h e g

Haute école de gestion
Genève

Throwing Vines at the Wall: Structure Learning via Random Search

Joint work with Thomas Nagler, LMU Munich

University of Applied Sciences Western Switzerland
Thibault Vatter <tvatter@hesge.ch>

Agenda

1 Motivation

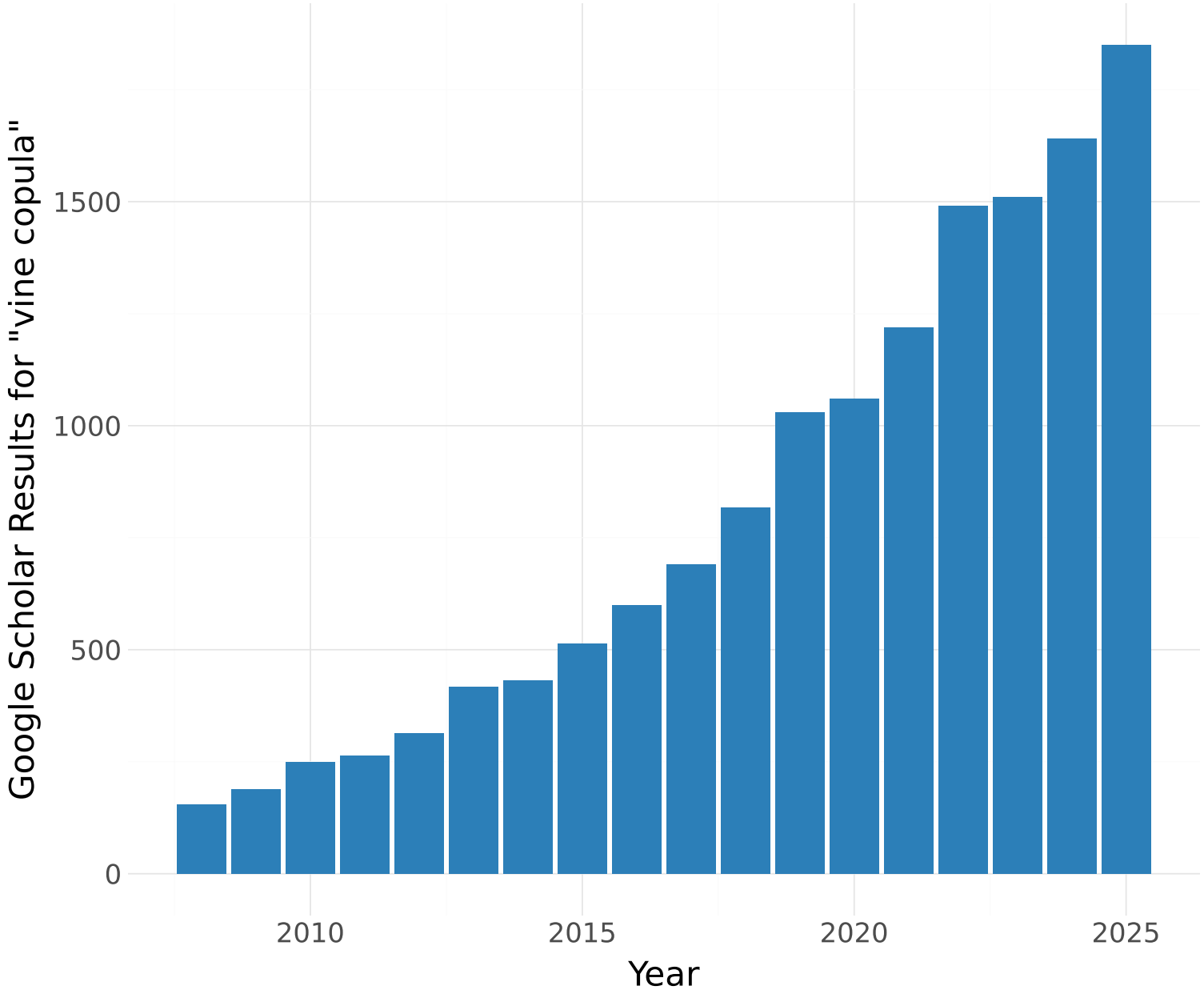
2 Methodology

3 Experiments

4 Conclusion

Agenda

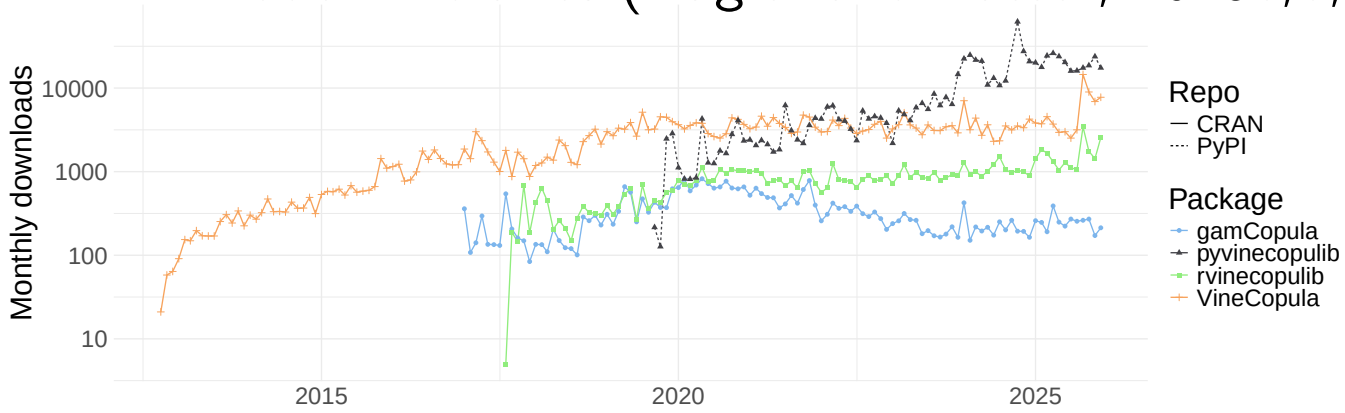
Vines Popularity



Vines Copulas 101

$$\textbf{Vine copula} = \underbrace{\textbf{Structure}}_{\text{Sequence of nested trees}} + \underbrace{\textbf{Parameters}}_{\text{Bivariate copulas on each edge}}$$

- Why are they popular ?
  - Flexible multivariate distributions (pdf, cdf, simulation, etc.) in (moderately) high dimensions.
  - **High quality open source implementations.**
    - Legacy package (Nagler et al., 2025)
    - Modern libraries (Nagler and Vatter, 2025c,b,a)



Setup

- Given
  - a structure $\mathcal{V}$
  - a dataset $\mathcal{D}$,

  Assume that $\exists$ an algorithm returning $\widehat{f}_{\mathcal{V},\mathcal{D}} \in \mathcal{F}$, with $\mathcal{F}$ the space of d-dim densities.

- **Structure learning : find $\mathcal{V}$ minimizing $\mathbb{E}[L(\widehat{f}_{\mathcal{V},\mathcal{D}}, \boldsymbol{Z}) \mid \mathcal{D}]$ for a loss $L \colon \mathcal{F} \times \mathbb{R}^d \to \mathbb{R}$.**

- How do users typically use the software ?
  - **Most of the time : default settings, i.e., a greedy heuristic to get $\widehat{\mathcal{V}}$ and return $\widehat{f}_{\widehat{\mathcal{V}},\mathcal{D}}$.**
  - Occasionally : tweak a few hyperparameters.

Can we do better ?

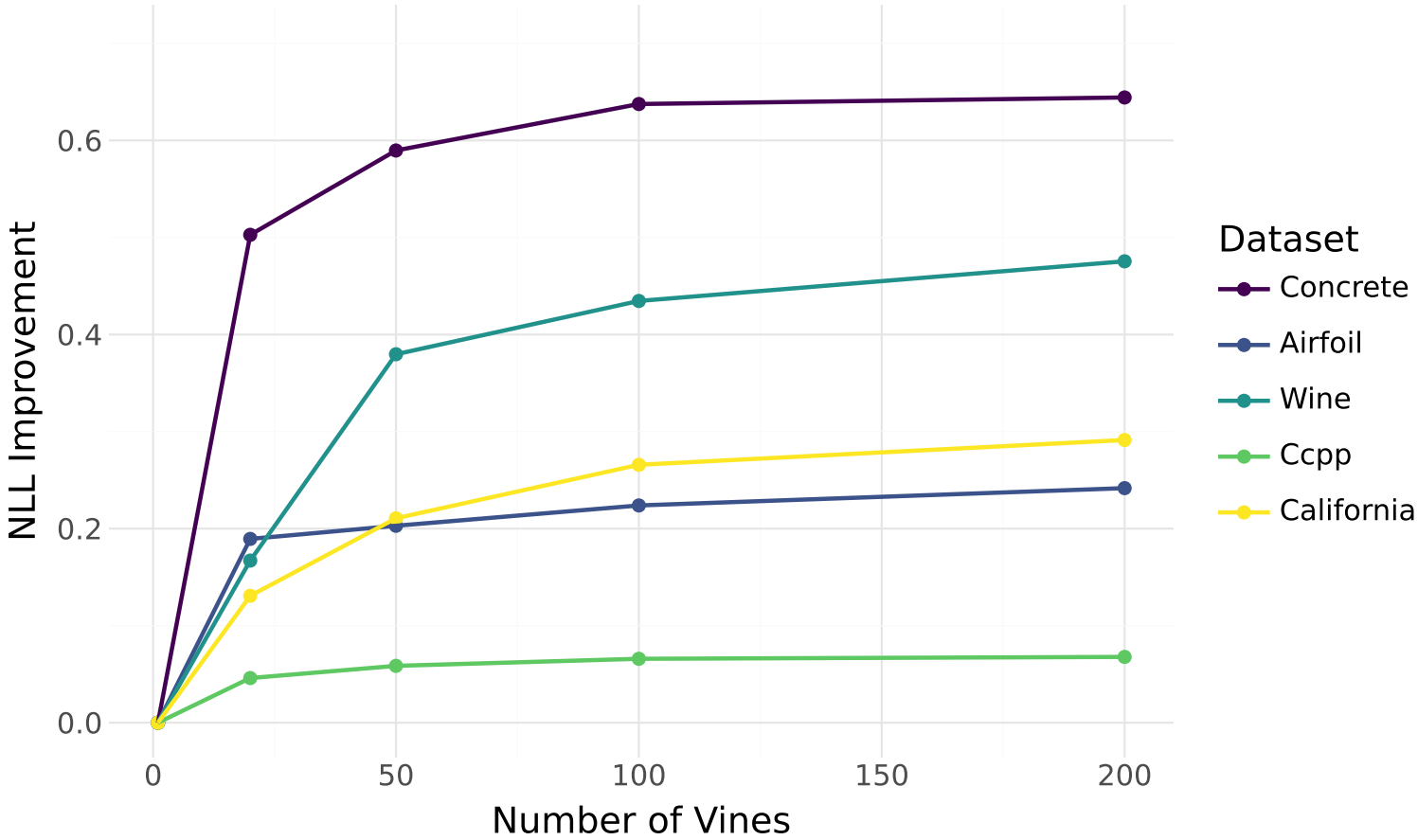
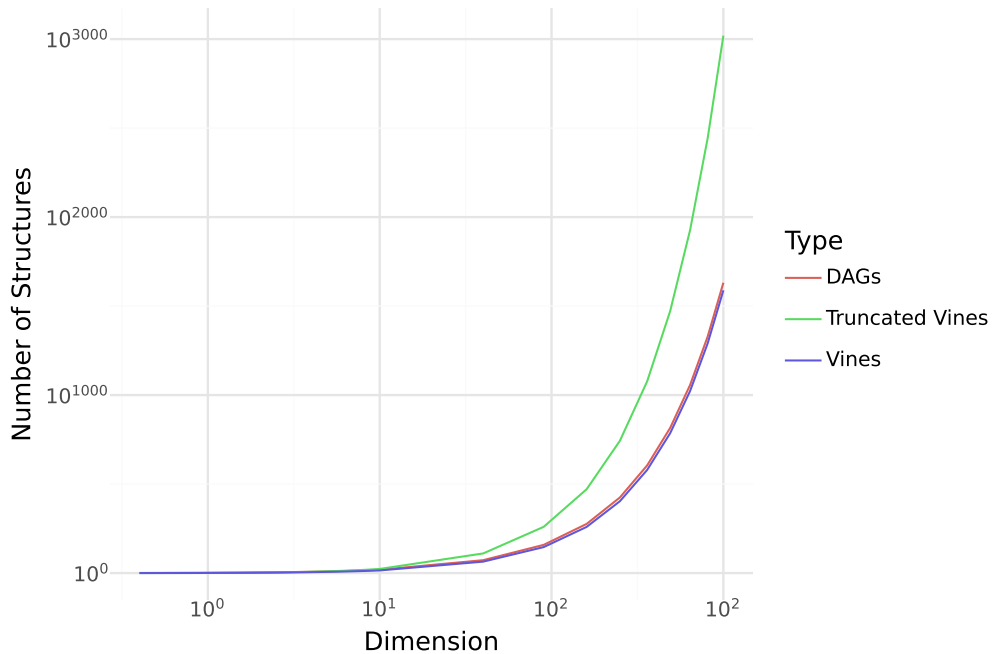


Figure – Our method's average relative improvement in % out-of-sample log-likelihood over Dissmann et al. (2013).

Structure learning : a major challenge

- $2^{(d-3)(d-2)/2-1}d!$ possible structures on d variables (Morales-Nápoles, 2011, Joe et al., 2011) : can't try them all !



- Greedy heuristics (Dissmann et al., 2013, Kraus and Czado, 2017b) : fast but often suboptimal.
- More sophisticated methods Gruber and Czado (2015), Chang et al. (2019), Sun et al. (2019) : excessively expensive even in small dimensions (say, $d \in [5, 10]$).

Agenda

Hold-out Random Search

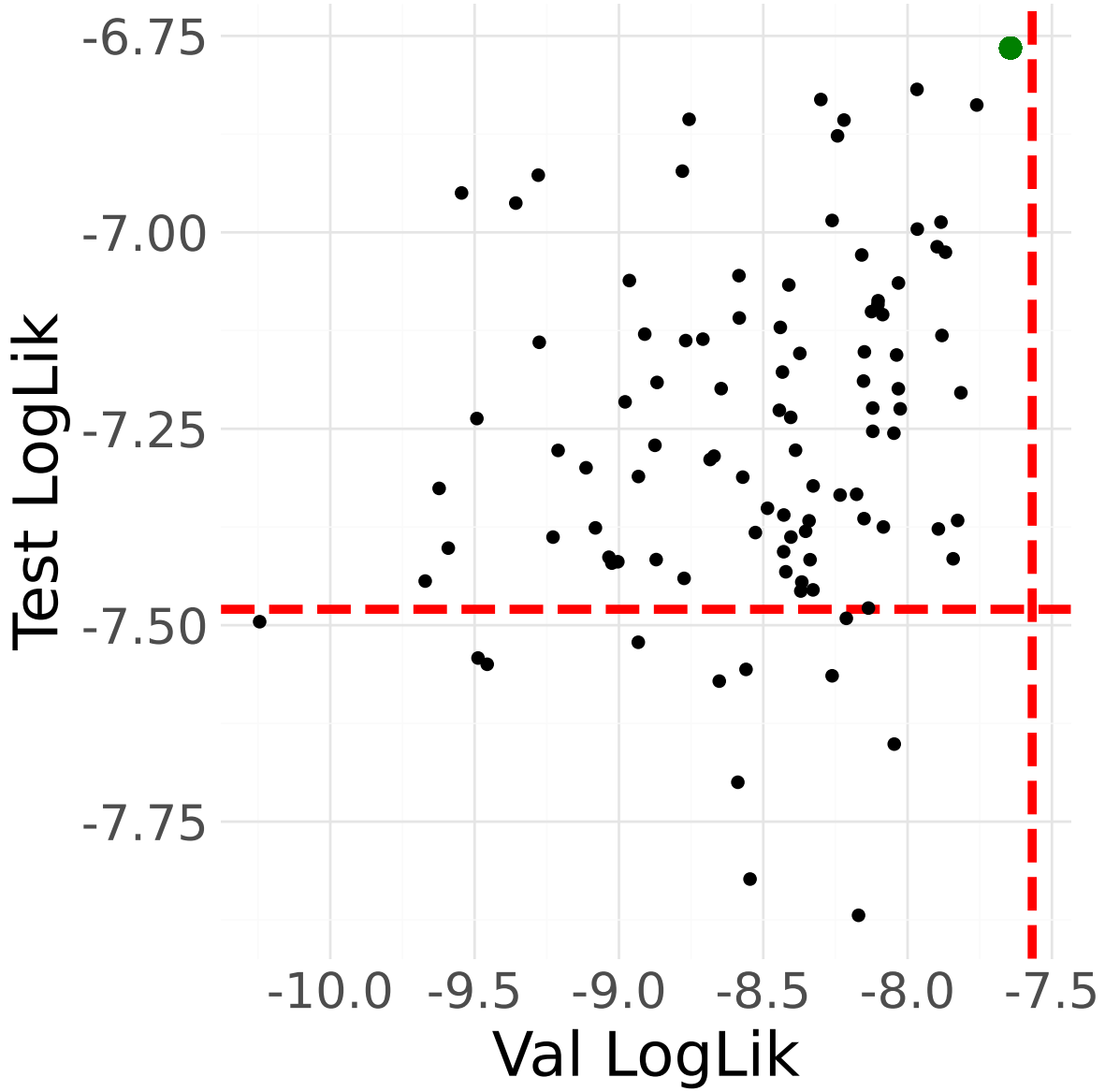
Algorithm Hold-out Random Search

Input: Dataset $\mathcal{D}$ of size n, hold-out ratio η, # of candidates M.
Output: Optimal structure $\widehat{\mathcal{V}}$.

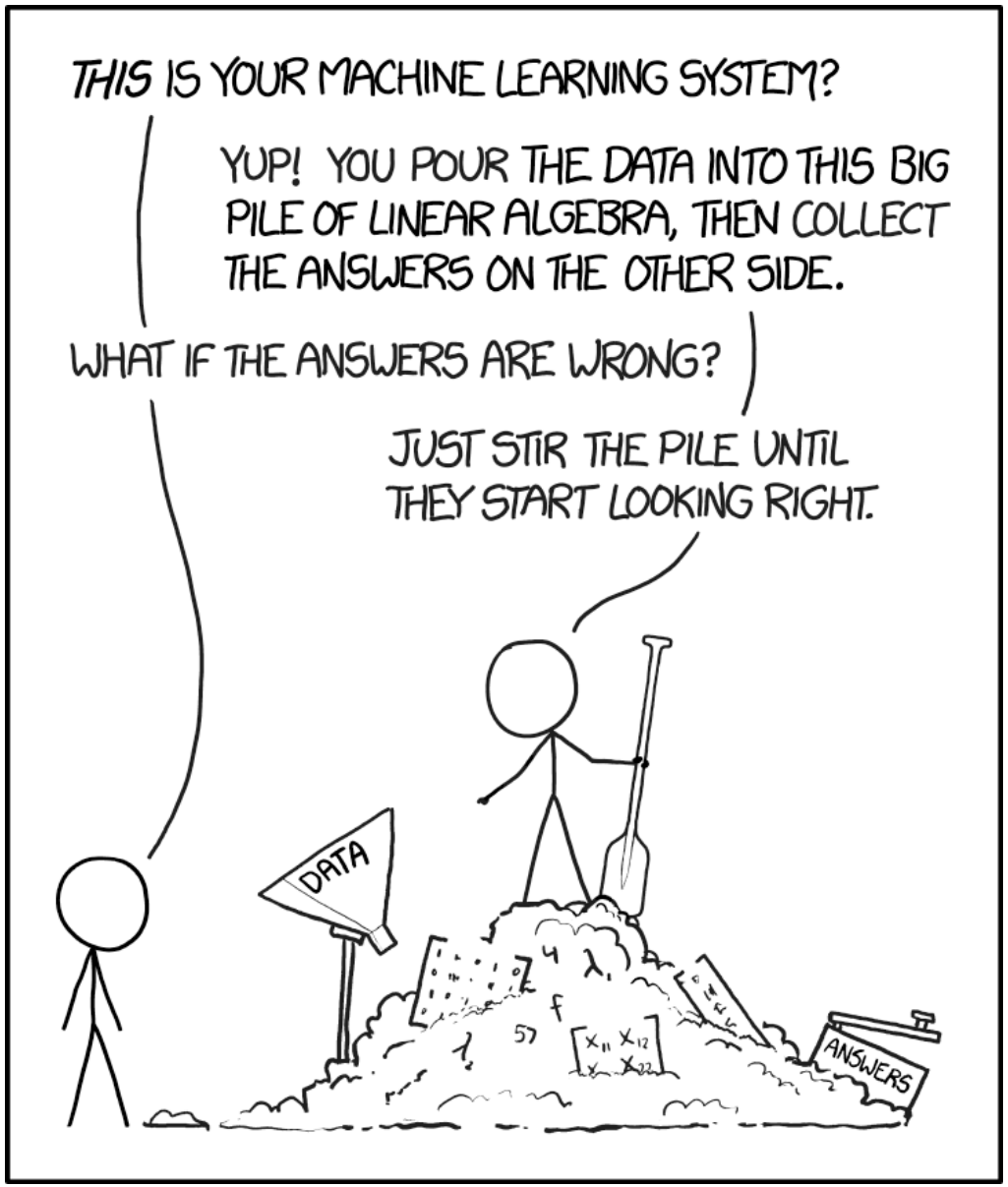
1: Split $\mathcal{D}$ into $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}} = \mathcal{D}$, $n_{\text{val}} = \lfloor \eta n \rfloor$, $n_{\text{train}} = n - n_{\text{val}}$.
2: Generate candidates $\Theta = \{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$.
3: For each $\mathcal{V} \in \Theta$:
4: Fit a density $\widehat{f}_{\mathcal{V},\mathcal{D}_{\text{train}}}$.
5: Compute $R_{\text{val}}(\mathcal{V}) = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{\boldsymbol{Z} \in \mathcal{D}_{\text{val}}} L(\widehat{f}_{\mathcal{V},\mathcal{D}_{\text{train}}}, \boldsymbol{Z})$.
6: Select : $\widehat{\mathcal{V}} = \arg\min_{\mathcal{V} \in \Theta} R_{\text{val}}(\mathcal{V})$.

- Simple, parallelizable, etc.
- How does it perform on test data ?

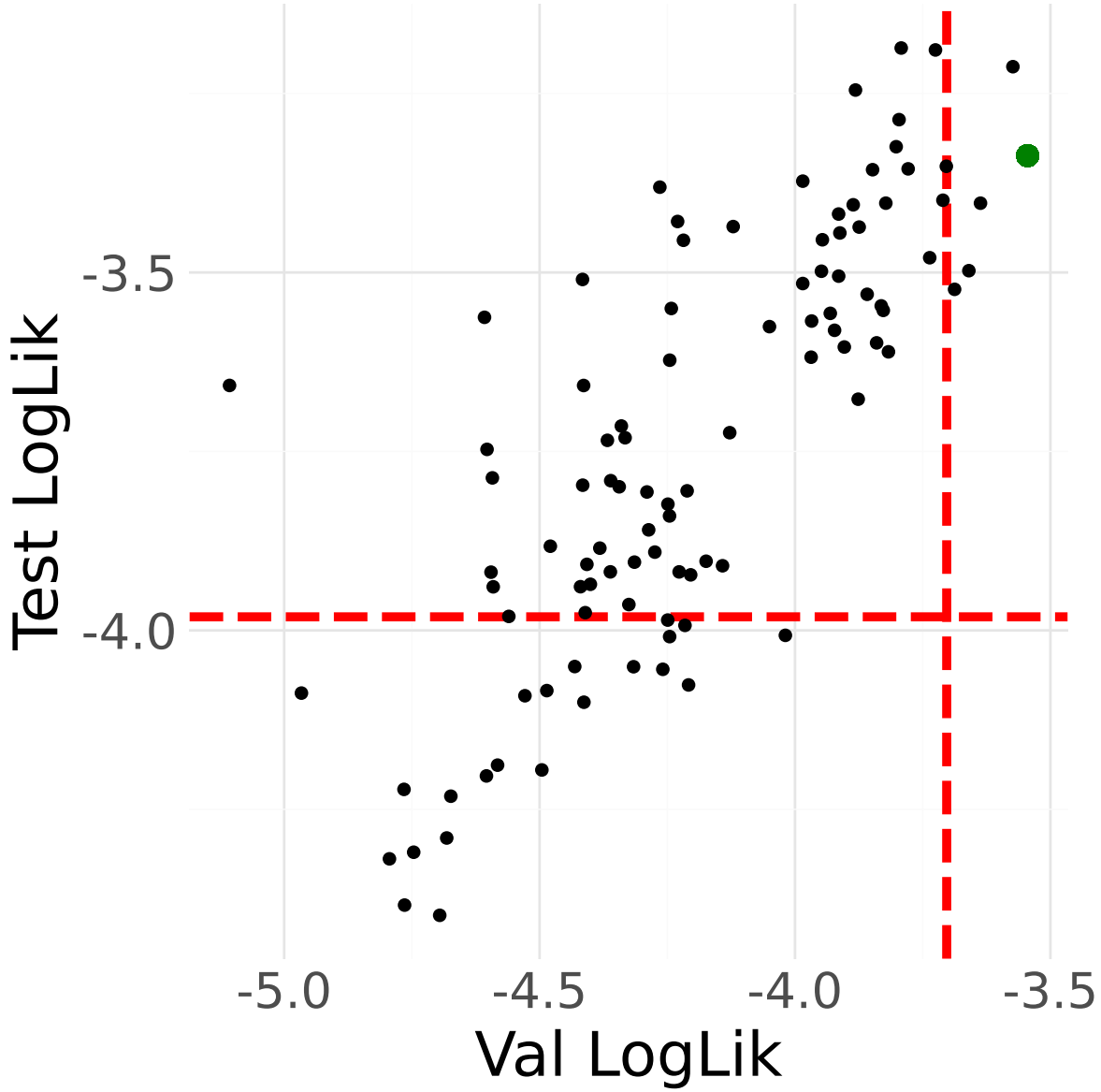
Concrete UCI Dataset



Is this it ?


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Airfoil UCI Dataset



Model Confidence Sets (MCS [1])

- Notation : $\mathbb{P}'(\cdot) = \mathbb{P}(\cdot \mid \mathcal{D}_{\text{train}})$, $\mathbb{E}'[\cdot] = \mathbb{E}[\cdot \mid \mathcal{D}_{\text{train}}]$, and $R'(\mathcal{V}) = \mathbb{E}'[L(\widehat{f}_{\mathcal{V},\mathcal{D}_{\text{train}}}, \boldsymbol{Z})]$.
- Consider
  - $\Theta = \{\mathcal{V}_1, \ldots, \mathcal{V}_M\}$ a set of M candidate structures,
  - $\Theta^* = \arg\min_{\mathcal{V} \in \Theta} R'(\mathcal{V})$ the subset of optimal structure(s).
- A random set $\widehat{\Theta} \subseteq \Theta$ is an α-level MCS if

$$\inf_{\mathcal{V} \in \Theta^*} \mathbb{P}'(\mathcal{V} \in \widehat{\Theta}) \geq 1 - \alpha,$$

1. See Hansen et al. (2011), Zhang et al. (2024), Kim and Ramdas (2025).

From a discrete argmin test to the MCS

- For any $\mathcal{V} \in \Theta$, let
  - $H_{0,\mathcal{V}}$: $R'(\mathcal{V}) \leq \min_{\mathcal{W}\in\Theta} R'(\mathcal{W})$,
  - $\widehat{T}_{\mathcal{V}}$ be the dimension-agnostic (DA) statistic of Kim and Ramdas (2025), i.e. $\widehat{T}_{\mathcal{V}} \to N(0,1)$ under $H_{0,\mathcal{V}}$.

Algorithm MCS for Vines.

Input: Same as before + confidence parameter α.
Output: A set of structures $\widehat{\Theta}$.

1: Follow Steps 1-4 from before.
2: For each $\mathcal{V} \in \Theta$, split $\mathcal{D}_{\text{val}}$ into two halves and use :
3: Half to identify the strongest competitor $\mathcal{V}_{\Theta\setminus\{\mathcal{V}\}}$;
4: Half to compute $\widehat{T}_{\mathcal{V}}$ from loss differences btw $\mathcal{V}$ and $\mathcal{V}_{\Theta\setminus\{\mathcal{V}\}}$.
5: Invert the test to form $\widehat{\Theta} = \{\mathcal{V} : \widehat{T}_{\mathcal{V}} \leq \Phi^{-1}(1-\alpha)\}$.
6: Return the MCS $\widehat{\Theta}$.

Theoretical guarantees & ensembling

Theorem

Whenever $\limsup_{n\to\infty} \max_{\mathcal{V}\in\Theta} \mathbb{E}'[|L(\widehat{f}_{\mathcal{V},\mathcal{D}_{train}}, \boldsymbol{Z})|^3] < \infty$, *the algorithm yields*
(i) $\liminf_{n\to\infty} \mathbb{P}'(\mathcal{V} \in \widehat{\Theta}) \geq 1-\alpha$ *for all* $\mathcal{V} \in \Theta^*$,
(ii) $\lim_{n\to\infty} \mathbb{P}'\big(\exists \mathcal{V} \in \widehat{\Theta}\colon R'(\mathcal{V}) - R'(\mathcal{V}^*) > a_n/\sqrt{n}\big) \to 0$ *for any* $\mathcal{V}^* \in \Theta^*$ *and* $a_n \to \infty$.

Combine models in $\widehat{\Theta}$ by ensembling :

$$\widehat{f}_{\widehat{\Theta}}(\boldsymbol{z}) = \frac{1}{|\widehat{\Theta}|} \sum_{\mathcal{V}\in\widehat{\Theta}} \widehat{f}_{\mathcal{V}}(\boldsymbol{z})$$

For any loss convex in f :

$$R(\widehat{f}_{\widehat{\Theta}}) \leq \frac{1}{|\widehat{\Theta}|} \sum_{\mathcal{V}\in\widehat{\Theta}} R'(\mathcal{V}) \leq R'(\mathcal{V}^*) + O_{\mathbb{P}}(a_n/\sqrt{n}),$$

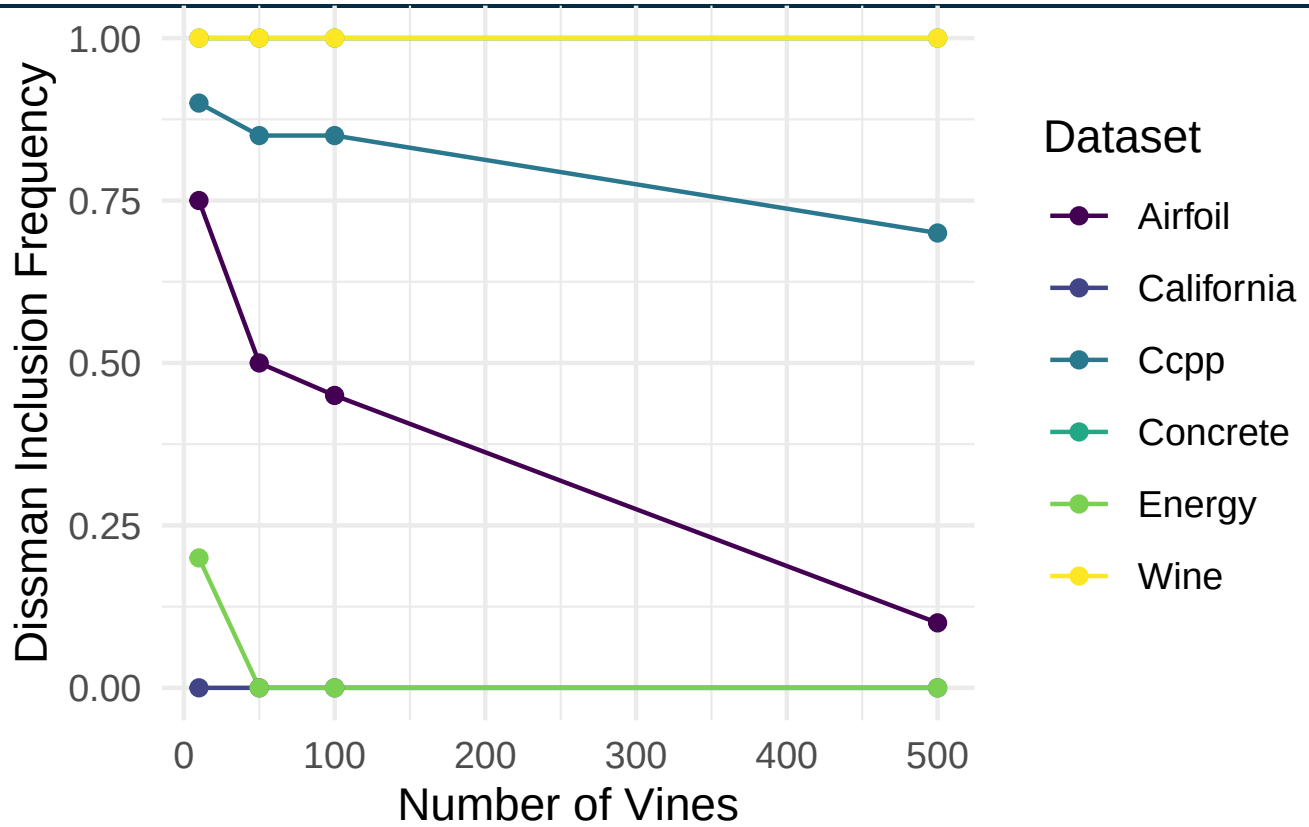
Agenda

Experimental Setup

Dataset	Energy	Concrete	Airfoil	Wine	Ccpp	California
Samples	768	1030	1503	1599	9568	20640
Features	8	8	5	11	4	8

- Tasks :
  - Density estimation (log-likelihood)
  - Mean regression (RMSE)
  - Probabilistic forecasting (CRPS)
- Models :
  - Dissmann et al. (2013) heuristic
  - (Kraus and Czado, 2017b) heuristic / (Kraus and Czado, 2017a) D-Vines
  - Random Search + MCS ensembling (RS-E)

Density Estimation Results



	Energy	Concrete	Airfoil	Wine	Ccpp	California
Dissmann	1.95 (0.35)	7.14 (0.11)	3.25 (0.04)	8.49 (0.19)	6.8 (0.01)	3.63 (0.16)
Kraus	1.51 (0.32)	7.16 (0.12)	3.27 (0.04)	8.39 (0.15)	6.8 (0.01)	3.6 (0.17)
RS-E (50)	0.12 (0.43)	6.55 (0.1)	3.05 (0.05)	8.11 (0.18)	6.74 (0.01)	3.41 (0.16)
RS-E (100)	0.13 (0.34)	6.51 (0.09)	3.03 (0.04)	8.06 (0.19)	6.74 (0.01)	3.36 (0.16)
RS-E (500)	**-0.28 (0.32)**	**6.49 (0.09)**	**3.0 (0.04)**	**7.92 (0.18)**	**6.73 (0.01)**	**3.31 (0.17)**

Prediction via Estimating Equations

- **Target :** $\beta(\boldsymbol{x})$, a functional from the conditional distribution, e.g. :
  - $\beta(\boldsymbol{x}) = \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]$ (conditional mean) ;
  - $\beta(\boldsymbol{x}) = F^{-1}_{Y|\boldsymbol{X}=\boldsymbol{x}}(\tau)$ (conditional τ-quantile).
- Let ψ_β be a function s.t.

$$\mathbb{E}[\psi_\beta(Y) \mid \boldsymbol{X} = \boldsymbol{x}] = 0 \iff \beta(\boldsymbol{x}) \text{ is the target,}$$

  e.g., $\psi_\beta(y) = y - \beta$, $\psi_\beta(y) = \mathbb{1}_{y<\beta} - \tau$, etc.
- Solve the copula-based empirical counterpart [2] using

$$\int \psi_{\beta(y)}\, \widehat{f}_{\widehat{\Theta}}(y \mid \boldsymbol{x})\, dy \propto \sum_{g=1}^{G} \psi_\beta(y_g) \widehat{f}_Y(y_g) \sum_{\mathcal{V} \in \widehat{\Theta}} \widehat{c}_{\mathcal{V},\mathcal{D}}(\widehat{F}_Y(\boldsymbol{x}), \widehat{F}_Y(y_g)),$$

  with equally spaced grid points $\{y_1, \ldots, y_G\}$.

2. Nagler and Vatter (2024)

Mean Regression & Probabilistic Forecasting

Average RMSE (standard error) on test data. Best results in bold :

	Energy	Concrete	Airfoil	Wine	Ccpp	California
`Dissmann`	2.76 (0.1)	7.12 (0.13)	4.52 (0.08)	0.63 (0.01)	4.09 (0.03)	0.66 (0.0)
`Kraus`	3.4 (0.13)	7.09 (0.14)	4.48 (0.1)	0.63 (0.01)	4.26 (0.02)	0.65 (0.0)
`RS-E (50)`	2.0 (0.06)	6.28 (0.1)	**3.73 (0.06)**	0.61 (0.01)	4.07 (0.03)	0.6 (0.0)
`RS-E (100)`	2.04 (0.07)	6.24 (0.1)	3.78 (0.06)	0.61 (0.0)	4.06 (0.03)	0.6 (0.0)
`RS-E (500)`	**1.87 (0.06)**	**6.24 (0.08)**	3.79 (0.07)	**0.61 (0.01)**	**4.06 (0.03)**	**0.6 (0.0)**

Average CRPS [3] (standard error) on test data. Best results in bold :

	Energy	Concrete	Airfoil	Wine	Ccpp	California
`Dissmann`	1.36 (0.03)	3.91 (0.07)	2.38 (0.04)	0.31 (0.0)	2.25 (0.01)	0.33 (0.0)
`Kraus`	1.41 (0.05)	3.94 (0.06)	2.37 (0.04)	0.31 (0.0)	2.38 (0.01)	0.33 (0.0)
`RS-E (50)`	0.93 (0.02)	3.39 (0.05)	**1.94 (0.03)**	0.29 (0.0)	2.24 (0.01)	0.29 (0.0)
`RS-E (100)`	0.95 (0.02)	3.37 (0.05)	1.97 (0.03)	0.29 (0.0)	2.24 (0.01)	0.29 (0.0)
`RS-E (500)`	**0.89 (0.02)**	**3.37 (0.04)**	1.97 (0.03)	**0.29 (0.0)**	**2.23 (0.01)**	**0.29 (0.0)**

3. $\text{CRPS}(F, y) = 2\int_0^1 \rho_\tau(y - F^{-1}(\tau))d\tau$ with $\rho_\tau(u) = u(\tau - \mathbb{1}\{u < 0\})$

Agenda

Conclusion

- Simple random search + MCS outperforms existing heuristics.
- Embarassingly parallel, theoretical guarantees.
- `sklearn` compatible API, will make it into `pyvinecopulib`.
- Future work : faster, better, smarter search strategies.

Thank you !

References I

Chang, B., Pan, S., and Joe, H. (2019). Vine copula structure learning via Monte Carlo tree search. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 353–361. PMLR.

Dissmann, J., Brechmann, E. C., Czado, C., and Kurowicka, D. (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59 :52–69.

Gruber, L. F. and Czado, C. (2015). Sequential Bayesian model selection of regular vine copulas. *Bayesian Analysis*, 10(4) :937–963.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2) :453–497.

Joe, H., Cooke, R. M., and Kurowicka, D. (2011). Regular vines : generation algorithm and number of equivalence classes. *Dependence Modeling : Vine Copula Handbook*, pages 219–231.

Kim, I. and Ramdas, A. (2025). Locally minimax optimal and dimension-agnostic discrete argmin inference. *arXiv preprint arXiv :2503.21639*.

Kraus, D. and Czado, C. (2017a). D-vine copula based quantile regression. *Computational Statistics & Data Analysis*, 110C :1–18.

Kraus, D. and Czado, C. (2017b). Growing simplified vine copula trees : improving Dißmann's algorithm. *arXiv :1703.05203*.

Morales-Nápoles, O. (2011). Counting vines. In Kurowicka, D. and Joe, H., editors, *Dependence Modeling : Vine Copula Handbook*, chapter 9, pages 189–218. Singapore, SG : World Scientific.

Nagler, T., Schepsmeier, U., Stoeber, J., Brechmann, E. C., Graeler, B., and Erhardt, T. (2025). *VineCopula : Statistical Inference of Vine Copulas*. R package version 2.6.1.

Nagler, T. and Vatter, T. (2024). Solving Estimating Equations With Copulas. *Journal of the Americal Statistical Association*, 119(546) :1168–1180.

Nagler, T. and Vatter, T. (2025a). *pyvinecopulib : High Performance Algorithms for Vine Copula Modeling*. Python package version 0.7.1, DOI : 10.5281/zenodo.10435751.

Nagler, T. and Vatter, T. (2025b). *rvinecopulib : High Performance Algorithms for Vine Copula Modeling*. R package version 0.7.1.1.1, DOI : 10.32614/CRAN.package.rvinecopulib.

References II

Nagler, T. and Vatter, T. (2025c). *vinecopulib : High Performance C++ Algorithms for Vine Copula Modeling*. C++ library version 0.7.3.

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de L'Institut de Statistique de L'Université de Paris*, 8 :229–231.

Sun, Y., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Learning Vine Copula Models for Synthetic Data Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01) :5049–5057.

Zhang, T., Lee, H., and Lei, J. (2024). Winners with confidence : Discrete argmin inference with an application to model selection. *arXiv preprint arXiv :2408.02060*.

Copulas

- **Copula :** multivariate distribution with uniform marginals.
- **(Sklar, 1959) :** Any multivariate distribution F can be expressed in terms of its marginals $F_1, \ldots, F_d$ and a copula C.

$$F(x) = C\{F_1(x_1), \ldots, F_d(x_d)\},$$

$$f(x) = c(F_1(x_1), \ldots, F_d(x_d)) \prod_{j=1}^{d} f_j(x_j).$$

- c, C available for some families (e.g., Elliptical, Archimedean) : evaluation, sampling, etc are straightforward.
- **However :** few flexible families for $d > 2$, strong restrictions (e.g., exchangeable dependence, symmetric tails).
- **Vines :** products of bivariate copulas, both flexible and tractable.

Vine Copulas

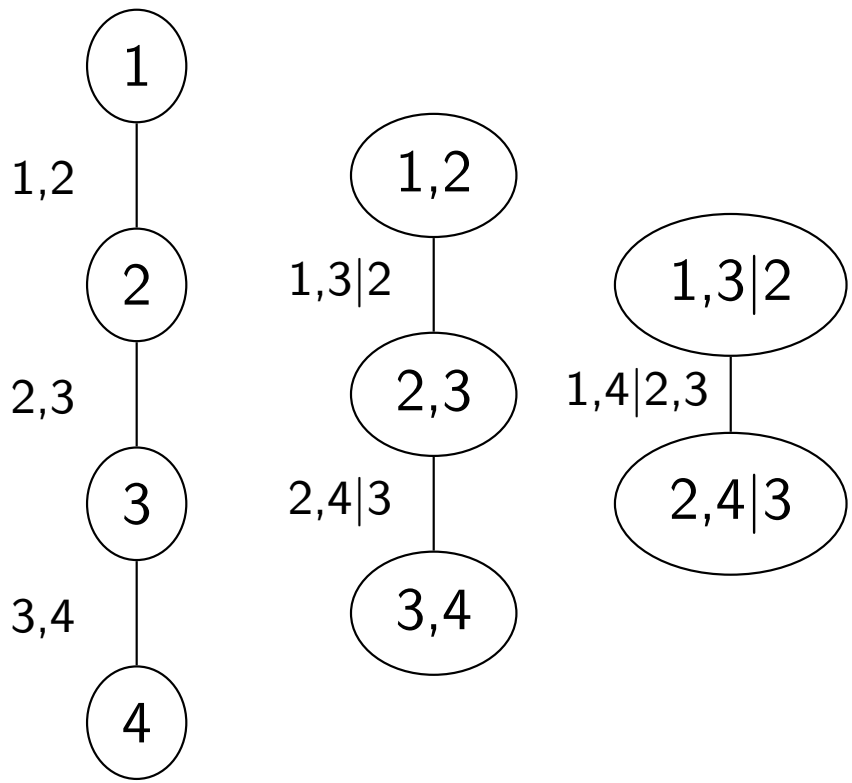
For $u_{j_e|D_e} = C_{j_e|D_e}(u_{j_e}|\boldsymbol{u}_{D_e})$, $C_{j_e|D_e}$ the cdf of $U_{j_e}|\boldsymbol{U}_{D_e}$, etc. :

$$c(\boldsymbol{u}) = \prod_{t=1}^{d-1} \prod_{e \in E_t} c_{j_e,k_e|D_e}(u_{j_e|D_e}, u_{k_e|D_e}).$$

Definition (Vine Structure)

A sequence of nested trees (V_t, E_t) satisfying :

1. $V_1 = \{1, \ldots, d\}$
2. $V_t = E_{t-1}$ for $t \geq 2$
3. Proximity condition on shared nodes



The DA discrete argmin statistic [4]

Algorithm DA discrete argmin statistic

Input: Losses $\{L_{i,m}\}_{1\le i\le N,\, 1\le m\le M}$, target index $r \in \{1,\dots,M\}$.

Output: Test statistic $\widehat{T}_{\mathcal{V}_r}$.

1: Split indices : $I_1 = \{1,\dots,\lfloor N/2\rfloor\}$ and $I_2 = \{\lfloor N/2\rfloor+1,\dots,N\}$.

2: On I_1, find $\widehat{m}_{-r} \in \arg\min_{m\in\{1,\dots,M\}\setminus\{r\}} \sum_{i\in I_1} L_{i,m}$.

3: On I_2, with $\Delta^{(i)}_{m,k} = L_{i,m} - L_{i,k}$ for $1 \le i \le N$, $1 \le m,k \le M$, compute

$$\widehat{T}_{\mathcal{V}_r} \;=\; \frac{1}{\widehat{\sigma}_{r,\widehat{m}_{-r}}} \cdot \frac{1}{\sqrt{|I_2|}} \sum_{i\in I_2} \Delta^{(i)}_{r,\widehat{m}_{-r}},$$

with $\widehat{\sigma}^2_{r,\widehat{m}_{-r}}$ the sample variance of $\{\Delta^{(i)}_{r,\widehat{m}_{-r}} : i \in I_2\}$.

4. (Kim and Ramdas, 2025)