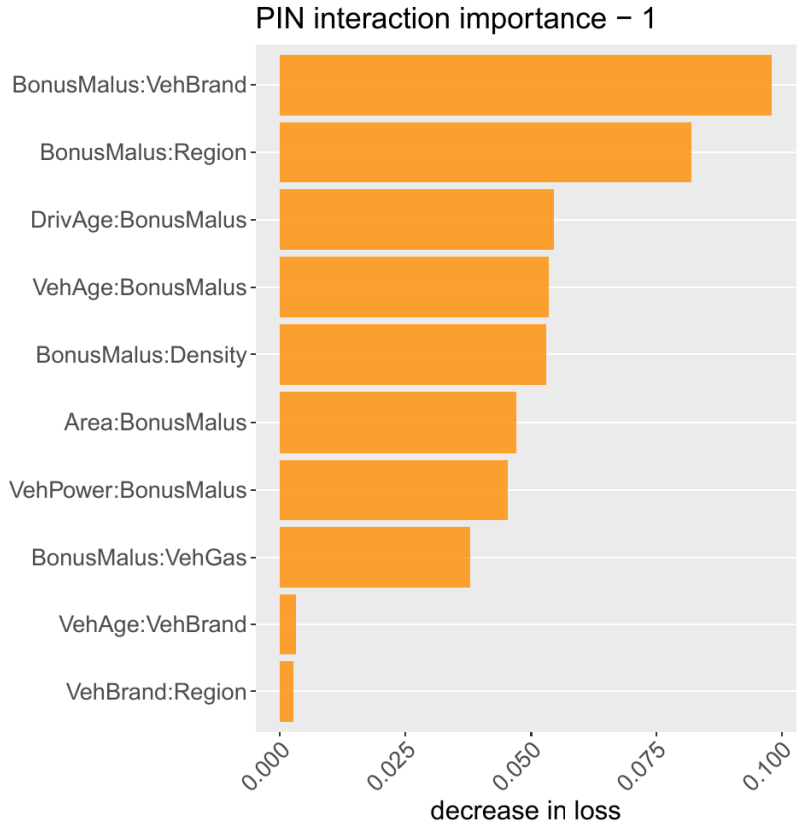
Model	# **Param.**	**Out-of-sample Poisson loss**	
Null model (intercept-only)	1	25.445	
Poisson GLM	50	24.102	
Poisson GAM	(66.7)	23.956	
Plain-vanilla FNN; W.–Merz (2023)	792	23.783	
FT transformer; Brauer (2024)	27,133	23.726	($\pm$ 0.006)
Credibility transformer; Richman et al. (2025)	1,746	23.711	
In-context learning (ICL); Padayachy et al. (2025)	46,439	23.676	
Pairwise interaction network (PIN)	4,147	23.667	($\pm$ 0.020)

- The network results are ensembles over 10 SGD fits.
- The out-of-sample Poisson deviance loss is dominated by irreducible risk (low signal-to-noise ratio), i.e., the magnitude of model improvement (systematic structure) lives on a smaller scale.
- Out-of-sample fluctuations (1sd) of SGD training are approximately: ± 0.020.

- **Section 2: Explainability**

Pairwise interaction strength





- (lhs): Friedman–Popescu's (2008) H^2-statistics (unnormalized version).
- (rhs): PIN interaction importance; this is similar to an ANOVA for GLMs by parallel training of including individual interaction terms.

SHapley Additive exPlanations (SHAP)

- SHAP introduced by Lundberg–Lee (2017) uses the Shapley (1953) values from cooperative game theory to additively decompose (explain) a prediction $\mu_{\text{PIN}}(\boldsymbol{X})$.

- Under Shapley's fairness axioms, there is exactly one decomposition $(\psi_j)_{j=1}^q$ of a gain of a cooperative game among its q players (under a given value function ν)

$$\psi_j = \frac{1}{q} \sum_{\mathcal{C} \subseteq \mathcal{Q} \setminus \{j\}} \binom{q-1}{|\mathcal{C}|}^{-1} \Big(\nu(\mathcal{C} \cup \{j\}) - \nu(\mathcal{C}) \Big), \qquad \text{for } j \in \mathcal{Q},$$

  where $\mathcal{C} \subseteq \mathcal{Q}$ are the possible coalitions of the grand coalition $\mathcal{Q} = \{1, \ldots, q\}$.

- Adopting this to machine learning explanations, there are two difficulties:

  1. Explicit computation of the Shapley values $(\psi_j)_{j=1}^q$ for large q (combinatorial complexity).
  2. Choice of value function ν; how should one mask the coalitions in predictions?

KernelSHAP and PermutationSHAP

- There are two equivalent alternative representations of the Shapley values:
  - ⋆ KernelSHAP version of Lundberg–Lee (2017);
  - ⋆ PermutationSHAP version of Štrumbelj–Kononenko (2010, 2014).

- **PermutationSHAP.** Denote by $\pi = (\pi_1, \ldots, \pi_q)$ a permutation of the ordered set $(1, \ldots, q)$. Let $\kappa(j) \in \mathcal{Q}$ be the index with $\pi_{\kappa(j)} = j$, and set

$$\mathcal{C}_{\pi,j} = \left\{\pi_1, \ldots, \pi_{\kappa(j)-1}\right\} \subset \mathcal{Q}.$$

  The Shapley values can equivalently be computed by

$$\psi_j = \frac{1}{q!} \sum_{\pi} \nu\left(\mathcal{C}_{\pi,j} \cup \{j\}\right) - \nu\left(\mathcal{C}_{\pi,j}\right).$$

- KernelSHAP and PermutationSHAP are approximated by Monte Carlo.

- Mayer–W. (2025) prove asymptotic normality results for both of these Monte Carlo approximations – this is useful to quantify the accuracy under finite samples.

Interventional SHAP

- We consider the *interventional* value function

$$\mathcal{C} \;\mapsto\; \nu_{\boldsymbol{x}}(\mathcal{C}) := \mathbb{E}_{\boldsymbol{X}}\left[\sum_{1\leq j\leq k\leq q} w_{j,k}\, h_{j,k}\left(\boldsymbol{x}_{\mathcal{C}}, \boldsymbol{X}_{\mathcal{Q}\setminus\mathcal{C}}\right) + b\right],$$

  in the fixed covariate value $\boldsymbol{x}$.

- **Remarks:**
  - ⋆ Usually, the interventional value function $\nu_{\boldsymbol{x}}$ considers the (PIN) predictions on the link scale
$$g^{-1}\left(\mu_{\text{PIN}}\left(\boldsymbol{x}_{\mathcal{C}}, \boldsymbol{X}_{\mathcal{Q}\setminus\mathcal{C}}\right)\right).$$
  - ⋆ This value function $\nu_{\boldsymbol{x}}(\mathcal{C})$ is approximated empirically using a background dataset.
  - ⋆ The *conditional* SHAP version is more complicated (and time-consuming) because it requires to compute proper conditional expectations (empirically).

SHAP for binary interactions

- We consider the *interventional* value function

$$\mathcal{C} \;\mapsto\; \nu_{\boldsymbol{x}}(\mathcal{C}) := \mathbb{E}_{\boldsymbol{X}}\left[\sum_{1\le j\le k\le q} w_{j,k}\, h_{j,k}\left(\boldsymbol{x}_{\mathcal{C}}, \boldsymbol{X}_{\mathcal{Q}\setminus\mathcal{C}}\right) + b\right],$$

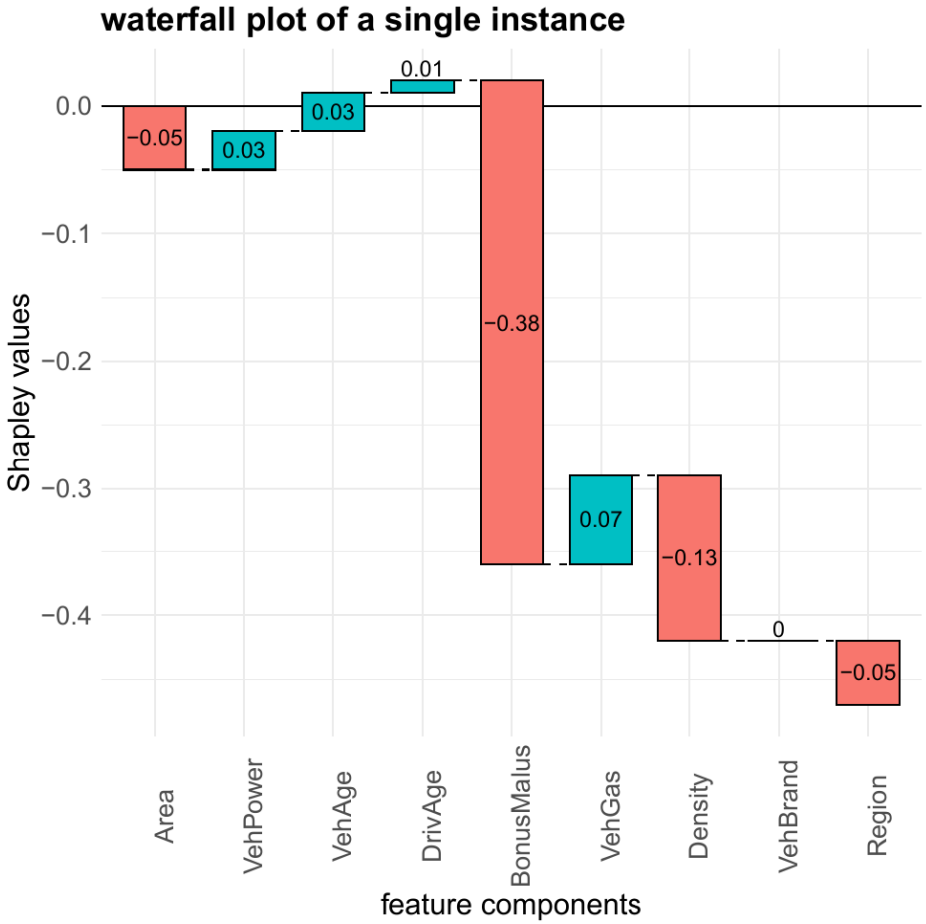
  in the fixed covariate value $\boldsymbol{x}$.

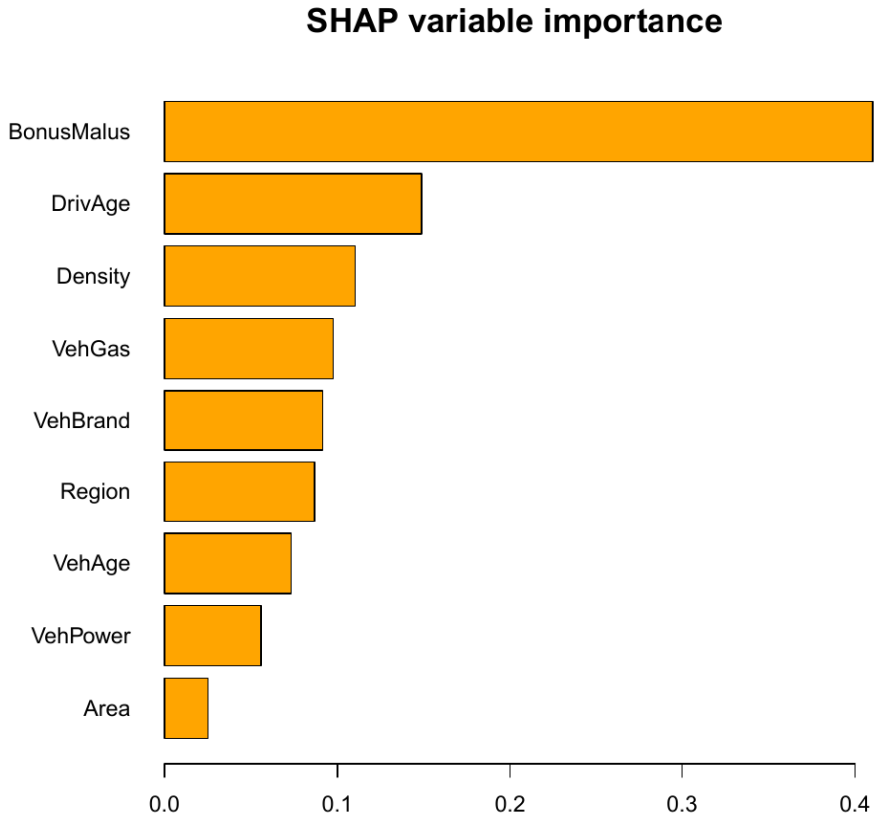
- **Important:** $\nu_{\boldsymbol{x}}(\mathcal{C})$ only contains *binary interactions* through $h_{j,k}$.

- **Proposition** [Lundberg (2018), Mayer–W. (2025)]**.** In case of a value function ν that contains at most binary interactions, it suffices to compute the Monte Carlo PermutationSHAP for one single permutation $\pi = (\pi_1, \ldots, \pi_q)$ and its reversed pair $(\pi_q, \ldots, \pi_1)$ to receive the exact Shapley values $(\psi_j)_{j=1}^q$. Any permutation π and its reversed pair does the job.

- This allows for very efficient SHAP computations in the binary interaction case.

SHAP illustrations





- There are many more post-hoc explainability tools based on SHAP decompositions.

Summary

- We have introduced the pairwise interaction network (PIN).
- Its implementation was motivated by the fact that classical feed-forward neural networks often struggle to deal with tabular input data.
- PIN shares many similarities with other machine learning models, e.g.,
  - ⋆ regression trees and gradient boosting machines;
  - ⋆ attention layers of transformer architectures; and
  - ⋆ graph neural networks.
- PIN has an excellent predictive performance – in fact, the best of the models compared – and it can be trained on an ordinary laptop without GPUs (for moderately large insurance datasets).
- Since PIN only contains binary interactions, it allows for an efficient computation of SHAP values, via Monte Carlo PermuationSHAP using one single permutation and its reversed pair.

References

[1] Brauer, A. (2024). Enhancing actuarial non-life pricing models via Transformers. *European Actuarial Journal* **14/3** 991-1012.

[2] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018). BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv:*1810.04805.

[3] Dutang, C., Charpentier, A., Gallic, E. (2024). Insurance dataset. *Recherche Data Gouv.* https://github.com/dutangc/CASdatasets

[4] Friedman, J.H., Popescu, B.E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics* **2/3**, 916-954.

[5] Gorishniy, Y., Rubachev, I., Babenko, A. (2022). On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems* **35**, 24991-25004.

[6] Gorishniy, Y., Rubachev, I., Khrulkov, V., Babenko, A. (2021). Revisiting deep learning models for tabular data. In: Beygelzimer, A., Dauphin, Y., Liang, P., Wortman Vaughan, J. (eds). *Advances in Neural Information Processing Systems*, **34**. Curran Associates, Inc., New York, 18932-18943.

[7] Huang, X., Khetan, A., Cvitkovic, M., Karnin, Z. (2020). TabTransformer: Tabular data modeling using contextual embeddings. *arXiv:*2012.06678.

[8] Lou, Y., Caruana, R., Gehrke, J., Hooker, G. (2013). Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 623-631.

[9] Lundberg, S.M. (2018). shap.PermutationExplainer. https://shap.readthedocs.io/en/latest/generated/shap.PermutationExplainer.html

[10] Lundberg, S.M., Lee, S.-I. (2017). A unified approach to interpreting model predictions. In: Guyon, I. Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R. Vishwanathan, S., Garnett, R. (eds.), *Advances in Neural Information Processing Systems* **30**, 4765-4774.

[11] Mayer, M., Wüthrich, M.V. (2025). Shapley values: paired-sampling algorithms. *arXiv:*2508.12947.

[12] Padayachy, K., Richman, R., Scognamiglio, S., Wüthrich, M.V. (2025). In-context learning enhanced credibility transformer. *arXiv*:2509.08122.

[13] Richman, R., Scognamiglio, S., Wüthrich, M.V. (2025). The credibility transformer. *European Actuarial Journal* **15/2**, 345-379.

[14] Richman, R., Scognamiglio, S., Wüthrich, M.V. (2025). Tree-like pairwise interaction networks. *arXiv*:2508.15678.

[15] Shapley, L.S. (1953). A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.), *Contributions to the Theory of Games*, AM-28, Volume II, Princeton University Press, 307-318.

[16] Štrumbelj, E., Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research* **11**, 1-18.

[17] Štrumbelj, E., Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* **41/3**, 647-665.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. *arXiv*:1706.03762v5.

[19] Wood, S.N. (2006). Low-rank scale-invariant tensor product smooths for generative additive mixed models. *Biometrics* **62**, 1025-1036.

[20] Wüthrich, M.V., Merz, M. (2023). *Statistical Foundations of Actuarial Learning and its Applications*. Springer Actuarial. https://link.springer.com/book/10.1007/978-3-031-12409-9

[21] Wüthrich, M.V., Richman, R., Avanzi, B., Lindholm, M. Maggi, M., Mayer, M., Schelldorfer, J., Scognamiglio, S. (2025). AI Tools for Actuaries. *SSRN Manuscript* ID 5162304. https://aitools4actuaries.com/