Environmental Modelling & Software 79 (2016) 311–321

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Improving scenario discovery for handling heterogeneous uncertainties and multinomial classified outcomes☆

Jan H. Kwakkel*, Marc Jaxa-Rozen

Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628 BX, Delft, The Netherlands

ARTICLE INFO

Article history:
Received 29 April 2015
Received in revised form
10 November 2015
Accepted 24 November 2015
Available online 30 December 2015

Keywords:
Scenario discovery
Deep uncertainty
Robust decision making

ABSTRACT

Scenario discovery is a novel model-based approach to scenario development in the presence of deep uncertainty. Scenario discovery frequently relies on the Patient Rule Induction Method (PRIM). PRIM identifies regions in the model input space that are highly predictive of producing model outcomes that are of interest. To identify these, PRIM uses a lenient hill climbing optimization procedure. PRIM struggles when confronted with cases where the uncertain factors are a mix of data types, and can be used only for binary classifications. We compare two more lenient objective functions which both address the first problem, and an alternative objective function using Gini impurity which addresses the second problem. We assess the efficacy of the modification using previously published cases. Both modifications are effective. The more lenient objective functions produce better descriptions of the data, while the Gini impurity objective function allows PRIM to be used when handling multinomial classified data.

Software

This paper makes use of the Exploratory Modeling Workbench, available via https://github.com/quaquel/EMAworkbench. Section 3.4 relies on extensions to classes available in the workbench. These extensions are provided as supplementary material. The detailed code with rudimentary documentation is provided in the form of 3 pdf representations of the underlying IPython notebooks.

1. Introduction

Scenario discovery is a relatively novel approach aimed at addressing the challenges of characterizing and communicating deep uncertainty associated with simulation models (Dalal et al., 2013). The basic idea is that the consequences of the various deep uncertainties encountered in a model-based decision support exercise are systematically explored through series of computational experiments (Bankes et al., 2013). These computational experiments are designed to exhaustively sample the space spanned by the various deeply uncertain factors. The results of the set of computational experiments are analyzed to identify regions in the uncertainty space that are of interest (Bryant and Lempert, 2010; Kwakkel et al., 2013). These identified regions can subsequently be communicated as scenarios.

A motivation for the use of scenario discovery is that the available literature on evaluating scenario studies has found that scenario development is difficult if the involved actors have diverging interests and worldviews (Bryant and Lempert, 2010; van't Klooster and van Asselt, 2006). Rather than trying to achieve consensus or facilitate a process of joint sense-making to resolve the differences between worldviews, scenario discovery aims at making transparent which uncertain factors actually make a difference for the decision problem at hand. Another shortcoming identified in the evaluative literature is that scenario development processes have a tendency to overlook surprising developments and discontinuities (Derbyshire and Wright, 2014; van Notten et al., 2005). This might be at least partly due to the fact that many scenario approaches move from a large set of relevant uncertain factors to a smaller set of drivers or "megatrends". In this dimensionality reduction, interesting plausible combinations of uncertain developments are lost. In contrast, scenario discovery first systematically explores the

☆ This paper is part of a thematic issue on Innovative Techniques for Quantitative Scenarios in Energy and Environmental Research.

* Corresponding author.
E-mail addresses: j.h.kwakkel@tudelft.nl (J.H. Kwakkel), M.Jaxa-Rozen@tudelft.nl (M. Jaxa-Rozen).

http://dx.doi.org/10.1016/j.envsoft.2015.11.020

consequences of all the relevant factors, and only then performs a dimensionality reduction in light of the resulting outcomes – thus potentially identifying surprising results that would have been missed with traditional scenario logic approaches.

Although scenario discovery can be applied on its own (Gerst et al., 2013; Kwakkel et al., 2013; Rozenberg et al., 2013), it is also a key step in Robust Decision Making (RDM) (Dalal et al., 2013; Hamarat et al., 2013; Lempert and Collins, 2007; Lempert et al., 2006). RDM aims at supporting the design of robust policies. That is, policies that perform satisfactorily across a very large ensemble of future worlds. In this context, scenario discovery is used to identify the combination of uncertainties under which a candidate policy performs poorly, allowing for the iterative improvement of this policy. This particular use case of scenario discovery suggests that it could be used also in other planning approaches that design plans based on an analysis of the conditions under which a plan fails to meet its goals (Walker et al., 2013).

Currently, the main statistical rule induction algorithm that is used for scenario discovery is the Patient Rule Induction Method (PRIM) (Friedman and Fisher, 1999), although other algorithms such as Classification and Regression Trees (CART) (Breiman et al., 1984) are sometimes used (Gerst et al., 2013; Lempert et al., 2008). PRIM aims at finding combinations of values for the uncertain input variables that result in similar characteristic values for the outcome variables. Specifically, PRIM seeks a set of subspaces of the uncertainty space within which the value of a single output variable is considerably different from its average value over the entire domain. PRIM describes these subspaces in the form of hyper rectangular boxes of the uncertainty space. To identify these subspaces, PRIM uses a lenient or patient, as opposed to greedy, hill climbing optimization procedure. In the context of scenario discovery, the outcome variable is typically a binary variable denoting whether a given set of inputs is of interest or not. The hyper rectangular boxes identified by PRIM are not always the best description of the combination of input variables that produces similar characteristic values for the outcome variables. Sometimes, these characteristic values are grouped along another axes than the set of uncertain input variables. Preprocessing the data using principal components analysis can help to identify such axes and rotate the data (Dalal et al., 2013). The most frequently employed implementation of PRIM that is being used for scenario discovery is the one provided by Bryant in the scenario discovery toolkit, written in R (Bryant, 2014). A Python implementation of PRIM, including support for the PCA preprocessing, is available as part of the Exploratory Modeling Workbench (Kwakkel and Pruyt, 2015).

There are two problems related to PRIM that are addressed in this paper. First, although originally presented as a regression based rule induction algorithm, in the context of scenario discovery PRIM is typically used on a binary classification of the data. In contrast to e.g. CART, PRIM cannot be used directly for handling the situation where the output data is classified using more than two classes (Gerst et al., 2013; Rozenberg et al., 2013). Second, when the uncertain factors are represented by integers or categories, the lenient hill climbing optimization procedure used in PRIM needs to account for this. Friedman and Fisher (1999) offer several suggestions for adapting the objective function used by PRIM to account for this. Both the scenario toolkit, and the Python implementation include these modified objective functions. However, to date the efficacy of these alternative objective functions has not been systematically evaluated in the context of scenario discovery. We address both problems in this paper because their solutions are both closely related with the lenient hill climbing optimization approach used in PRIM.

To address these two problems, we first outline in Section 2 in more detail the PRIM algorithm. We will discuss the suggestions of Friedman and Fisher (1999) for handling integer and categorical data in evaluating the next possible steps of the algorithm. To address the problem of multinomial classified data, we draw on the way in which CART handles this and show how by adapting the objective function used by PRIM, it can be made applicable also to problems where the data is classified using multinomial classification. The resulting modifications to PRIM do not affect the efficacy of preprocessing steps such as employed in PCA-PRIM (Dalal et al., 2013). We provide an open source implementation in Python for this modified version of PRIM.

In Section 3, we assess the efficacy of alternative ways of accounting for categorical and discrete data in the objective function used by PRIM. In particular, we apply it to the same data as used in the original paper of Bryant and Lempert (2010), the case study of Rozenberg et al. (2013), and the case used by Hamarat et al. (2014). The first case covers continuous uncertain factors, the second case covers discrete uncertain factors, and the third case has continuous, discrete, and categorical uncertain factors. In Section 4, we explore the objective function for handling multinomial classified data and compare it to both CART and a sequential PRIM approach. For this we use the case study of Rozenberg et al. (2013). A discussion of the results is presented in Section 5 and the conclusions are presented in Section 6.

2. Method

2.1. PRIM

Fig. 1 offers a visual explanation of the PRIM algorithm. In the top left corner we see the dataset. The dataset consists of 110 computational experiments, 30 of which are of interest. Each experiment is described by two variables. The first variable, U_1, is a categorical variable and the possible values are $\{a,b,c\}$. The second variable, U_2, is a continuous variable ranging between 0 and 2. Together, U_1 and U_2 span the uncertainty space. We use PRIM to find an orthogonal subspace, or box, within the uncertainty space that has a high concentration of experiments of interest.

PRIM starts with an initial box B_1 that covers all of the data. Next, the size of this box is recursively reduced. Reducing the size of the box is done by removing a small slice of data along one of the dimensions. To find the best slice of data to remove, the algorithm first enumerates all possible slices, b_j, that can be removed, and next uses an objective function to determine the best possible slice to remove. This results in a new box B_l. The series of boxes resulting from this recursive peeling is also known as the peeling trajectory.

How does PRIM enumerate all the possible slices of data that can be removed? PRIM will only remove data along a single dimension. So, for each dimension, PRIM enumerates all the possibilities. The exact possibilities depend on the data type of the dimension. In the example given in Fig. 1, we have two different data types. U_1 is a categorical variable. In this case, PRIM will consider the removal of each of the individual categories.[1] In our example, this means that there are three alternative slices of data that PRIM considers for removal for this dimension. U_2 is a continuous variable. In this case, PRIM will consider the removal of a small slice from the top and a small slice from the bottom. Continuous variables will thus contribute two alternative slices of data that PRIM will consider for removal. The same is true for integer data.

[1] The Python implementation of PRIM, available as part of the EMA workbench, follows the outlined approach for the generation of candidate boxes. Categorical variables are not presently supported by the R implementation in the Scenario discovery toolkit. For more details on the consequences of this, see the supplementary material.