Environmental Modelling and Software 108 (2018) 61–66

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A comparison of methods for discretizing continuous variables in Bayesian Networks

Tomas Beuzen[a,*], Lucy Marshall[b], Kristen D. Splinter[a]

[a] Water Research Laboratory, School of Civil and Environmental Engineering, UNSW Sydney, NSW, Australia
[b] School of Civil and Environmental Engineering, UNSW Sydney, NSW, Australia

ABSTRACT

Bayesian Networks (BNs) are an increasingly popular method for modelling environmental systems. The discretization of continuous variables is often required to use BNs. There are three main methods of discretization; manual, unsupervised, and supervised. Here, we compare and demonstrate each approach with a BN that predicts coastal erosion. Results reveal that supervised discretization methods produced BNs of the highest average predictive skill (73.8%), followed by manual discretization (69.0%) and unsupervised discretization (64.8%). However, each method has specific advantages that may make them more suitable for particular applications. Manual methods can produce physical meaningful BNs, which is favorable in environmental modelling. Supervised methods can autonomously and optimally discretize variables and may be preferred when predictive skill is a modelling priority. Unsupervised methods are computationally simple and versatile. The optimal discretization scheme should consider both the performance and practicality of the scheme.

1. Introduction

Bayesian Networks (BNs) are probabilistic graphical models that can be used to represent causal systems (Pearl, 1988). BNs have several key features that make them useful for environmental modelling; they can easily handle non-linear systems, have low computational cost, can deal with missing data and data from different sources, explicitly include uncertainties, and have a simple and intuitive graphical structure that is easily understood by non-technical users (Uusitalo, 2007; Chen and Pollino, 2012). As a result, BNs are an increasingly popular method of environmental modelling (Aguilera et al., 2011) and have been used in a variety of applications to date, for example: modelling and supporting decision making in water resource management (Castelletti and Soncini-Sessa, 2007); conducting ecological risk assessments (Pollino et al., 2007) and modelling wildlife habitat and population viability (Marcot et al., 2001); modelling coastal vulnerability to sea level rise (Gutierrez et al., 2011); and integrated modelling of socioeconomic and biophysical processes for natural resource management (Kragt et al., 2011).

Despite their wide applicability to environmental modelling, most commonly used BN software packages and algorithms require discrete data, which is a limitation because environmental systems are often characterized by continuous attributes. This means that discretization of continuous data is often necessary to effectively and efficiently use BNs for environmental modelling. There are three main methods of discretizing continuous data for use in BNs: (1) Manual, in which discretization is specified by an expert user; (2) Supervised, in which the value of the output variable(s) is used to automatically optimize discretization of other variables in the system; and (3) Unsupervised, in which information about the output variables is not available or not used and discretization is based on the distribution of each individual variable (Dougherty et al., 1995). The process of discretizing continuous data for use in a BN can result in a loss of information from the system (Friedman and Goldszmidt, 1996) and can significantly influence BN model performance (Fienen and Plant, 2015; Nojavan et al., 2017). Despite this, the impacts of different methods of discretization on BN performance have not been well discussed in the literature (Death et al., 2015; Nojavan et al., 2017), and discretization has been an overlooked and undocumented process in many environmental BN applications to date. For example, in a recent review of BN applications in environmental modelling, Aguilera et al. (2011) noted that approximately 50% of studies that discretized continuous data for use in a BN did not discuss the discretization method used.

To begin to address this issue, a recent study by Nojavan et al. (2017) compared different algorithms of unsupervised discretization and found that while no one algorithm consistently outperformed others, the method of discretization could influence model performance. The study by Nojavan et al. (2017) did not evaluate manual or supervised discretization. Supervised discretization algorithms are of particular interest here because, unlike manual and unsupervised methods, they remain largely unused in environmental BN applications. This is surprising, as supervised discretization algorithms are an

* Corresponding author.
E-mail address: t.beuzen@unsw.edu.au (T. Beuzen).

https://doi.org/10.1016/j.envsoft.2018.07.007
Received 20 December 2017; Received in revised form 14 June 2018; Accepted 17 July 2018
Available online 17 July 2018
1364-8152/

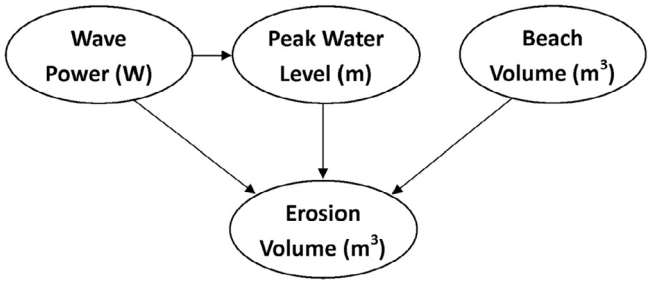

Fig. 1. The Bayesian Network modelling coastal erosion used in this study to compare discretization methods.

efficient method of autonomously discretizing continuous data to maximize predictability of the output variable and have been shown to produce more predictive models than unsupervised discretization algorithms (e.g., Fayyad and Irani, 1993; Dougherty et al., 1995).

The present study aims to extend the work of Nojavan et al. (2017) by comparing the effect of manual, unsupervised and supervised discretization on the performance of an environmental BN used to predict coastal erosion from storms. Each method of discretization is evaluated and practical guidelines for their use in future BN studies are proposed.

2. Methodology

2.1. Bayesian Networks

A BN is a graphical representation of the joint probability distribution of a system. The structure of a BN is formally known as a directed acyclic graph and is composed of nodes representing variables in the system and arcs representing causality between nodes (e.g., Fig. 1). Conditional dependencies amongst variables in the system are quantified in conditional probability tables (CPTs). To make predictions of the system, these conditional dependencies and the prior distribution of variables are used with Bayes' Theorem:

$$P(R_i|O_j) = \frac{P(O_j|R_i) * P(R_i)}{P(O_j)} \quad (1)$$

where $P(R_i|O_j)$ is the predicted or posterior probability of a response (R_i) conditioned on the observation(s) (O_j), $P(O_j|R_i)$ is the likelihood function, and $P(R_i)$, and $P(O_j)$ are the prior probabilities of the response and observation(s), respectively. Discrete data are typically required to learn the CPTs describing a system and perform inference with them. The discretization scheme of each variable determines its prior distribution and the conditional dependencies learnt by the BN and is therefore a key factor in BN model performance. For a thorough introduction into BNs, the reader is referred to Pearl (1988) and Charniak (1991).

2.2. Discretization methods

The main methods to discretizing continuous variables in BNs can be classified as manual, supervised and unsupervised (Chen and Pollino, 2012). Within each method, multiple algorithms have been proposed for developing effective BNs across a range of modelling applications.

2.2.1. Manual discretization

Manual discretization (also referred to as expert discretization) involves a user manually selecting discretization thresholds based on physical meaningfulness, theoretical knowledge, or their expert interpretation of the problem domain (Chen and Pollino, 2012). Manual discretization can be aided through the use of tools such as histograms or regression trees to better understand thresholds present in the data. In a review of 128 published papers on BN applications in environmental modelling, Aguilera et al. (2011) noted that manual discretization was the most common method used in studies that required discretization of continuous data. This method is often favored because it allows continuous variables to be discretized at intervals interpretable and relevant to the data or model objectives (Uusitalo, 2007; Chen and Pollino, 2012), and no discretization algorithm or additional computation is required.

2.2.2. Unsupervised discretization

Unsupervised discretization is a method of discretizing continuous data based on the intrinsic data distribution of each individual variable. It is commonly used to discretize continuous variables for BN applications when manual discretization is not available due to the absence of theoretical or expert knowledge of the data or system being modelled (Aguilera et al., 2011). Unsupervised discretization is popular because it is computationally simple and objective. The equal-width (EW) and equal-frequency (EF) algorithms tested in this study are two of the most commonly used unsupervised discretization algorithms in environmental applications (Aguilera et al., 2011; Chen and Pollino, 2012). EW discretization divides continuous data into a predefined number of intervals of equal width. This method can perform well with approximately uniform continuous distributions, but generates inappropriate intervals of imbalanced probabilities when data is highly skewed or contains outliers (Chen and Pollino, 2012). EF discretization divides continuous data into a predefined number of intervals of equal frequency. EF discretization generates a uniform (non-informative) distribution of the continuous data which is useful for capturing the 'modes' of the distribution (Nojavan et al., 2017), but it can hide outliers in the data (which are often of interest in environmental systems) and in cases where there is a high frequency of the same value, that value may be forced to split into different intervals (Chen and Pollino, 2012). Using unsupervised discretization algorithms like EF binning that produce non-informative prior distributions is often favored because the resulting BN is purely driven by the data (i.e., states in the BN are not predisposed by a prior). To apply EW or EF discretization, the number of intervals to partition the data into must be specified. Most BN applications typically use between 2 and 10 intervals (Uusitalo, 2007).

2.2.3. Supervised discretization

Supervised discretization is an informative method of discretization that utilizes the state of the output variable to inform and optimize the discretization of each individual input variable. Supervised discretization algorithms are frequently used in the computer science literature and have been shown to outperform unsupervised discretization when using BNs on a variety of datasets (Dougherty et al., 1995). Despite this, they remain largely absent from the environmental BN literature; due in part to a lack of capability in the BN software packages commonly used for environmental BN modelling (such as Netica, (Norsys Software Corporation, 2017)), as well as remaining knowledge gaps between environmental modelling and the broader computer science and machine learning literature. Like unsupervised discretization, a drawback of supervised discretization algorithms is that the thresholds they produce are often physically meaningless. In addition, supervised algorithms may produce potentially spurious discretization thresholds that are fit to noise in the data rather than thresholds that increase BN predictive skill. Supervised discretization also requires a discrete output variable to inform the discretization of the continuous input variables. This means that, if the output variable is continuous, *a-priori* knowledge, assumptions or an unsupervised discretization method would be required to discretize it before the input variables can be discretized using a supervised method. While there are many supervised discretization algorithms available, the Fayyad & Irani (F&I) (Fayyad and Irani, 1993) and Kononenko (KO) (Kononenko, 1995) algorithms are well-tested and are available in commonly used software packages such as R and Python. Both algorithms are based on entropy minimization