Dependence modeling and computational algorithms: A surprising symbiosis

Radu Craiu

Department of Statistical Sciences
University of Toronto

Toronto - June 4, 2025

Two themes

- Methods for modelling dependence patterns in the data so we can:
  - Fuse different data streams (today)
  - Extend the range of statistical models to capture complex generative processes
  - Improve inference (prediction, estimation, etc) (today)
  - Modelling dependence is accompanied by computational challenges. (today)
- Design of efficient computational algorithms for sampling or optimization.
  - Sampling is of paramount importance to Bayesian statisticians
  - Optimization is important to everyone
  - Introducing dependence in the design of sampling algorithms can improve performance (today)
  - Coming up with the "right" dependence is challenging.

Outline

Serially correlated data with hidden structures

Hidden Markov Models with Multivariate Observations

Copula Generalization

Estimation and Computation

A unified approach to antithetic sampling

The Antithetic Swindle

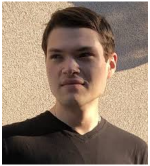
Antithetic Sampling techniques

Desirable properties

Sampling on Segments

Numerical Illustration

Copulas for serially correlated data with hidden structures



Robert Zimmerman (Imperial College London)

Vianey Leos Barajas (Toronto)

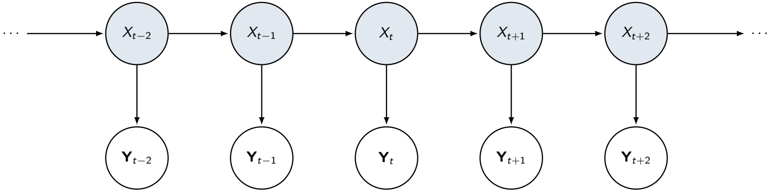
Paper: Copula Modelling of Serially Correlated Multivariate Data with Hidden Structures (JASA, 2024).

Hidden Markov Models: An Example

- We observe a system in time, e.g. the evolution of several stocks, the humidity and temperature in a room, the number of people late for work in Toronto, etc.
- We believe that these measurements are informative about variables that are not observed directly (they are hidden):
  - Stocks $--$ > State of Economy
  - Room H and T $--$ > State of occupancy
  - People late $--$ > Traffic level
  - The hidden variables are not constant, but they also change in time.

Hidden Markov Models: Briefly put

- A hidden Markov model (HMM) pairs an observed time series $\{\mathbf{Y}_t\}_{t\geq 1} \subseteq \mathbb{R}^d$ with a Markov chain $\{X_t\}_{t\geq 1}$ on some state space $\mathcal{X}$, such that the distribution of $\mathbf{Y}_s \mid X_s$ is independent of $\mathbf{Y}_t \mid X_t$ for $s \neq t$:



- $\mathbf{Y}_{t,h}|\{X_t = k\} \sim f_{k,h}(\cdot|\lambda_{k,h})\ \forall h = 1, \ldots, d$
- $\{X_t\}$ is a Markov process (finite state space $\mathcal{X}$) with initial probability mass distribution $\{\pi_i\}_{i\in\mathcal{X}}$ and transition probabilities $\{\gamma_{i,j}\}_{i,j\in\mathcal{X}}$

Inferential aims for HMMs

- Typically, the chain $\{X_t\}_{t\geq 1}$ is partially or completely unobserved.
- The hidden states can correspond to a precise variable (occupancy data) or might be postulated (psychology, ecology, etc)
- Aim 1: Model the data generating mechanism Nasri et al. (2020)
- Aim 2: Decode (i.e., classify) or predict the X_t's from the observed data.

Fusion of Multiple Data Sources

- In real-world applications (sports, stock exchange, animal movement, etc), various sensors capture multiple streams of data, which are "fused" into a multivariate time series $\{\mathbf{Y}_t\}_{t\geq 1}$
- In such situations, the components of any $\mathbf{Y}_t = (Y_{t,1}, \ldots, Y_{t,d})$ cannot be assumed independent (even conditional on X_t)
- Instead, it is common to assume that $\mathbf{Y}_t$ follows a multivariate Gaussian distribution, but this places limits on marginals and dependence structures
- What if the strength of dependence between the components of $\mathbf{Y}_t$ could be informative about the underlying state X_t?

Occupancy Data

- The ability to detect whether a room is occupied using sensor data (such as temperature and CO_2 levels)
- Consider three publicly-available labelled datasets presented by Candanedo and Feldheim (2016) which contain multivariate time series of four environmental measurements (light, temperature, humidity, CO_2) and one derived metric (the humidity ratio), as well as binary indicators for whether the room was occupied or not at the time of measurement

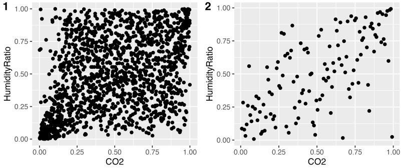


Figure: Pseudo-observations computed from unoccupied (Panel 1) and occupied (Panel 2) subsets.

At the root of it all, a theorem

- Copulas are distribution functions on $[0,1]^d$ that model dependence between continuous random variables.
- Sklar's Theorem: If $Y_1, Y_2, \ldots Y_d$ are continuous r.v.'s with distribution functions (df) $F_1, \ldots, F_d$, there exists an unique copula function $C : [0,1]^d \to [0,1]$ such that

$$H(t_1, \ldots, t_d) = \mathbb{P}(Y_1 \leq t_1, \ldots, Y_d \leq t_d) = C(F_1(t), \ldots, F_d(t_d)).$$

- The copula bridges the marginal distributions of $Y_1, \ldots, Y_d$ with the joint distribution. It corresponds to a distribution on $[0,1]^d$ with uniform margins.

Copulas: The Joys

- Copulas are mathematical devices used to model dependence between random variables regardless of their marginals.
- Copulas are useful for data fusion/integration as they lead to coherent joint models, even when the marginals are in different families or of different types.
- Copulas unlock information contained in the dependence part of the distribution (second-order) that complements the information in the marginals.
- Copulas extend statistical methods beyond the use of a multivariate Gaussian or Student.

Copulas Within HMMs

- Here we consider a HMM $\{(\mathbf{Y}_t, X_t)\}_{t\geq 1} \subseteq \mathbb{R}^d \times \mathcal{X}$ in which the state-dependent distributions use copulas:

$$\mathbf{Y}_t \mid (X_t = k) \sim H_k(\cdot) = \underbrace{C_k\Big(F_{k,1}(\cdot\,;\lambda_{k,1}), \ldots, F_{k,d}(\cdot\,;\lambda_{k,d}) \,\Big|\, \theta_k\Big)}_{\text{depends on the hidden state value } k}.$$

- $C_k(\cdot, \ldots, \cdot \mid \theta_k)$ is a d-dimensional parametric copula
- $\{X_t\}_{t\geq 1}$ is a Markov process on finite state space $\mathcal{X} = \{1, 2, \ldots, K\}$ and K is known.
- In this model, all aspects of the state-dependent distributions are allowed to vary between states

Information in the dependence

- For a range of $\theta \in [0, 100)$, we simulated a bivariate time series of length $T = 100$ from the 2-state HMM

$$\mathbf{Y}_t \mid (X_t = k) \sim C_{\text{Frank}}\left(\mathcal{N}(0,1), \mathcal{N}(0,1) \mid (-1)^k \cdot |\theta|\right), \quad k = 1, 2$$

and then separately assessed the accuracy of a standard decoding algorithm, first assuming independent margins and then the true model:

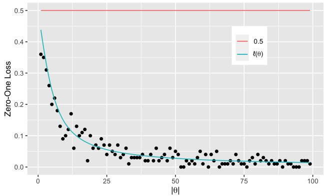


Figure: Zero-one losses for independent margins (red dots) and true model (blue dots)

Estimation with missing data

- Data consist in observed $\mathbf{Y}_{1:T}$ and missing $X_{1:T}$
- Parameters are $\eta = \{\lambda_{h,k}\}_{\substack{h=1:d \\ k=1:T}} \cup \{\theta_k\}_{k=1:T} \cup \{\gamma_{i,j}\}_{\substack{i=1:K \\ j=1:K}}$ $\cup \{\pi_j\}_{j=1:K}$.
- The complete-data log-likelihood for one trajectory of the copula HMM is given by

$$\ell_{\text{com}}(\boldsymbol{\eta} \mid \mathbf{y}_{1:T}, X_{1:T}) = \pi_{X_1} + \sum_{t=2}^{T} \log \gamma_{X_{t-1},X_t} + \sum_{h=1}^{d} \log f_{X_t,h}(y_{t,h}; \lambda_{X_t,h})$$
$$+ \sum_{t=1}^{T} \log c_{X_t}\left(F_{X_t,1}(y_{t,1}; \lambda_{X_t,1}), \ldots, F_{X_t,1}(y_{t,d}; \lambda_{X_t,d}) \mid \theta_{X_t}\right). \qquad (1)$$

Computation for HMMs Via the EM Algorithm

- Without copula, the estimation is done via the EM algorithm
- The **E-Step** is straightforward
- The maximization required by the **M-Step** is unstable or plain unfeasible.
- The solution we found was to perform optimization in two steps:
  - First optimize the parameters of the marginal distributions
  - Second, optimize the parameters of the copulas after plugging in the marginal estimates obtained in the previous step.
- However, this approach changes the nature of the algorithm which is no longer "EM-like" so a proof of its validity is required.

Does This Work?

- For $T \in \{100, 1000, 5000\}$ and $d \in \{2, 5, 10\}$, we simulated a d-dimensional time series of length T from the 2-state HMM

$$\mathbf{Y}_t \mid (X_t = 1) \sim C_{\text{Frank}}\left((\mathcal{N}(\mu_{1,h} = -h, 1))_{h=1}^{d} \mid \theta_1 = 3\right)$$
$$\mathbf{Y}_t \mid (X_t = 2) \sim C_{\text{Clayton}}\left((\mathcal{N}(\mu_{2,h} = h, 1))_{h=1}^{d} \mid \theta_2 = 3\right)$$

  and estimated $\boldsymbol{\eta} = (\mu_{1,1}, \ldots, \mu_{2,d}, \theta_1, \theta_2)$ using both approaches

- Applied to the basic EM algorithm, R's `optim` with `L-BFGS-B` (i.e., quasi-Newton with box constraints) typically fails as soon as $d \geq 3$
  - The procedure is extremely sensitive to initial values and requires $\hat{\boldsymbol{\eta}}^{(0)} \approx \boldsymbol{\eta}$ just to avoid overflow
  - This kind of tuning is very tedious or impossible in high dimensions

Does This Work?

- We keep track of the time (in seconds) until the algorithm converges, and the L_2 error of the resulting estimate, $\epsilon = \|\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}\|_2$
  - We used the `lbfgsb3c` package, which is more stable than `optim`

	$d = 2$	$d = 5$	$d = 10$
$T = 100$	111.9 s, $\epsilon = 0.14$	123.4 s, $\epsilon = 299.98$	111.8 s, $\epsilon > 10^9$
$T = 1000$	166.6 s, $\epsilon = 0.63$	169.5 s, $\epsilon > 10^{11}$	418.23 s, $\epsilon = 725.06$
$T = 5000$	?	?	?

Table: EM Algorithm

	$d = 2$	$d = 5$	$d = 10$
$T = 100$	5.1 s, $\epsilon = 0.29$	3.0 s, $\epsilon = 0.94$	4.2 s, $\epsilon = 0.58$
$T = 1000$	34.4 s, $\epsilon = 0.57$	22.9 s, $\epsilon = 0.60$	34.4 s, $\epsilon = 0.80$
$T = 5000$	172.6 s, $\epsilon = 0.13$	106.2 s, $\epsilon = 0.12$	168.7 s, $\epsilon = 0.19$

Table: EFM Algorithm

Numerical Experiment I

- Generative model:

$$\mathbf{Y}_i \mid (X_i = k) \sim C_k \left(SN(\cdot; \xi_{k,1}, \omega_{k,1}, \alpha_{k,1}), SN(\cdot; \xi_{k,2}, \omega_{k,2}, \alpha_{k,2}) \mid \tau_k \right),$$

for $k \in \{1, \ldots, 4\}$.

State	Copula family	τ_k	$\xi_{k,1}$	$\omega_{k,1}$	$\alpha_{k,1}$	$\xi_{k,2}$	$\omega_{k,2}$	$\alpha_{k,2}$
1	Clayton	0.2	-4	1	5	-1	1	-3
2	B4	0.4	-2	1	3	2	1	-3
3	Gaussian	0.6	0	1	5	3	1	-5
4	$t_{(\nu=5)}$	0.8	2	1	3	4	1	-5

Table: True parameters for the state-dependent distributions.

Numerical Experiment I

T:		500	1000	2500	5000
Stopping Rule Tolerance:	0.01	14	24	23	15
	0.001	17	26	25	17
	0.0001	36	59	62	39
	0.00001	230	115	460	269
Classifier:	k-means	0.9020	0.9090	0.9200	0.9196
	Local state decoding	0.9640	0.9640	0.9696	0.9732

Table: For each $T \in \{500, 1000, 2500, 5000\}$: (Top rows) Number of iterations taken by the EIFM algorithm applied to $\mathbf{Y}_{1:T}$ before stopping using L_1-norm tolerances in $\{0.01, 0.001, 0.0001, 0.00001\}$. (Bottom rows) Classification accuracy of initial k-means clustering and local decoding with parameter estimates provided by the EIFM algorithm.

Occupancy Data

Classifier	Train	Test 1
k-means clustering	0.865	0.818
Independence copulas within HMM	0.895	0.846
BB7/Tawn copulas within HMM	0.900	0.852

Table: Overall state classification accuracy for the training dataset and the test dataset, using *k*-means clustering and local decoding via the HMM with independent margins and the copula-within-HMM model.

Living on the Edge: An Unified Approach to Antithetic Sampling

Roberto Casarin (Ca'Foscari Venice)

Lorenzo Frattarolo (Verona)

Christian P. Robert (Paris and Warwick)

Paper: Living on the Edge: An Unified Approach to Antithetic Sampling (Statistical Science, 2024).

Outline

Serially correlated data with hidden structures

Hidden Markov Models with Multivariate Observations

Copula Generalization

Estimation and Computation

A unified approach to antithetic sampling

The Antithetic Swindle

Antithetic Sampling techniques

Desirable properties

Sampling on Segments

Numerical Illustration

The Monte Carlo method

- The Monte Carlo method is at the core of model-based scientific exploration.
- It relies on approximating an integral of interest $I = \int f(\mathbf{x})\pi(\mathbf{x})d\mathbf{x}$ with $\hat{I}_d = \frac{1}{d}\sum_{i=1}^{d} f(\mathbf{X}_i)$ where
  - π is a probability density,
  - $f : \mathbb{R}^p \mapsto \mathbb{R}$ is a integrable function with respect to π
  - d is the Monte Carlo sample size
  - $\mathbf{X}_1, \ldots, \mathbf{X}_d$ are iid samples from π.

The Antithetic Swindle

- Techniques needed to reduce the Monte Carlo sample size d, while maintaining the desired precision in estimation, are essential.
- *Variance reduction techniques* use statistical properties induced by the sampling design to reduce $\mathbb{V}ar(\hat{I}_d)$
- If the independence condition between samples $X_1, \ldots, X_d$ is dropped then

$$\mathbb{V}ar(\hat{I}_d) = \frac{1}{d^2}\sum_{i=1}^{d} \mathbb{V}ar\left(f\left(X_i\right)\right) + \frac{1}{d^2}\sum_{i \neq j} \mathbb{C}ov\left(f\left(X_i\right), f\left(X_j\right)\right).$$

- The antithetic swindle is executed when we are able to generate $X_1, \ldots, X_d$ so that the average covariance

$$\frac{1}{d^2}\sum_{i \neq j} \mathbb{C}ov\left(f\left(X_i\right), f\left(X_j\right)\right)$$

is negative.

Pairwise Construction

- The pairwise antithetic coupling introduced by Hammersley and Morton (1956) relies on $d/2$ (assume for a moment d is even) iid pairs of negatively correlated random variables (X_{1i}, X_{2i}), $i = 1, \ldots, d/2$.
- This is achieved by sampling using the quantile coupling:

$$X_{1i} \sim \pi, \quad X_{2i} = F_{\pi}^{-1}\left(1 - F_{\pi}\left(X_{1i}\right)\right).$$

or

$$X_{1i} = F_{\pi}^{-1}(U), \quad X_{2i} = F_{\pi}^{-1}(1 - U)$$

where $U \sim U(0, 1)$

- This procedure minimizes the correlation for any monotonic f in the case $d = 2$ and $p = 1$.
- However, the result doesn't hold for $d > 2$.

Beyond pairs: Latin Hypercube Sampling

- A popular procedure applicable to the general $d \geq 2$, $p \geq 1$ is the Latin Hypercube sampling (McKay et al., 1979).
- Given a standard uniform d-dimensional random vector $\mathbf{V}$ and $\mathcal{D}^{\sigma} = (\sigma(0), \ldots, \sigma(d-1))^T$, a permutation of $\{0, 1, \ldots, d-1\}$ independent of $\mathbf{U}$, set

$$\mathbf{U} = \frac{1}{d}\left(\mathcal{D}^{\sigma} + \mathbf{V}\right). \tag{2}$$

- $d = 2$ and $D^{\sigma} = (0, 1)$:

$$U_1 = \frac{V_1}{2} \in \left(0, \frac{1}{2}\right), \; U_2 = \frac{1 + V_2}{2} \in \left(\frac{1}{2}, 1\right).$$

- $d = 3$ and $D^{\sigma} = (2, 0, 1)$:

$$U_1 = \frac{2 + V_3}{3} \in \left(\frac{2}{3}, 1\right), \; U_2 = \frac{V_1}{3} \in \left(0, \frac{1}{3}\right), \; U_3 = \frac{1 + V_2}{3} \in \left(\frac{1}{3}, \frac{2}{3}\right).$$

Concordance Order

- Let $\mathbf{X}$ and $\mathbf{Y}$ be random vectors with CDFs F and G, respectively.
- $\mathbf{Y}$ is more concordant than $\mathbf{X}$ (written $\mathbf{X} \prec_C \mathbf{Y}$) if

$$F(x) \leq G(x), \;\; \forall x \in \mathbb{R}^d.$$

- If $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^d$ satisfy $\mathbf{X} \prec_C \mathbf{Y}$ then

$$\mathbb{V}ar\left(\sum_{h=1}^{d} b_h f\left(X_h\right)\right) \leq \mathbb{V}ar\left(\sum_{h=1}^{d} b_h f\left(Y_h\right)\right), \; \forall \mathbf{b} \in \mathbb{R}^d$$

Countermonotonicity

- Consider a random vector $\mathbf{U} \in \mathbb{R}^d$ with uniform marginals
- $\mathbf{U}$ is said to be d-countermonotonic (d-CTM), if there exists a family $\{g_h\}_{1 \leq h \leq d}$ of strictly increasing continuous functions $[0, 1] \mapsto \mathbb{R}$ and some $k \in \mathbb{R}$ such that:

$$\sum_{h=1}^{d} g_h(U_h) = k \text{ a.s.} \tag{3}$$

- It has been shown that the set of d-CTM vectors is contained in the subset of elements minimal in concordance order Lee et al. (2017)
- Bottom line: construct $\mathbf{U} \in \mathbb{R}^d$ with uniform marginals such that $\sum_{l=1}^{d} g_l(U_l) = const$ (often $g_l(u_l) = u_l$)
- Note that the LH vector $\mathbf{U}$ is d-CTM iff $\mathbf{V}$ is d-CTM.

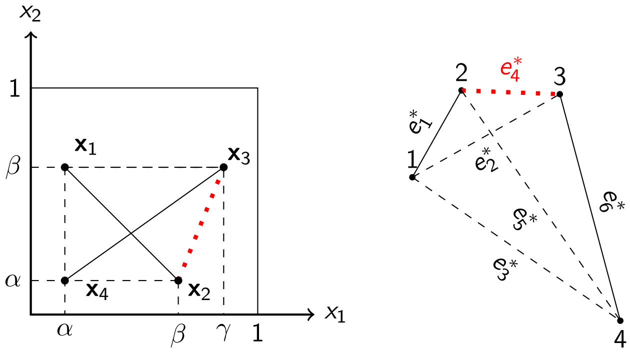
Sampling on segments

- Sampling with equal probability on a collection $\mathcal{S}$ of line segments in the d-dimensional Euclidean space.
- Since each segment is uniquely characterized by its endpoints or vertexes, the collection $\mathcal{S}$ can be equivalently represented by the set of vertex pairs that define the segments and their coordinates.
- This representation is efficient in large dimensions even when the segments share some of their vertexes.

Set up and notation

- Consider the vertex set $\mathcal{V} = \{1, \ldots, n\}$ as a set of points in the d-dimensional hypercube
- The coordinates of the k-th vertex form the column vector $\mathbf{x}_k \equiv (x_{1k}, \ldots, x_{dk})^T \in [0,1]^d$
- The coordinate matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ as the collection of vertex coordinates.
- There is an edge $e = (i, j)$ between i and j, with $i < j$, if there is a segment joining the two vertices i and j

Illustration



Consider the collection of segments in the left plot with coordinate matrix

$$\mathbf{X} = \begin{pmatrix} \alpha & \beta & \gamma & \alpha \\ \beta & \alpha & \beta & \alpha \end{pmatrix}, \tag{4}$$

where $\alpha < \beta \leq \gamma \in \mathbb{R}$. The edge set is: $\mathcal{E}^* = \{e_1^* = (1,2)\,,\;\; e_2^* = (1,3)\,,$ $e_3^* = (1,4)\,,\; e_4^* = (2,3)\,,\; e_5^* = (2,4)\,,\; e_6^* = (3,4)\}$

Sampling on segments

- The collection of segments is defined by the edge set $\mathcal{E} = \{(i,j) \in \mathcal{V} \times \mathcal{V}\}$.
- Then $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is an undirected graph and $\mathcal{S} = \{\mathcal{G}, \mathbf{X}\}$ is the collection of segments.
- The lexicographic order on vertex indexing induces an order on the edge set
- The map $\varphi_{\mathcal{E}} : \{1, \ldots, |\mathcal{E}|\} \mapsto \mathcal{E}$, $k \to (i(k), j(k))$ associates the k-th element, $e_k \in \mathcal{E}$, to its couple of vertices.

Sampling on segments

1. Draw $V \sim \mathcal{U}[0,1]$ and $W \sim \mathcal{U}[0,1]$ independently;
2. Use W to choose with uniform probability on the edge set $\mathcal{E}$ the edge e_K and obtain the random pair of vertices $(I, J) = (i(K), j(K))$ with $(i(K), j(K)) = \varphi_{\mathcal{E}}(K)$;
3. Obtain a random point on the segment joining vertices I and J with uniform probability

$$\begin{aligned} U_1 &= x_{1I} V + (1 - V) x_{1J}, \\ &\vdots \\ U_d &= x_{dI} V + (1 - V) x_{dJ}. \end{aligned} \tag{5}$$

Edge living and d-CTM

- d-CTM leads to:

$$\mathbb{E}\left[\sum_{j=1}^{d} U_j\right] = \sum_{j=1}^{d} \mathbb{E}\left[U_j\right] = \frac{d}{2}.$$

- The constant sum condition can be written as a linear restriction on the coordinates of the vertices $\mathbf{x}_k$, that is

$$\sum_{h=1}^{d} U_h = \sum_{h=1}^{d} x_{hJ} + V\left[\sum_{h=1}^{d} x_{hI} - \sum_{h=1}^{d} x_{hJ}\right] = \frac{d}{2},$$

- This needs to be valid for all V and (I, J) so all vertices should be in the hyperplane of constant sum, i.e.

$$\sum_{h=1}^{d} x_{hk} = \sum_{h=1}^{d} \sum_{m=1}^{n_l} a_{h,m} \mathbb{I}_{\mathcal{M}_{h,m}}(k) = \frac{d}{2} \qquad k = 1, \ldots, n.$$

Additional remarks

- Sampling on segments can be guaranteed to produce uniform samples in the unit hypercube under certain verifiable assumptions (about the segments).
- We recover many existing antithetic constructions.
- The joint distribution of $(U_1, \ldots, U_d)$ is a copula (remember them?).
- This is a new copula family with some unusual properties (under study).

Probit regression

- The data represent the clinical characteristics summarized by two covariates of 55 patients, of which 19 were diagnosed with lupus.
- $Y_i \sim \mathcal{B}er\left(\Phi\left(\mathbf{x}_i^T\boldsymbol{\beta}\right)\right)$ where Φ is the standard normal CDF and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^T$ is the vector of parameters.
- We introduce latent variables $\psi_i \sim \mathcal{N}\left(\mathbf{x}_i^T\boldsymbol{\beta}, 1\right)$
- The sampling algorithm iterates between these two steps:
  - Sample $\beta|\boldsymbol{\psi} \sim \mathcal{N}\left(\tilde{\boldsymbol{\beta}}, \left(X^TX\right)^{-1}\right)$ with $\tilde{\boldsymbol{\beta}} = \left(X^TX\right)^{-1}X^T\boldsymbol{\psi}$ with X the data matrix whose i-th row is $\mathbf{x}_i$.
  - $\psi_i|\beta, Y_i \sim \mathcal{TN}\left(\mathbf{x}_i^T\boldsymbol{\beta}, 1, Y_i\right)$ where $\mathcal{TN}\left(\mu, \sigma^2, Y\right)$ is a the normal distribution with mean μ and variance σ^2, truncated to be positive if $Y > 0$ or negative otherwise.

Probit regression

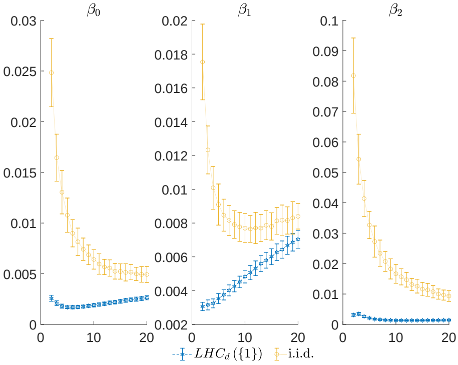


Figure: Monte Carlo variance of the posterior mean estimator (vertical axis) for different numbers of antithetic variates d (horizontal axis) In each plot: the average variance of antithetic Gibbs (blue dots) and of iid Gibbs (yellow dots) with their range (vertical segments). Note: all estimates are based on 100 independent experiments. In each experiment, the Gibbs sampler runs for 10 seconds.

References

CANDANEDO, L. M. and FELDHEIM, V. (2016). Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models. *Energy and Buildings* **112** 28–39.

HAMMERSLEY, D. C. and MORTON, K. V. (1956). A new Monte Carlo technique: antithetic variates. *Mathematical Proceedings of the Cambridge Philosophical Society* **52** 449–475.

LEE, W., CHEUNG, K. C. and AHN, J. Y. (2017). Multivariate countermonotonicity and the minimal copulas. *Journal of Computational and Applied Mathematics* **317** 589–602.

MCKAY, M. D., BECKMAN, R. J. and CONOVER, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** 239–245.

NASRI, B. R., REMILLARD, B. N. and THIOUB, M. Y. (2020). Goodness-of-fit for regime-switching copula models with application to option pricing. *Can. J. Stat.* **48** 79–96.