Matlatzinca

A PyBANSHEE-based graphical user interface for elicitation of non-parametric Bayesian networks from experts

Rongen, Guus; Morales-Nápoles, Oswaldo

DOI
10.1016/j.softx.2024.101693

Publication date
2024

Document Version
Final published version

Published in
SoftwareX

SoftwareX 26 (2024) 101693

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Software update

Matlatzinca: A PyBANSHEE-based graphical user interface for elicitation of non-parametric Bayesian networks from experts

Guus Rongen[a,b,*], Oswaldo Morales-Nápoles[a]

[a] Civil Engineering and Geosciences, Delft University of Technology, The Netherlands
[b] Pattle Delamore Partners Ltd., New Zealand

ARTICLE INFO

Keywords:
Expert judgment
Dependence modeling
Graphical user interface

ABSTRACT

This article describes the development of a GUI that addresses the challenge of eliciting dependencies between uncertain quantities elicited by experts. While software for eliciting univariate uncertainties is widely available, the mathematical complexity of multivariate dependence models makes direct elicitation difficult. To overcome this, we developed Matlatzinca,[1] a GUI built on top of the Python module PyBANSHEE. The GUI facilitates the elicitation process and allows experts to model dependencies using a non-parametric Bayesian network without the need for *ad hoc* programming. A recent practical application shows that the developed GUI is a useful tool for performing dependence elicitations, highlighting the significance of the program for dependence assessment with expert judgment.

Code metadata

Current code version	v1.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00055
Code Ocean compute capsule	–
Legal Code License	GNU General Public License
Code versioning system used	None
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	Python version 3.8+ with modules PyQt5, Numpy, Matplotlib, and Pydantic
If available Link to developer documentation/manual	–
Support email for questions	g.w.f.rongen@tudelft.nl

Software metadata

Current software version	v1.0.0
Permanent link to executables of this version	https://github.com/grongen/Matlatzinca/releases/tag/v1.0.0
Legal Software License	List one of the approved licenses
Computing platforms/Operating Systems	Microsoft Windows
Installation requirements & dependencies	–
If available, link to user manual - if formally published include a reference to the publication in the reference list	Quick-start is included with the program and available from within the program. The documentation is included in the repository: *./doc/build/html/index.html*.
Support email for questions	g.w.f.rongen@tudelft.nl

DOI of original article: https://doi.org/10.1016/j.softx.2020.100588.
* Corresponding author at: Civil Engineering and Geosciences, Delft University of Technology, The Netherlands.
E-mail address: g.w.f.rongen@tudelft.nl (G. Rongen).

[1] Matlatzinca may be translated from Nahuatl (the language of the Aztecs) to English as "The people that make nets". This is the name that the Aztecs gave to the inhabitants of the Valley of Toluca in central Mexico who were well known fishermen at the time. Our first release of a GUI for PyBANSHEE is meant for people that wish to quantify (make) Non-parametric Bayesian Networks (Nets) with expert judgments. Hence we name our GUI Matlatzinca.

https://doi.org/10.1016/j.softx.2024.101693
Received 8 February 2023; Accepted 13 March 2024
Available online 22 March 2024

1. Motivation and significance

In this update paper we present the first stand-alone open source Graphical User Interface (GUI) of PyBANSHEE [1,2]: Matlatzinca. Matlatzinca uses the computational engine of PyBANSHEE to assist experts in quantifying Non-Parametric Bayesian Networks (NPBNs) [3]. The methodology for quantifying NPBNs through expert judgments has been discussed in [4] and applications of the method were presented, for example, in [5,6].

2. Software description

Many variables in the world around us are uncertain. Such an uncertainty, for example the length of a human begin, can be mathematically described with a probability distribution. However, related uncertainties are often dependent on each other, like a human being's length with his/her weight. Such dependencies can be described with a multivariate probability distribution or dependence model. When no data are available to determine a suitable model, expert judgment can be utilized to come up with a model for the joint distribution of interest. This is no trivial matter, as dependence models are mathematically complex. Matlatzinca is a program to schematize and quantify a dependence model, specifically the Non-parametric Bayesian Network.

2.1. Description of the GUI

Fig. 1 presents a screenshot of Matlatzinca. It presents an example of a Bayesian Network (BN) for hydrological modeling quantified by one of the seven experts that participated in a recent application for estimating extreme river discharges in the Meuse River [7]

The current version of the GUI consists of 3 main panels:

- The drawing panel. This is where the Directed Acyclic Graph (DAG) that represents the dependence structure of the BN is drawn. This can be done by using the `Add node`, `Add edge` and `Delete selected node/edge` buttons (see Fig. 2 left). Notice that arcs provide information regarding the ordering of parents in the DAG according to the protocol discussed in [3]. That is, which arcs represent conditional rank correlations and the respective conditioning sets.
- The input panel. This panel presents the labeling of `Nodes` (random variables) used in the drawing panel. It also presents the `Edges` that users must quantify through expert judgments. Two options are available for the quantification of arcs: (i) introducing Spearman's conditional rank correlations (`Conditional rank corr.`) or (ii) Non-conditional rank correlations (`non-conditional rank corr.`). Notice that the interval in which non-conditional rank correlations may vary is also provided by the software (`Range non-cond. rank corr.` see Fig. 2 bottom-right). This interval depends on the structure of the DAG and the size of the correlations input by users in the ancestors' arcs. Notice also that the first parent of a node corresponds to a non-conditional rank correlation which may take values in $(-1, 1)$. Conditional rank correlations may also take values in this interval. The interval for non-conditional rank correlations indicates the restrictions imposed by the conditional independence statements embedded in the DAG and the restrictions of the correlation matrix itself. The `Range non-cond. rank corr.` is updated as users introduce values for either conditional or non-conditional rank correlations.
- The correlation matrix panel. The magnitude and direction of the individual correlation coefficients are represented by circles of different diameters that vary according to magnitude and a `Colormap` which varies according to direction and may be chosen by the user. Fig. 2 top-right presents the correlation matrix panel of Matlatzinca for the same case shown in Fig. 1 with a different colormap. The correlation matrix panel also allows to `Plot conditional probabilities` (Fig. 2). This shows $P(F_{X_1}(x_{1,q}) > q | F_{X_2}(x_{2,q}) > q)$ for 3 different user-defined values of percentiles q in the distributions of two random variables X_1 and X_2 as a function of the rank correlation coefficient of the random variables [4] under the Gaussian copula assumption. This with the intention of helping users quantify their models, by showing the relation between rank correlations and conditional probabilities.

The different panes are interactively connected. Clicking a node or edge in the left pane highlights the corresponding rows in the tables on the lower right. A project can be saved and loaded from the `File`-menu. Display properties can be adjusted in the `View`-menu, and the `Export`-menu contains options of exporting the nodes, edges, or correlation matrix to CSV or the clipboard. Finally, the documentation, including a Quick-start description are accessible via the `Help`-menu.

2.2. Code functionality

The back-end of the code is largely based on PyBANSHEE. Only the functionality needed for the computation of the conditional rank correlations was adopted. To make it suitable for a PyQt5 GUI, the functions and variables were grouped into Python classes. Several methods were simplified or restructured to make them easier understandable or faster. Two computational methods that were added to the GUI, but are not present in PyBANSHEE, are:

- The option to enter a non-conditional correlation, and get the resulting rank correlation.
- The limits for this correlation coefficient, to get a valid $(-1, 1)$ conditional rank correlation.

2.3. Software architecture

The program is coded in the Python programming language (version 3.8+). Apart from built-in Python packages, four additional modules were used. These are the well-known Matplotlib, Numpy, and PyQt5, and the lesser known module Pydantic, to provide easy validation, import, and export of the project data structure. The number of modules was deliberately kept small, such that the program can by compiled into a (not too large) stand-alone executable using PyInstaller. This is also the reason for adopting part of the PyBANSHEE code, rather than creating a branch and importing the module; PyBANSHEE has many more dependencies that would need to be included, making the executable significantly larger.

3. Impact

The program was initially designed for a practical application in which a dependence model was elicited with expert judgment. Estimating dependencies is considered more difficult than estimating single-variable uncertainties, partly because of the more complex mathematics of dependence models. With this program, we hope to facilitate to some extent this mathematical obstacle and pave the road for further dependence elicitation applications.

4. Final comments

Matlatzinca is our first GUI built around PyBANSHEE. The main limitations of the current version of Matlatzinca are:

- It does not allow for evaluation of expert performance and combination of multivariate uncertainty quantification as our previous tools for one dimensional uncertainty quantification by experts do [8–10]. Future versions of Matlatzinca will implement the dependence-calibration (or d-cal [11–13]) score already implemented in PyBANSHEE. Other metrics may also be explored.

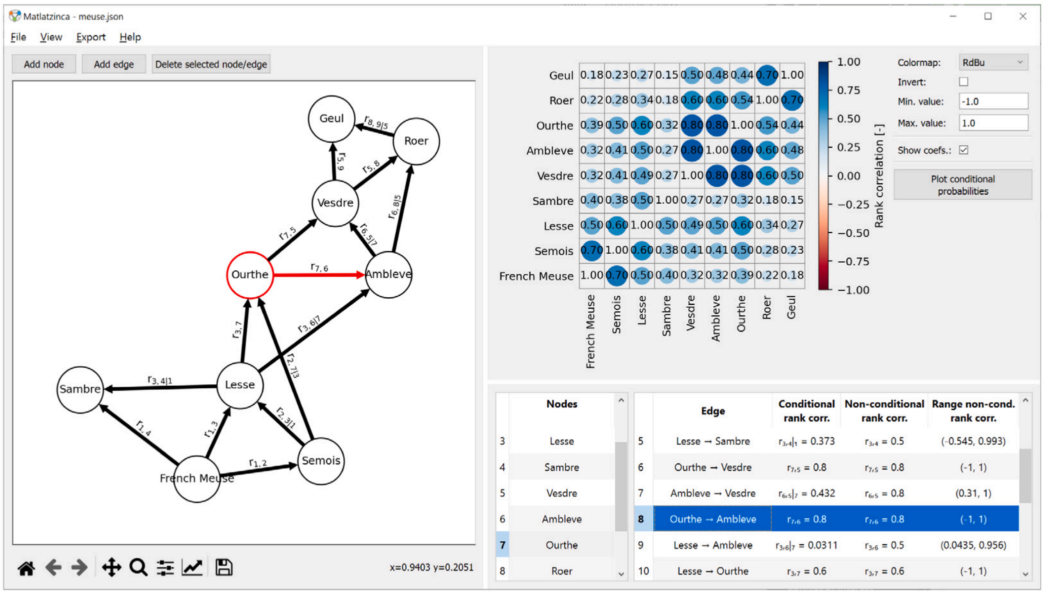

Fig. 1. PyBANSHEE GUI for elicitation of Non-Parametric Bayesian Networks from experts: Matlatzinca. On the left, the drawing panel. On the top-right, the correlation matrix panel. On the bottom-right, the input panel.

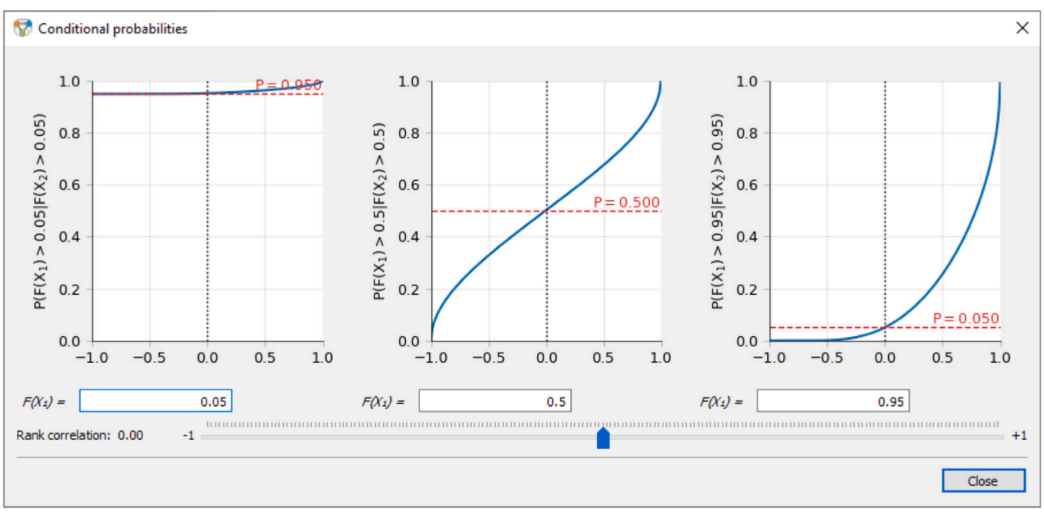

Fig. 2. Conditional probabilities of interest as a function of rank correlation available in the matrix panel of Matlatzinca.

- The current version of Matlatzinca is designed for quantification of NPBNs through expert judgments. Data-based models may be built in PyBANSHEE but not in the present version of Matlatzinca. These functionalities should be implemented in future versions of our GUI.

Despite its limitations, Matlatzinca has already been used in an in-person expert elicitation, during which Matlatzinca proved to be a powerful tool for the quantification of NPBNs through expert judgment.

CRediT authorship contribution statement

Guus Rongen: Conceptualization, Programming, Writing – editing & reviewing. **Oswaldo Morales-Nápoles:** Writing – original draft, Reviewing, Conceptualization, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This development was funded by the TKI project EMU-FD. This research project is funded by Rijkswaterstaat, Deltares, and HKV consultants.

References

[1] Koot P, Mendoza-Lugo MA, Paprotny D, Morales-Nápoles O, Ragno E, Worm DT. Pybanshee version (1.0): A Python implementation of the MATLAB toolbox BANSHEE for non-parametric Bayesian networks with updated features. SoftwareX 2023;21:101279.

[2] Paprotny D, Morales-Nápoles O, Worm DT, Ragno E. BANSHEE–a MATLAB toolbox for non-parametric Bayesian networks. SoftwareX 2020;12:100588. http://dx.doi.org/10.1016/j.softx.2020.100588.

[3] Hanea A, Morales Napoles O, Ababei D. Non-parametric Bayesian networks: Improving theory and reviewing applications. Reliab Eng Syst Saf 2015;144:265–84. http://dx.doi.org/10.1016/j.ress.2015.07.027.

[4] Morales O, Kurowicka D, Roelen A. Eliciting conditional and unconditional rank correlations from conditional probabilities. Reliab Eng Syst Saf 2008;93(5):699–710. http://dx.doi.org/10.1016/j.ress.2007.03.020.

[5] Delgado-Hernández D-J, Morales-Nápoles O, De-León-Escobedo D, Arteaga-Arcos J-C. A continuous Bayesian network for earth dams' risk assessment: An application. Struct Infrastruct Eng 2014. http://dx.doi.org/10.1080/15732479.2012.731416.

[6] Morales-Nápoles O, Delgado-Hernández DJ, De-León-Escobedo D, Arteaga-Arcos JC. A continuous Bayesian network for earth dams' risk assessment: Methodology and quantification. Struct Infrastruct Eng 2014. http://dx.doi.org/10.1080/15732479.2012.757789.

[7] Rongen G, Morales-Nápoles O, Kok M. Using the classical model for structured expert judgment to estimate extreme river discharges: A case study of the meuse river. 2024, [Under review].

[8] Leontaris G, Morales-Nápoles O. ANDURIL: A MATLAB toolbox for analysis and decisions with UnceRtaInty: Learning from expert judgments. SoftwareX 2018;7:313–7. http://dx.doi.org/10.1016/j.softx.2018.07.001.

[9] 't Hart CMP, Leontaris G, Morales-Nápoles O. Update (1.1) to ANDURIL — A MATLAB toolbox for analysis and decisions with UnceRtaInty: Learning from expert judgments: ANDURYL. SoftwareX 2019;10:100295. http://dx.doi.org/10.1016/j.softx.2019.100295.

[10] Rongen G, 't Hart CMP, Leontaris G, Morales-Nápoles O. Update (1.2) to ANDURIL and ANDURYL: Performance improvements and a graphical user interface. SoftwareX 2020;12:100497. http://dx.doi.org/10.1016/j.softx.2020.100497.

[11] Morales-Nápoles, Worm D. Hypothesis testing of multidimensional probability distributions. In: TNO (Netherlands organization for applied scientific research) report. 2013.

[12] Morales-Nápoles O, Hanea AM, Worm DTH. Experimental results about the assessments of conditional rank correlations by experts: Example with air pollution estimates. In: Safety, reliability and risk analysis: beyond the horizon - proceedings of the European safety and reliability conference. 2014, p. 1359–66.

[13] Werner C, Bedford T, Cooke R, Hanea A, Morales-Nápoles O. Expert judgement for dependence in probabilistic modelling: A systematic literature review and future research directions. European J Oper Res 2017;258(3):801–19. http://dx.doi.org/10.1016/j.ejor.2016.10.018.