# Behavior-Based Scenario Discovery

## Induction of decision-relevant input subspaces from nonlinear model outputs using time series clustering

by

Patrick Steinmann

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday August 31, 2018 at 09:00 AM.

| | | |
|---|---|---|
| Student number: | 4623991 | |
| Project duration: | February 12, 2018 – August 31, 2018 | |
| Thesis committee: | Prof.dr.ir. A. Verbraeck, | TU Delft, chair |
| | Dr. J. H. Kwakkel, | TU Delft, first supervisor |
| | Dr. E. J. L. Chappin, | TU Delft, second supervisor |

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Associated code and models are available at `http://github.com/steipatr/`.

TU Delft

# Executive Summary

Many societal, environmental and technological challenges can be characterized as wicked problems by virtue of being difficult to understand, define and solve. Examples include sustainable management and consumption of resources, resilient technical infrastructure or curbing plastic pollution of the oceans. One method of tackling such wicked problems is the use of computer-aided modelling and simulation. Model-based decision support is a growing discipline involving the use of computer models of complex systems to explore, understand and manage them. A core concept in model-based decision support is scenario discovery.

In scenario discovery, a model's inputs and outputs are related to understand under which conditions policy-relevant outputs may occur. In a first step, a diverse set of inputs is used to generate a variety of outputs. In a second step, the subset of decision-relevant outputs is identified among the outputs through some external criterion, such as a threshold value. Finally, the inputs which generated those outputs of interest are identified, and a generative rule set is induced which usefully predicts under which conditions an input will generate an output meeting the external criterion. This rule set bounds an input subspace of interest, from which (most of) the outputs of interest originate.

While scenario discovery performs adequately for quasi-linear and simple models, it is not well suited to behaviorally complex, nonlinear models. This is both because external criteria are hard to define for complex model behaviors, and also because there are often significant interactions and dependencies between model inputs, which current rule induction algorithms have trouble identifying. Furthermore, the application of external criteria introduces subjective judgement and potential bias into the analysis process. Finally, unsophisticated external criteria can conflate distinct policy vulnerabilities in the model's input space, impeding effective strategy design.

To alleviate these issues, this work proposes behavior-based scenario discovery, a novel approach to identifying decision-relevant input subspaces. Rather than applying an external criterion to identify outputs of interest, intrinsic model behaviors are identified and grouped using time series clustering. This removes subjective judgement from the analysis process, and can separate distinct policy vulnerabilities.

Methodologically, behavior-based scenario discovery begins with dense random sampling of the model input space. These inputs are then used to generate time series outputs for a model metric of interest. Using cluster validity indices and a range of time series clustering algorithms, likely cluster counts are identified. The set of output time series is then partitioned into an according number of clusters. For each cluster, the generative rule set is induced from the inputs associated with the output cluster's constituents, creating a set of input subspaces individually associated with distinct output behavior clusters.

In this thesis, the approach described above is applied to three nonlinear system dynamics models. The first test case describes an insect population. Using behavior-based scenario discovery, the analytical bifurcation regions inherent in the model can be reasonably approximated, even though no analysis of the model itself is performed. For a second test case, concerning an unusual oscillating chemical reaction, the bifurcation behavior of the model can similarly be identified using rule induction performed on inputs associated with clustered time series outputs. These two simple test cases indicate the method holds promise, and also enable the comparison of a variety of different time series clustering methods regarding speed, implementation maturity and bifurcation region separation. A third, highly complex test case is therefore analysed using the most promising clustering methods. While the analytical bifurcation regions cannot be identified due to the inherent complexity, it is found that behavior-based scenario discovery can still identify predictive input subspaces associated with distinct model output behaviors. This indicates that behavior-based scenario discovery can be a useful addition to the exploratory modelling and analysis toolbox, especially where nonlinear models are concerned.

The re-characterization of scenario discovery as a search for multiple behaviorally distinct input subspaces rather than a single subspace of interest introduces a variety of new criteria by which the quality of such subspaces can be assessed. While each subspace can individually be scrutinized using established local criteria such as coverage, density or interpretability, the relations between subspaces and input space itself can now also be quantified using global criteria such as subspace separability, global coverage, population and validity. The distinction between local and global criteria also requires a revision of established rule induction methods, which is discussed conceptually.

During research on this topic, three tangential insights appeared. Firstly, both exploratory modelling literature and practice do not clearly distinguish between reducible (often parametric) and irreducible (often structural) uncertainties. This is aggravated by the established practice of proxying structural uncertainties through parametric uncertainties. Drawing a clear distinction between the two would not only improve the policy relevance of the rule sets induced with scenario discovery, but also enable new deep uncertainty-specific model analysis methods such as structural uncertainty sensitivity analysis. Secondly, time series clustering could also be used to communicate distinct model behaviors by identifying and visualizing cluster centroids. These centroids are the most representative members of each cluster, and therefore together best represent the possible behaviors of the model over time. Thirdly, some current robust decision making methods require the provision of a set of reference scenarios against which policy designs can be evaluated. There is currently no established method for identifying these scenarios, so analysts must use their judgement and experience. Partitional time series clustering with centroids provides a deterministic and objective method of identifying the most dissimilar (and therefore representative) outputs of a model.

In short, this thesis proposes and validates a novel approach to scenario discovery using intrinsic model behavior rather than extrinsic criteria. Furthermore, conceptual implications of searching for multiple behaviorally distinct input subspaces, rather than a single subspace of interest, are explored. Additionally, a shortcoming in the representation of deep uncertainty in exploratory modelling is identified. Finally, two potential applications of partitional time series clustering and the resulting centroids are discussed. While further effort is needed in all described lines of research, the usefulness of time series clustering in exploratory modelling in general, and scenario discovery in particular, is amply demonstrated.

In the interest of open science, all relevant code for application of the described methods is provided through a public online repository.

# Acknowledgements

My gratitude goes to my committee - Prof.dr.ir. Alexander Verbraeck, Dr. Jan Kwakkel, and Dr. Emile Chappin - for their support, time and patience. I would also like to thank Dr. Erik Pruyt for being himself. Finally, I am grateful to Willem Auping for sharing his data and models on future energy scenarios, without which my most interesting analyses wouldn't have been possible.

*Patrick Steinmann*
*Delft, August 2018*

# Contents

# 1 Research Problem

## 1.1. Modelling for Decision Support

Many significant societal challenges can be characterized as wicked problems (Rittel and Webber, 1973). Policy makers and planners confronted with such wicked problems often turn to models in order to understand the dilemmas involved, and to design corresponding policies. As computer power has increased, the traditional application of qualitative models has been complemented with quantitative, computer-based decision support methods (Parker, Srinivasan, Lempert, and Berry, 2015). Examples of such methods include Robust Decision Making (Groves and Lempert, 2007), Many Objective Robust Decision Making (Kasprzyk, Nataraj, Reed, and Lempert, 2013), Information Gap Modeling (Hipel and Ben-Haim, 1999), Decision Scaling (Brown, Ghile, Laverty, and Li, 2012), and Dynamic Adaptive Policy Pathways (Haasnoot, Kwakkel, Walker, and ter Maat, 2013). Such methods can rapidly generate and analyze wide ranges of possible futures, how they occur, and how to manage them (Kwakkel, 2017), (Davis, Bankes, and Egner, 2007).

### 1.1.1. Deep Uncertainty

Computer-based modelling and analysis tools are especially useful in cases where significant uncertainty about the system in question exists (Bankes, 1993). This inherent and multidimensional uncertainty has been conceptualized as ”deep uncertainty” by Lempert, Popper, and Bankes (2003):

> ... problems requiring decisionmaking under conditions of deep uncertainty — that is, where analysts do not know, or the parties to a decision cannot agree on, (1) the appropriate conceptual models that describe the relationships among the key driving forces that will shape the long-term future, (2) the probability distributions used to represent uncertainty about key variables and parameters in the mathematical representations of these conceptual models, and/or (3) how to value the desirability of alternative outcomes.

### 1.1.2. Exploratory Modelling and Analysis

In the interest of modelling and analyzing deeply uncertain systems for decision support (sometimes referred to as Decision Making Under Deep Uncertainty (Kwakkel, Haasnoot, and Walker, 2016)), a wide range of computer-based methods has been developed, ranging from model exploration (Halim, Kwakkel, and Tavasszy, 2016) and analysis (Cariboni, Gatelli, Liska, and Saltelli, 2007) to iterative policy design algorithms (Kasprzyk et al., 2013). A selection of these methods has been incorporated into the Exploratory Modelling and Analysis Workbench (Kwakkel, 2017), which provides an integrated platform for connecting data, models, analysis tools, and visualizations, while also structuring the modelling and analysis process into functionally distinct steps, for which a variety of different methods can be chosen based on the particularities of the problem at hand. The EMA Workbench builds on the XLRM model framework (Lempert et al., 2003), which structures system models as combinations of externalities *X*, policy levers *L*, causal system relations *R* and metrics of interest *M*. This allows such models to be described as mathematical functions:

$$M = f_R(X, L)$$

where uncertainties and levers *X,L* are inputs, metrics *M* are outputs, and causal relations *R* represent the model or function itself. A third implied input is time - it is generally understood that such models are executed over some time base. This representation allows the application of a wide variety of mathematical methods to questions of policy and is the key to model-based decision support (Kwakkel, 2017). In this context, inputs generally take the form of scalars, while outputs are generally time series (a sequence of scalars spaced evenly over time) or scalars. The causal relations - the function itself - can be expressed through a variety of modelling paradigms including system dynamics, agent-based or discrete event models.

### 1.1.3. Scenario Discovery

Two keystones of exploratory modelling and analysis are the related concepts of sensitivity analysis and scenario discovery. Both aim to reason about the connection between model inputs and outputs, though in different directions. Where sensitivity analysis examines the variance in outputs in relation to the variance in inputs in order to determine the inputs with disproportionate effect on the outputs (Saltelli and Annoni, 2010), scenario discovery looks for the inputs which generate specific decision-relevant subsets of outputs (Bryant and Lempert, 2010). Both methods treat the model as a black box (Wiener, 1961), considering only inputs and corresponding outputs without analyzing model structure.

Scenario discovery has been used with success in a variety of applications including logistics (Halim et al., 2016), environmental studies (Greeven, Kraan, Chappin, and Kwakkel, 2016) and resource scarcity (Kwakkel and Pruyt, 2013), also because it is relatable to established scenario-based planning techniques from business and operations management (Guivarch, Lempert, and Trutnevyte, 2017).

## 1.2. Research Gap

Despite the widespread use of scenario discovery, there are some lingering questions about its suitability for analysis of complex systems.

Firstly, complex systems often show a variety of dynamic behaviors (Yücel and Barlas, 2011). Such transitions in model behavior conceptually align well with the notions of nonlinear and chaotic behavior in mathematics - a minor change in input variables causes a disproportionately (to use a more policy-relevant term: unexpectedly) large change in output (Strogatz, 1994). These behaviors are generally created by interacting feedback loops, which are a staple of complex systems modelling. However, foundational scenario discovery literature downplays the necessity of sampling input parameter spaces with high granularity/resolution (Davis et al., 2007), claiming that model behavior is unlikely to vary significantly (in mathematical terms: nonlinearly) across the input space:

> We also expect that increasing granularity or resolution along any given dimension will not be particularly worthwhile—unless, for some reason, the landscape is rough in spots, with the output under study varying rapidly with the variables being considered. Thus, when exploring a given uncertainty, we expect that looking at endpoint values, or endpoint values and a few intermediate points, will ordinarily be adequate for exploration.

This reasoning stands in contrast to more recent works on scenarios generated by complex system models (Gerst, Wang, and Borsuk, 2013; Haasnoot, Middelkoop, Offermans, van Beek, and van Deursen, 2012; Kwakkel and Pruyt, 2013) which discuss nonlinear and dissimilar model dynamics over time, and how these may be dealt with.

A second point of contention is the method by which decision-relevant model outputs are selected. The current scenario discovery practice is to apply an external statistical criterion to model outputs in order to classify them as relevant or irrelevant. This criterion can be as simple as a threshold value at a specific point in time (e.g. Halim et al. (2016)). However, while such a criterion may be easy to conceptualize and communicate to a stakeholder, this simplification may confound fundamentally different model behaviors based on their states at a specific point in the model run time. These different behaviors may be sensitive to different model inputs, thus requiring individual tipping points (Kwadijk, Haasnoot, Mulder, Hoogvliet, Jeuken, van der Krogt, van Oostrom, Schelfhout, van Velzen, van Waveren, and de Wit, 2010) and adaptive policy measures (Haasnoot et al., 2013). However, if they are lumped together through poorly informed or generic external criteria, overall policy performance may degrade, as different policy vulnerabilities can no longer be specifically targeted. Consider a model which generates two outputs: a constant value over time, and a sine curve oscillating about that constant value. If a classification criterion considers the mean, the two outputs may both be considered equally interesting (or not), but if the criterion considers maximum value, the two outputs are suddenly dissimilar - and if the criterion checks the output value at a specific time step, the classification varies over time. More recent works consider model time series outputs as developments over time or "transient scenarios" (Kwakkel et al., 2016) rather than world states at a particular point in time. One method of identifying such transient scenarios is clustering a model's time series outputs into behaviorally comparable subsets, and then using those subsets to identify decision-relevant outputs. This was first described by Kwakkel and Pruyt (2013) for a single clustering method based on atomic behavior. Other clustering methods may provide different insights into model dynamics.

A final drawback is that conventional scenario discovery assumes that all model outputs for a given characteristic originate from a single orthogonal input subspace (Bryant and Lempert, 2010). However, the assumption of orthogonal input subspaces does not hold true for every model. This may reduce the effectiveness of policy measures identified using scenario discovery, as policy measures might have negative effects outside their intended scope, or fail to influence all potential future states which would require interventions, as described by Dalal, Han, Lempert, Jaycocks, and Hackbarth (2013):

> However, its ability to successfully describe scenarios that illuminate a policy's vulnerabilities is often limited because the regions are often not well described by a hyper-rectangle in the simulation model's input space. In many situations, a trian- gular scenario or one that lies along some other axes than those of the model inputs may best illuminate a policy's vulnerabilities. PRIM-based scenario discovery cannot easily describe such shapes.

In summary, the current state of the art in scenario discovery lacks consideration for the dynamics inherent to complex system models in all three phases of the scenario discovery process - in generating outputs through input uncertainty space sampling, in selecting decision-relevant outputs, and in determining the inputs subspace(s) generating the decision-relevant outputs. These shortcomings are unspecifically acknowledged by Lempert, Groves, Popper, and Bankes (2006) in the footnotes on the chosen scenario discovery method, called PRIM:

> PRIM does not guarantee the ability to find such meaningful clusters. Although clearly important, RDM [Robust Decision Making, an iterative policy design process] can be conducted without them as was done in LPB [Lempert/Popper/Bankes, the authors referring to their previous joint works]. PRIM, an algorithm developed for other purposes, was used here for RDM without significant modification. Further research may provide additional or more-effective algorithms for generating RDM clusters for a wide variety of decision problems.

In established scenario discovery practice, the goal is to identify a set of decision-relevant outputs, which in turn relates to a single input subspace of interest within the entire input space. The volume of the subspace of interest $V_S$ is significantly smaller than the volume of the input space $V_I$:

$$V_S << V_I$$

However, the time series clustering approach described above covers the entire output space, and therefore also the entire input space. Thus, the volumes of the input subspaces across all $k$ clusters will sum to the volume of the input space:

$$\sum_{n=1}^{k} V_{s,n} = V_I$$

This represents a fundamental shift in how scenario discovery is conducted. The implications of this new approach are unclear and should be explored.

Overall, a clear research gap emerges between the lacklustre treatment of complex model dynamics in all phases of scenario discovery, and the need to understand these dynamics and their causes to better support reasoning and decision making about the affected systems. A second research gap stems from the re-characterization of the model input space as the sum of input subspaces aligned with distinct model behaviors, rather than merely containing a subspace of interest, and how this might affect the process of scenario discovery.

## 1.3. Research Questions

The main research question summarizes the research gaps identified for this thesis:

> (M) In complex system models, how can the input subspaces associated with decision-relevant system behaviors over time be found?

I have described scenario discovery as a three-step process, and shown how all three phases lack recognition of model dynamics. The first step (output generation) is readily addressed through literature review (Islam and Pruyt, 2016; Pruyt, Logtens, and Gijsbers, 2011; Yücel and Barlas, 2011). Therefore, I will not consider this step any further. However, the latter two steps (identification of decision-relevant outputs and rule induction) are worth investigating. Once a variety of model outputs over time has been generated using adaptive or high-resolution sampling, the outputs may be processed using time series clustering to find subsets of similar outputs, thus separating the outputs by policy vulnerability:

> (1) How can time series clustering be applied to the outputs of simulation models to partition the outputs into subsets with similar dynamics?

Once the output clusters have been identified, the generative input subspaces for each cluster can be induced, completing the scenario discovery process.

> (2) How can the generative input subspaces for multiple subsets of model outputs be induced?

These two research questions address the first research gap identified earlier. The second research gap - the implications of searching for all behaviorally distinct input subspaces rather than a single input subspace of interest - warrants a third research question:

> (3) How does the transition from searching for a single input subspace of interest to searching for multiple distinct subspaces conceptually affect the scenario discovery process?

## 1.4. Research Flow

This thesis consists of seven chapters. In this first chapter, I have introduced the research problem and the research questions. Subsequently, I present my literature review, which focuses on exploratory modelling, nonlinear dynamics and time series clustering. The following two chapters present two simple test cases for time series clustering in nonlinear model outcomes - the spruce budworms model and the Brusselator model. From these test cases, I identify the most suitable time series clustering methods, and apply them to a highly complex and nonlinear model concerning future oil price developments. These three chapters relate to research questions 1 and 2. Subsequently, I discuss some implications of the shift from searching for an input subspace of interest towards behaviorally distinct subspaces, and explore how scenario discovery might be re-characterized based on this shift. A conclusive chapter will revisit the research questions, critically discuss the demonstrated methods, reflect on the performed work, and mention future research avenues.

## 1.5. Research Relevance

This research effort is generally relevant to the field of model-based decision support, especially where nonlinear models are concerned. As it focuses on methodological development, the resulting artefacts may be applicable in a variety of future research efforts and applications. Applications include open model exploration and policy design processes, especially where models exhibiting a wide variety of dynamic behavior are used.

## 1.6. Chapter Summary

Model-based decision support is a collective term for a variety of approaches to using computer models to reason about and manage complex systems under deep uncertainty. One such approach is scenario discovery, which is used to characterize the relations between the out- and inputs of complex system models, and identify under which circumstances decision-relevant outputs may occur. However, the current scenario discovery state of the art lacks consideration for nonlinear dynamics in complex models. As these dynamics may be decision-relevant, this indicates a research gap in the field of exploratory modelling. This thesis will address that research gap.

# 2
# Literature Review

## 2.1. Scenario Discovery

Scenario discovery is an algorithmic method of finding and understanding policy-relevant outputs of complex system models, and was first proposed by Groves and Lempert (2007). It follows a three-step process:

1. Generate a variety of model outputs by repeatedly running a simulation model with a wide variety of inputs
2. Identify the decision-relevant outputs as a subset of the entirety of generated outputs
3. Using the inputs associated with each decision-relevant output, determine the input subspace of interest which generates the outputs of interest.

For example, a policy analyst might create a model of a bicycle sharing service's fleet utilization. He would now like to know under which conditions the fleet will be utilized less than 25% - perhaps this is the break-even point for the company. Scenario discovery might reveal that the conditions for such under-utilization are an average bike age of over 2 years, a rental fee of over 1.4 Euro/h and a peak daytime temperature above 30°C. These insights could then be used by the fleet manager to ensure this ”failure scenario” is avoided, insofar possible - for example by reducing rental fees on hot days, or continually renewing the fleet.

### 2.1.1. Scenarios

The phrase ”scenario discovery” implies that simulation models generate scenarios which can be discovered. Confusingly, the term ”scenario” is not used consistently across the scenario discovery literature. Groves and Lempert (2007) and Lempert, Bryant, and Bankes (2008) use the term to represent the entire input subspace associated with a set of outputs which meet some condition. This can be understood as a satisficing approach - ”every output which fulfills my conditions is grouped into one scenario, regardless of how well it fulfills these conditions”. The focus lies on the input subspace which generates the conditions-fulfilling outputs. In contrast, Kwakkel and Pruyt (2013) and Halim et al. (2016) understand every single output of a simulation model experiment (that is, every unique pairing of inputs and outputs for a given model) to be a separate scenario. Even infinitesimally different outputs are considered separate scenarios. The focus lies on the individual outputs. The schism in usage of the term ”scenario” can be traced to Kwakkel and Pruyt (2013), where time series model outputs are first perceived no longer as states of the world at a specific time (the original usage), but as developments over time. The term ”transient scenario” (Haasnoot, Schellekens, Beersma, Middelkoop, and Kwadijk, 2015; Kwakkel et al., 2016; Werners, Pfenninger, van Slobbe, Haasnoot, Kwakkel, and Swart, 2013) is also used for this understanding of a scenario as a state changing over time.

The two usages given above represent a significant difference in conceptual understanding of what a scenario is, although this is not surprising, as the term has a wide variety of meanings and interpretations across different disciplines (Börjeson, Höjer, Dreborg, Ekvall, and Finnveden, 2006). In this thesis, I will avoid using the term scenario where possible. Where I do use it, my meaning is that of the ”transient scenario” - a single experiment outcome, clearly relatable to a unique set of input parameters - unless clearly stated otherwise.

### 2.1.2. Scenario Discovery in Model-Based Decision Support

A review of model-based decision support literature indicates scenario discovery is used in two primary roles in the modelling and analysis workflow (Halim et al., 2016).

An important step after creating a simulation model is to explore it to understand what behaviors the system is capable of generating, and how it interacts with policies, externalities and uncertainties. The goal may to understand under which conditions a policy may do well, what the worst possible system performance could be, or similar considerations. Scenario discovery is useful as a standalone technique for this exploration of model behavior (Gerst et al., 2013; Kwakkel and Pruyt, 2013; Rozenberg, Hallegatte, Vogt-Schilb, Sassi, Guivarch, Waisman, and Hourcade, 2010).

Building on the standalone usage of scenario discovery in model behavior exploration, it can also be used in iterative decision-making or policy-finding processes such as Robust Decision Making (Lempert et al., 2003,0), or its development, Many Objective Robust Decision Making (Kasprzyk et al., 2013). These methods use iterative cycles of simulation, assessment, and (human-in-the-loop) policy modification to find robust and adaptive strategies for managing complex systems. Scenario discovery represents the link between one iteration’s model outputs, which are assessed by the modeller, and the next iteration’s policy inputs, which are revised based on the insights from the scenario discovery.

### 2.1.3. Patient Rule Induction Method

A comparison of scenario discovery methods (Lempert et al., 2008) gives two main methods - PRIM and CART. The Patient Rule Induction Method (Friedman and Fisher, 1999), or PRIM, was originally developed as a tool for predictive data analysis, with cited examples including marketing and geology. It is a bump-hunting algorithm - it looks for regions in hyperdimensional data space where some values are higher than elsewhere. The algorithm then tries to find the bounding box which usefully fits around these bumps, in effect inducing the rules that create the bump.

Consider a collection of points which exists in *n*-dimensional space. These points might be data entries in a table, such as students and their attributes (age *a*, distance between Delft and place of birth *b*, and GPA *p*). To determine which conditions likely predict a high GPA, PRIM constructs a 2-dimensional space of *a* and *b*, and assigns each data entry to a point in this space, with the GPA as value:

$$p = f(a, b)$$

This (simplistic) representation of academic success might now be used to determine which students are likely to have a high GPA, based on their circumstances.

Starting from a box representing the entire input space, PRIM removes small slices of the dataset. The peel to be removed is determined by the objective function, which attempts to maximize the value of the points in the box - at each step, the peeling action which improves the box value the most is chosen. This iterative restriction of dimensions eventually leads to boxes of higher and higher GPA, but as the peeling also (albeit slowly) removes data points from the box, the usefulness of the box slowly decreases. Therefore, the objective function must also consider the coverage (points in the box vs. points in the set) and density (points of high value in the box vs. points of high value in the set). Finally, as the box should be as simple as possible, the objective function also attempts to maximize interpretability my minimizing the number of dimensions which are restricted. This process is referred to as (iterative) peeling. PRIM can also paste (enlarge boxes rather than reduce them) if the objective function finds this to be useful. However, (Friedman and Fisher, 1999) indicate that the pasting rarely has significant effect on the final boxes.

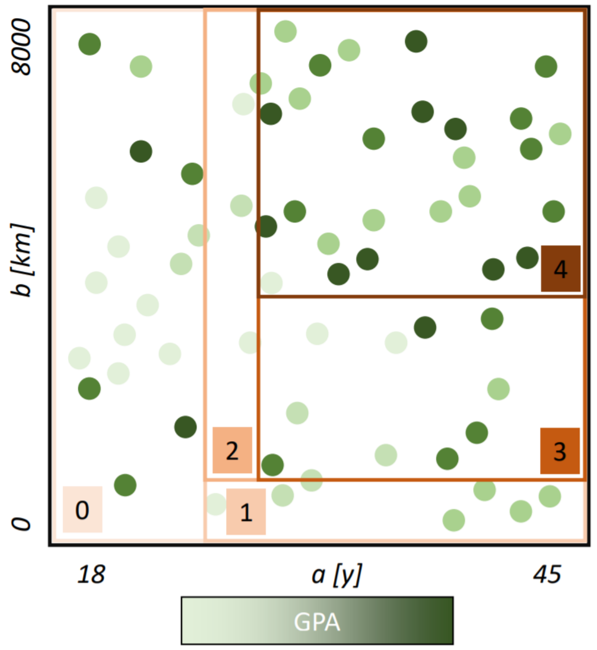



Figure 2.1: PRIM sequence of operations

In the presented example, PRIM might find that students who come to Delft from far away and are no longer in their teens or twenties are likely to have the highest GPAs, represented by box 4. The figure shows only peeling steps without any pasting.

In Bryant and Lempert (2010), PRIM is adapted to scenario discovery in exploratory modelling. The authors propose a slightly revised objective function, where model outputs - in this case, time series outputs for relevant metrics - are classified binarily whether they are of interest or not by testing them against a threshold. This moderately changes the coverage and density calculations (which tend towards 1 for a ”perfect” box), but otherwise has no significant effect. The advantage of this approach is that the binary classification can be extended to any desirable metric - the outputs with the least variance over time, or the outputs with the highest peak values, or the outputs which oscillate. However, this flexibility also requires some understanding of the system in that the modeller must explicitly specify the metric of choice.

Multiple researchers have modified and enhanced the PRIM technique presented in Bryant and Lempert (2010). In Dalal et al. (2013), PCA PRIM is presented, which applies principal component analysis to inputs space in order to orthogonally rotate it into an orientation more suitable for PRIM to work on. This has the advantage of creating more orthogonally aligned subspaces, but can reduce clarity as the input axes now represent linear combinations of (potentially completely unrelated) input parameters rather than single parameters.

A further improvement found in literature is the adaptation of PRIM to operate on heterogenous data types, presented in Kwakkel et al. (2016). This is achieved through changes in the PRIM objective function. This is a marked improvement over standard PRIM implementations, as binomial, multinomial and continuous model inputs can now be processed together.

PRIM and its variants are used in a wide variety of model-based decision support studies, including water scarcity (Trindade, Reed, Herman, Zeff, and Characklis, 2017), technological progress forecasting (Hamarat, Kwakkel, and Pruyt, 2013), development of global shipping routes (Halim et al., 2016), and climate mitigation (Greeven et al., 2016).

### 2.1.4. Classification and Regression Trees

CART, or Classification and Regression Trees, is a classification method first introduced by Breiman, Friedman, Olshen, and Stone (1984). As PRIM has been found to be more useful in the scenario discovery role (Lempert et al., 2008) and will be my method of choice for this thesis, this subsection is mostly for context.

CART is in some respects quite similar to PRIM. It also seeks to find subspaces with higher-than-expected values. The chosen method here is to split the space at every step, continually chopping away insufficiently valuable data points. This algorithm is considered ”greedy” (Lempert et al., 2008) as it removes many data points in very few pruning cycles, quickly reducing the quality of the found restricted subspaces. Nevertheless, it can also achieve useful results comparable to PRIM for certain data sets. The resulting pruning trajectory, which splits in two branches at every step, can be likened to a tree where unsuccessful branches (that is, input subspaces generating unsatisfactory or uninteresting outputs) are pruned and ignored. CART has no analogue to PRIM's peeling.

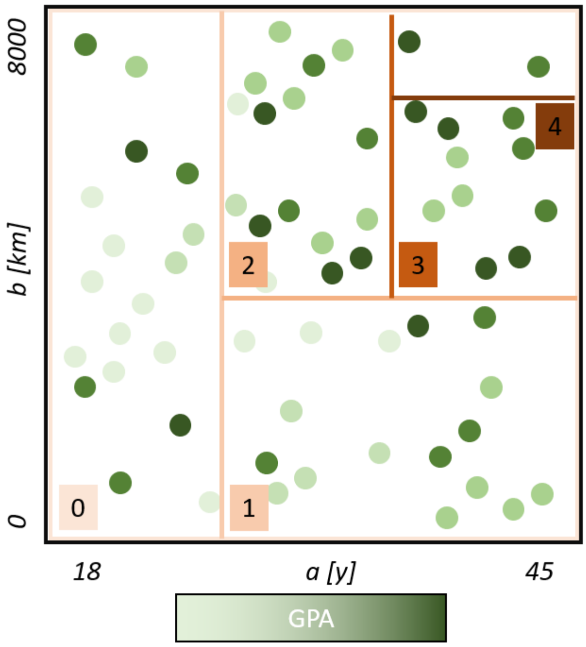



Figure 2.2: CART sequence of operations

In the presented figure, CART finds a similar subspace to PRIM, but using a completely different trajectory of input space reduction.

### 2.1.5. Rule Induction Methods under Development

PRIM characterizes rule induction as a single objective optimization problem - maximizing density of cases of interest within the induced box, and iteratively reducing the size of this box to improve this density. The two additional metrics of coverage and interpretability are only assessed after the fact. An ongoing line of research is the re-casting of rule induction from a single objective optimization problem to a many objective optimization problem (Kwakkel, 2018), which should provide a clearer understanding of the tradeoffs between the three metrics when inducing rules. However, the inherent drawback of PRIM's orthogonal dimension restricting is not avoided through this method - the resulting boxes may be better, but they are still boxes.

## 2.2. Time Series Clustering

Time series are chronological sequences of observations on a variable of interest (Montgomery, 2016). The observations are commonly categorical or scalar values, evenly spaced over some interval. Examples include a stock's closing price over a year (see Figure 2.3), or monthly rainfall statistics over a year. They are often used in forecasting and planning processes to understand trends and future developments (Cryer, 2008).

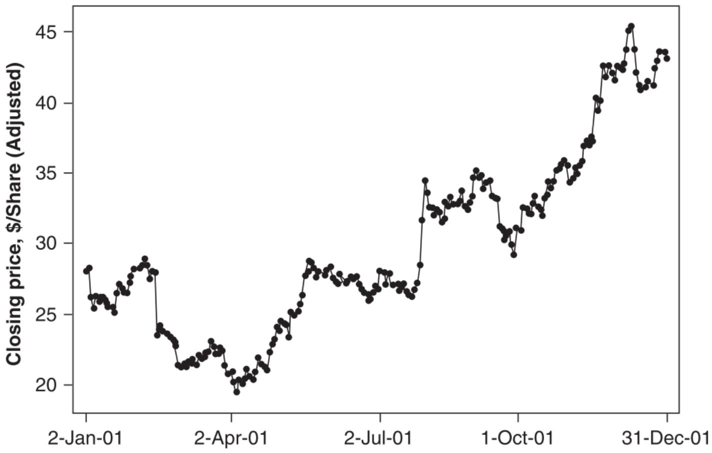



Figure 2.3: Exemplary time series of a publicly traded stock, from Montgomery (2016)

Where multiple related time series are present, the time series cannot just be analyzed independently (regression, mean, bounds, etc.), but also in relation to one another. Such analysis broadly falls into two categories - clustering and classification. The former is an unsupervised method which attempts to group certain time series together based on a specific similarity metric, while the latter is a supervised method which attempts to identify to which predefined class a time series belongs. This literature review will focus on clustering as delineated in the Introduction.

### 2.2.1. Clustering Concepts

Time series clustering refers to the process of grouping a variety of time series into ”sets” or ”clusters” based on their similarity to each other. A wide variety of methods exists, but they generally follow a three-step process:

1. Select or generate a set of time series to be clustered.
2. Specify a (dis)similarity metric, and calculate the (dis)similarity between every pair of time series.
3. Specify a clustering style, and use similarity data to place each time series relative to the others within the chosen framework.

Selecting or generating time series is done from an existing data set or by performing experiments on a simulation model, respectively. Simulation models are exceptionally suitable for time series generation because they inherently fulfill many of the limiting criteria of time series analysis. Clustering methods often struggle with imperfections such as missing data points, unequal time series lengths, different value scales or lack of variety (Montgomery, 2016). These issues are averted in simulation models - a virtually limitless set of time series model outputs can be generated at whim, output length is controllable, and the generated series have a consistent time step and broadly comparable value ranges.

Similarity metrics define how two time series are assessed for likeness, and may be feature-, data- or model-based (Warren Liao, 2005). Feature-based metrics attempt to extract certain features from each time series, such as number of peaks, mean value, or oscillation period, and then find similarities in those features. This can be considered a (significant) reduction in time series dimensionality, which explains why these methods are often very computationally fast. It also makes them less constrained regarding the imperfections mentioned above. Data-based methods attempt to match every observation in one time series with every observation in another time series, and then measure the distance or dissimilarity between them. They are generally computationally expensive, especially for long time series, and sensitive to inherent differences in time series such as length. Normalization may also be required for time series with vastly differing amplitudes/value ranges. Model-based methods attempt to reduce each series to a generative model of itself, and then group the series by their models (or similarities thereof). These models can range from simple linear regressions over Auto Regressive Integrated Moving Average (ARIMA) to hidden Markov chain models. The distinction between feature- and model-based methods is not always clear. Models are simplified abstractions of reality - but depending on the purpose, a descriptive collection of time series features may also be considered a model.

Clustering styles define how the resulting similarities between every pair of time series should be structured. This can be done as partitional clustering, where all time series are separated into $k$ clusters, or hierarchical clustering, where time series are merged bottom-up into larger and larger clusters (thus eliminating the need to specify $k$). Hierarchical clustering can be seen as iterative partitional clustering. While some consider the latter more useful (Ratanamahatana, Lin, Gunopulos, Keogh, Vlachos, and Das, 2010), it is computationally far more demanding, and therefore limited to smaller datasets. Partitional clustering can be done using hard (every time series belongs to one cluster) or soft (every time series belongs to every cluster to some degree) clustering rules.

### 2.2.2. Clustering Solution Validity

The analysis of clustering solutions is difficult. Two general approaches exist: internal and external validation (Montgomery, 2016).

Internal validation considers all time series within a specific cluster, and evaluates how similar they are. A variety of methods exist for this purpose. Internal validation is highly context- and method-dependent, since no optimal clustering algorithm exists (Cryer, 2008). Nevertheless, a recent comparison of cluster validity indices (Arbelaitz, Gurrutxaga, Muguerza, Pérez, and Perona, 2013) indicates that some methods are more accurate than others for assessing cluster validities where some ground truth is present. This comparison explicitly mentions the silhouette width (Rousseeuw, 1987) as a very good validity index. The silhouette of a cluster is given by comparing the similarity of that cluster's members to each other compared to the similarity of the members with those of another cluster. Thus, for each cluster member, the next best cluster is determined. The average value across all time series for their similarities to their cluster's co-members over their similarities to the next best cluster's members tends towards 1 for a perfect clustering solution, and -1 for a perfectly wrong clustering solution. Thus, the maximization of the silhouette width can be used as a criterion for finding decisive clustering solutions. This does not imply the clustering solutions are correct, they merely represent well-performing partitions for the chosen similarity metric. By extension, the silhouette width can also be used to compare clustering solutions for different numbers of clusters $k$ to determine which choice best represents and partitions the data.

External validation refers to the comparison of a proposed clustering solution of data with an external "true" clustering of that data. For example, one might combine a dataset containing 20 time series of average rainfall over a year, and another dataset containing 20 stock market closing prices over a year, and then test whether a clustering method is able to separate the 40 time series correctly back into two sets of 20. This requires that some external truth, i.e. true cluster memberships, is known. In the context of nonlinear dynamics models, outputs can be separated into clusters based on their input values in relation to the bifurcation behavior of the models.

### 2.2.3. Selected Clustering Methods

As this thesis is about evaluating time series clustering as a tool to overcome the inherent difficulties (and drawbacks) of user-in-the-loop identification of outputs of interest, rather than developing a time series clustering algorithm for that specific purpose, I am necessarily restricted to using those time series clustering methods already implemented and publicly available as computer code. A review of two common open-source data analysis languages, R and Python, reveals that time series clustering is performed almost exclusively in R. Specifically, I identify the packages given in Table 2.1 as useful for my purposes.

| **Package** | **Reference** | **Purpose** |
|---|---|---|
| TSclust | Montero and Vilar (2014) | a variety of time series clustering methods |
| dtwclust | Sarda-Espinosa (2017) | a variety of data-based time series clustering methods |
| seqHMM | Helske and Helske (2017) | data analysis using hidden Markov models |
| cluster | Rousseeuw et al. (2018) | various clustering analysis tools |
| clue | Hornik and Böhm (2017) | various clustering analysis tools |

Table 2.1: Reviewed R packages for time series clustering

From the packages listed in Table 2.1, I select a variety of existing time series clustering methods for consideration in this thesis. They are given in Table 2.2. I base the selection on two criteria - clustering performance (both intuitive usefulness and execution speed) on a simple test model, and the similarity concept. To explore how different clustering concepts perform in the scenario discovery context, I aim to get a roughly even distribution of feature-, data- and model-based methods. The final list includes three feature-, six data- and five model-based techniques.

For each method, I give a three- or four-letter code. I will use these codes throughout this thesis to refer to the different methods. The codes are taken from the implementation packages and generally refer to the distance metric used in the method.

### 2.2.4. Description of Methods

ACF (Galeano and Pella, 2000) finds the autocorrelation function for every time series and determines the distance between time series by comparing the parameters of their autocorrelation functions.

CID (Batista, Wang, and Keogh, 2011) measures the Euclidean distance between time series by pairing points in each series with each other, and calculating their difference in value. This distance is then corrected by the estimated complexity, which is calculated as the root of the sum of the squares of every value difference between two consecutive observations. The correction is based on the notion that complex time series are often (incorrectly) considered further apart than they truly are by established time series clustering methods.

CORT (Chouakria and Nagabhushan, 2007) similarly measures the raw Euclidean distance between two time series, but then corrects this value with a temporal correlation coefficient, which considers how similar the two time series' dynamic behaviors are at every point in time. This method bridges the gap between data- and model-based methods.

DWT (Zhang, Ho, Zhang, and Lin, 2006) breaks every time series down into a collection of parametrised wavelets. This is conceptually similar to a Fourier transformation. Similarity is then assessed using the parameters of these wavelets.

LLR (Vilar and Pértega, 2004) takes the logarithmic spectra of the time series in question and determines the dissimilarity from the difference in spectra between two time series.

LPC (Kalpakis, Gada, and Puttagunta, 2001) calculates the inverse Fourier transforms of the logarithmic amplitude spectra of the ARIMA models of two time series. The Euclidean distance between the Fourier coefficients is then used as the distance between the time series.

PDC (Brandmaier, 2015) uses the Kullback-Leibler divergence to find the distance between the permutation distributions of a time series, which characterize its complexity.

PER (de Lucas, 2010) uses the periodogram (analysis of frequencies in data) of each time series to determine the similarity between series. This method is an example of where the boundaries between feature- and model-based methods are blurred - the inherent frequencies of a time series are clearly features, but the time series could also usefully be reconstructed from the periodogram, possibly making it more model-based.

PIC (Piccolo, 1990) defines the dissimilarity of two time series as the Euclidean distance between the AR operators of the ARIMA model representations of two time series.

DTW (Sarda-Espinosa, 2017) uses dynamic time warping to measure the distance between two time series. Rather than applying Euclidean distance, where the observation values of tow time series are compared at every time step, DTW allows the series to be stretched and shifted within limits. The time series therefore do not map one-on-one anymore, as one observation in one series may be matched to multiple observations in the other series. This method is stochastically influenced.

SBD (Paparrizos and Gravano, 2015) uses Fast Fourier Transforms to find the distance between the normalized coefficient sequences of two time series. This method is stochastically influenced, and may be sensitive to 32-bit computing architecture due to the FFT numerical precision requirements.

GAK (Cuturi, 2011) considers all possible alignments of two time series, and then selects the most efficient alignment possible. From this alignment, the distance is computed. This method is stochastically influenced.

TAD (Begum, Ulanova, Wang, and Keogh, 2015) was proposed as a method to speed up dynamic time warping-based clustering. Rather than find the distance between all time series, the algorithm prunes those calculations which are unlikely to produce a distance low enough for the two time series to be considered members of the same cluster. The distance calculation itself follows dynamic time warping.

LCM (Helske and Helske, 2017) uses hidden Markov models to cluster time series. The premise is that dependencies between time series can be traced back to a single latent class (the cluster identifier). A Markov model with a single hidden state for each cluster is randomly generated and then fitted to the supplied time series. As Markov models are transition likelihoods between categorical states, the time series (ordinal time axis, continuous observation value axis) to be analyzed must first be transformed into state sequences (ordinal time axis, ordinal observation state axis). However, the number of ordinal states into which the observation values can be discretized is not obvious. Therefore, I implement a Bayesian Information Criterion (Montgomery, 2016) to evaluate the validity of a variety of different discretization choices (number of possible observation states). The most representative number of states is then applied in the latent class Markov model for clustering. This method is heavily stochastically influenced and regularly fails to converge. I therefore apply this clustering method in a try-loop for up to 20 attempts until any solution is found. Because of this workaround, the clustering solution returned should be treated with caution.

### 2.2.5. Prior Applications of Clustering in Exploratory Modelling

Multiple researchers have used output clustering in model-based decision support and exploratory modelling in the past.

In Pruyt et al. (2011), Adaptive Bayesian Pooling (Duncan, Gorr, and Szczypula, 2001) is used to extract a set of representative system behaviors (cluster prototypes) from a set of 2000 experiments. I believe this was done as a form of open exploration to understand the variety of possible dynamics. The clusters are not investigated further.

While Gerst et al. (2013) do not investigate time series per se, they do cluster experiments using multiple scalar outcomes of interest, thus extending the established scenario discovery methodology to consider multiple outcomes of interest. The chosen clustering method is not explicitly described beyond being hierarchical.

In Kwakkel, Auping, and Pruyt (2013), the time series model outputs are broken down into atomic behaviors such as exponential growth or linear decline, and then hierarchically clustered on their sequences of atomic behaviors. This is essentially a featurization approach, and is derived from Yücel and Barlas (2011).

A similar atomic behaviors approach is used in Kwakkel and Pruyt (2013), though it is based on Ford (1998). The authors specifically conclude that more research on time series clustering in exploratory modelling is required.

| Code | Distance metric | Package | Family | Reference |
|---|---|---|---|---|
| ACF | Simple autocorrelation coefficients | TSclust | model | Galeano and Pella (2000) |
| CID | Complexity-invariant Euclidean distance | TSclust | data | Batista et al. (2011) |
| CORT | Combined temporal correlation and raw values behaviors | TSclust | data | Chouakria and Nagabhushan (2007) |
| DWT | Dynamic wavelet transformation | TSclust | feature | Zhang et al. (2006) |
| LLR | Local-linear estimation of log-spectra | TSclust | feature | Taylor et al. (2014) |
| LPC | Linear predictive coding coefficients | TSclust | model | Kalpakis et al. (2001) |
| PDC | Permutation distribution distance | TSclust | feature | Brandmaier (2015) |
| PER | Integrated periodogram dissimilarity | TSclust | model | de Lucas (2010) |
| PIC | Piccolo's ARIMA distance | TSclust | model | Piccolo (1990) |
| DTW | Dynamic time warping | dtwclust | data | Sarda-Espinosa (2017) |
| SBD | Shape-based distance | dtwclust | data | Paparrizos et al. (2015) |
| GAK | Fast global alignment kernels | dtwclust | data | Cuturi (2011) |
| TAD | Time-series anytime density peaks | dtwclust | data | Begum et al. (2015) |
| LCM | Latent class (hidden Markov) model | seqHMM | model | Helske and Helske (2017) |

Table 2.2: Selected clustering methods

While the latter two works are conceptually similar to my thesis, they do not demonstrate the effects of different clustering algorithms, and do not consider the implications of non-orthogonal input subspaces beyond using an improved version of PRIM called PCA-PRIM (Dalal et al., 2013).

## 2.3. Nonlinear Dynamics in Systems

Dynamics is the study of systems changing over time (Strogatz, 1994). These dynamics can be linear or nonlinear, the latter including feedback effects - a value growing proportionately to its own size, or similar. A linear system might be the price of a good as a function of its material costs, while a nonlinear system might be the population of a country - as more children are born, those children eventually go on to have even more children, who in turn have even more children! These intrinsic connections or loops can result in a variety of interesting system behaviors such as goal-seeking, regular oscillations or exponential/infinite growth. However, they also make the system much harder to analyze. While (models of) linear systems can be broken down and recombined at will, nonlinear systems require feedback loops to function as expected, and can therefore not so easily be separated.

> Systems of information-feedback control are fundamental to all life and human endeavor, from the slow pace of biological evolu- tion to the launching of the latest space satellite. . . . Everything we do as individuals, as an industry, or as a society is done in the context of an information-feedback system. - Jay W. Forrester

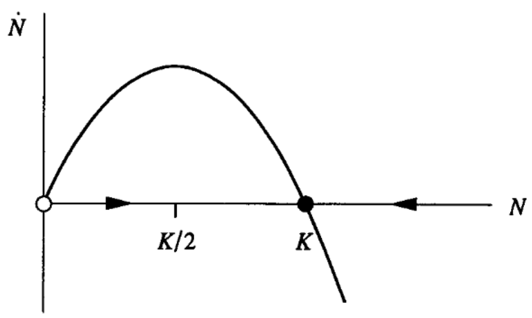
Nonlinear dynamics can be broadly structured into one- and two-dimensional problems, or flows (Strogatz, 1994). The term *flow* is used as the qualitative solutions of nonlinear dynamic equations are easiest conceptualized as as an infinite number of trajectories flowing through hyperdimensional space with dimensions $\{x, \dot{x}, \ddot{x}, ...\}$ where $x$ is a quantity of interest.

### 2.3.1. One-Dimensional Flows

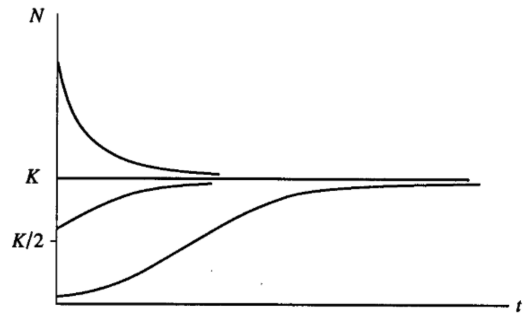
Consider a simple population model with population $N$, growth rate $r$ and carrying capacity $K$:

$$\dot{N} = rN(1 - \frac{N}{K})$$

A two-dimensional plot of $\{N,\dot{N}\}$ given in Figure 2.4(a) allows graphic investigation of the system behavior. Two fixed points are apparent - an unstable point (white) and a stable point (black). The stability is indicated by the direction of the phase arrows, which always point towards stable points. The stable point lies at $N = K$ - the population will stabilize at the carrying capacity, which makes intuitive sense. In Figure 2.4(b), a number of qualitative solutions is shown - depending on whether the population level is instantiated above or below the carrying capacity, it will show sigmoid or S-shaped growth/decline to that value. This is equivalent to moving along the phase line (following the arrows towards the stable point) in Figure 2.4(a).




(a) Phase plot




(b) Qualitative solutions

Figure 2.4: Stability of a simple population model

As the parameters of a nonlinear function change, the phase line itself shifts around the phase plot. By extension, this means the fixed points (intersections of the phase line and $x$ axis) move as well. If the parameter changes are large enough, new fixed points may be created (new intersection of phase line with $x$ axis), existing ones may collide (phase line tangentially touches $x$ axis) or vanish (phase line no longer intercepts $x$ axis). Such fundamental changes can have significant effects on model behavior, and are referred to as bifurcations. The parameter values which cause such transitions in phase behavior are called bifurcation points. Bifurcations provide models of transitions and system instabilities as control parameters are varied. A simple example is the buckling of a thin beam under compression. As the compression is increased (control parameter), the beam slightly compresses until it eventually violently buckles - a bifurcation occurs, dependent on how strong the beam is and how forceful the compression.

Bifurcations in both one- and two-dimensional flows are interesting because they create significantly different output behavior for minor changes in input values. Consider the beam example - If Euler's critical load for a beam is given as $F = 40kN$, a shift in load from $30kN$ to $39.95kN$ will have almost no noticable effect on the beam. However, a further $0.1kN$ load will cause the beam to spectacularly fail.

### 2.3.2. Two-Dimensional Flows

If a system is defined by differential equations for two quantities of interest, it is considered a two-dimensional flow. While the analysis of such systems is conceptually similar to one-dimensional flows, it is far more difficult to examine them, as the formulas are too complicated to provide much insight (assuming they are even available), and the phase plots technically consist of an infinite number of possible solutions. Thus, two-dimensional flows are often analyzed qualitatively using a small selection of phase lines. An example of such a phase portrait is given in Figure 2.5. Note how the phase lines now also go along the vertical axis, and by linear combination in every planar direction. Here, $A$, $B$, and $C$ are unstable fixed points, while $D$ is a stable closed orbit, or periodic solution.

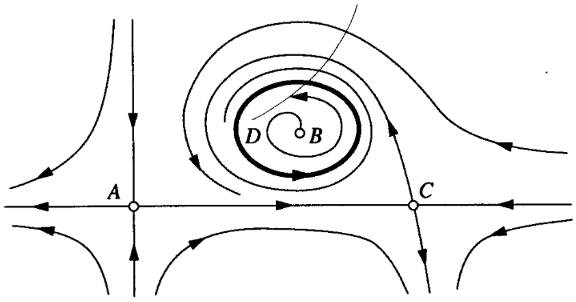



Figure 2.5: Phase portrait of a two-dimensional flow

Bifurcations occur in two-dimensional flows much like they do in one-dimensional flows. However, they show a wider variety of bifurcation behavior, which can not usefully be deduced from the defining equations anymore.

## 2.4. Chapter Summary

Scenario discovery is a technique within model-based decision support, used primarily to find decision-relevant input subspaces using external criteria applied to model outputs. The Patient Rule Induction Method is the most common tool for rule induction in scenario discovery. Time series clustering describes the process of taking a set of time series and grouping them based on some similarity metric. Three general categories of similarity metrics exist: model-, data- and feature-based approaches. A wide variety of different methods are available for each approach. Nonlinear dynamics in systems are generally the product of feedback loops, and make analysis of such systems difficult. Bifurcation points and curves describe when nonlinearities occur in such models, an are key to understanding why small changes in input values can cause dramatic shifts in output values.

# 3
# Budworms

To compare the performance and effects of different time series clustering methods and parameters in conjunction with PRIM, two test cases are introduced in the following two chapters - a one- and a two-dimensional flow. They are interesting in the context of scenario discovery because they exhibit nonlinear behavior, which conventional scenario discovery struggles with.

## 3.1. The Spruce Budworms Population Model

A simple model of an insect population in a forest was presented by Ludwig, Jones, and Holling (1978) and later reproduced in a slightly modified form by Strogatz (1994), which is the definition used here. The model was also discussed in Meadows (2012) as an exemplary model of a system with unexpected feedback effects. Strogatz (1994) gives the overall insect population ($N$) change rate as

$$\dot{N} = RN(1 - \frac{N}{K}) - p(N)$$

with $R$ as the growth or reproduction rate of the budworms, and $K$ as the carrying capacity of the forest. Ludwig et al. (1978) specify the predation rate as

$$p(N) = \frac{BN^2}{A^2 + N^2}$$

where $A, B > 0$ and represent some parameters of the ecosystem affecting predation, such as leaf density or predator behavior constraints. Thus the overall population change rate becomes

$$\dot{N} = RN(1 - \frac{N}{K}) - \frac{BN^2}{A^2 + N^2}$$

where the minuend determines growth of the population, and the subtrahend predation of it.

This four-parameter $(R, K, A, B)$ form can be reduced to a dimensionless two-parameter form for easier analysis:

$$\dot{x} = rx(1 - \frac{x}{k}) - \frac{x^2}{1 + N^2}$$

with $x$ as the dimensionless population ($x = \frac{N}{A}$), $r$ as the dimensionless growth or reproduction rate, and $k$ as the dimensionless carrying capacity of the forest. Again, minuend and subtrahend represent growth and decline of the population level, respectively.

The two-parameter form given above can be translated into a system dynamics model (Figure 3.1) using a single stock for the population (with $x_0$ as initial population), and one in- and outflow $(c, p)$ respectively for creation and predation of budworms. Model inputs are highlighted in green. While the form shown earlier gives a single equation for the change rate of the population, system dynamics best practices dictate avoiding negative inflows, thus, the minuend and subtrahend are split into separate flows. This model is implemented in Ventana Vensim, a system dynamics modelling program.

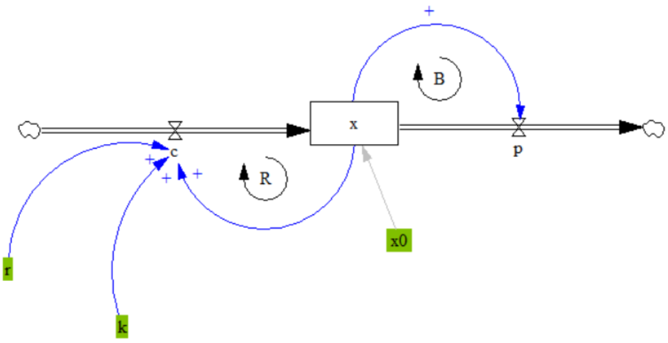



Figure 3.1: The spruce budworms population model in Vensim

The combination of one reinforcing and one balancing feedback loop with limiting exogenous conditions is commonly referred to as the "Limits to Growth" archetype (Braun, 2002), and was widely introduced by Meadows (1972) in the report of the same name. The behavior of the archetype - goal seeking behavior - is reflected well by the budworms model.

This model is interesting because it exhibits a cusp bifurcation with a pronounced bistable region. Figure 3.3(a), showing the output $x$ against the inputs $r, k$ visualizes this. The budworm population can exist in two different states - outbreak and refuge. These are the two stable fixed points of the system shown in Figure. Technically, a third fixed point exists between them, but is unstable and therefore only reached if the system is statically instantiated exactly at the fixed point.

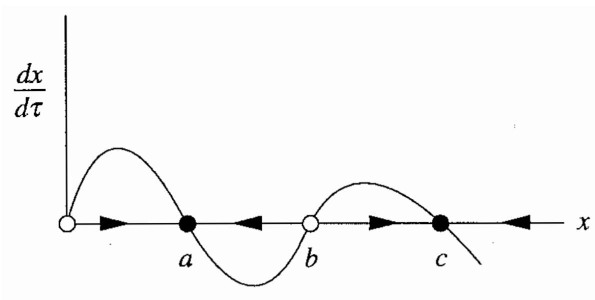
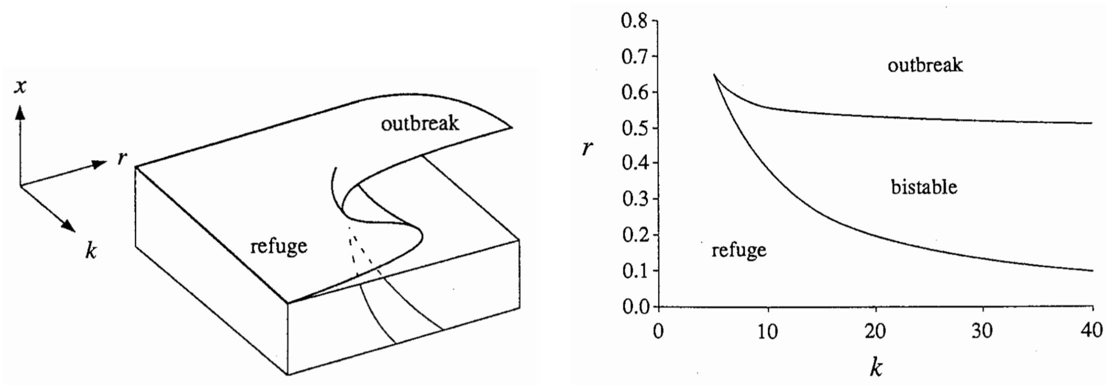



Figure 3.2: Phase plot of the budworms model

For fixed $r, k$ values, the population level shows classical goal-seeking behavior towards the nearest fixed point. However, if $r$ or $k$ (or both) were to vary over time, the population level could "fall over the cusp" by transitioning from one stable fixed point's region of influence to the other. Under certain conditions, a miniscule change in input values could trigger a significant change in output values - this is classical nonlinearity.

The overhang in Figure 3.3(a) represents the bistable region, where the function can produce two distinct output values for identical input value sets, depending on the prior states of the system. This makes the model non-holonomic.

In the $r, k$ plane of Figure 3.3(a), the two bifurcation curves can be seen as shadows of the cusp folds. This plane is plotted more precisely in Figure 3.3(b).




(a) Population levels $x$ over $r, k$ (b) bifurcation regions over $r, k$

Figure 3.3: Budworms model bifurcation details

## 3.2. Behavior Exploration

As the interesting behavior of this model only arises for varying parameters, such a variation must be introduced exogenously. Either parameter (or both) could be varied over time. In the interest of relating this exogenous change to a plausible real-world intervention, a variable decrease in the reproduction rate $r$ is introduced. This might be related to the application of reproduction-inhibiting pesticides, for example. The carrying capacity $k$ is held constant, as affecting the carrying capacity would functionally mean purposefully harming the forest.

To avoid introducing exogenous nonlinearities into the analysis of an endogenously nonlinear system, the decrease in $r$ is implemented as a smoothed step function, shown in Figure 3.4, where Vensim's default STEP() function is given in green, and the custom "continuous step" in red. Such a decrease also more closely reflects the way $r$ might change if such behavior was endogenized in the model, for example by coupling the budworms population model with a forest growth and pest control model.

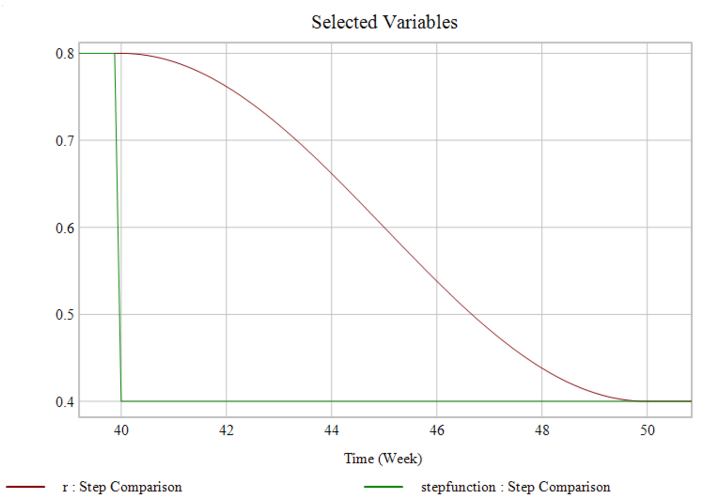



Figure 3.4: Vensim's default STEP() function and the implemented smoothed step function for $r$

With a constant value for $k$ and a variable decrease in $r$ referred to as $r_step$, the model again has two dimensionless parameters.

The Vensim model is loaded into the Exploratory Modelling and Analysis Workbench (Kwakkel, 2017). Under default Latin Hypercube sampling, 200 experiments are performed using the inputs specified in Table 3.1. $x_0$ and $t_{r_{step}}$ refer to the initial population of budworms, and the time at which the decrease in $r$ is initiated.

Table 3.1: Budworms inputs

| variable | min | max | static |
|---|---|---|---|
| $r$ | 0.45 | 0.7 | |
| $r_{step}$ | 0.1 | 0.4 | |
| $k$ | | | 10 |
| $x_0$ | | | 1 |
| $t_{r_{step}}$ | | | 40 |

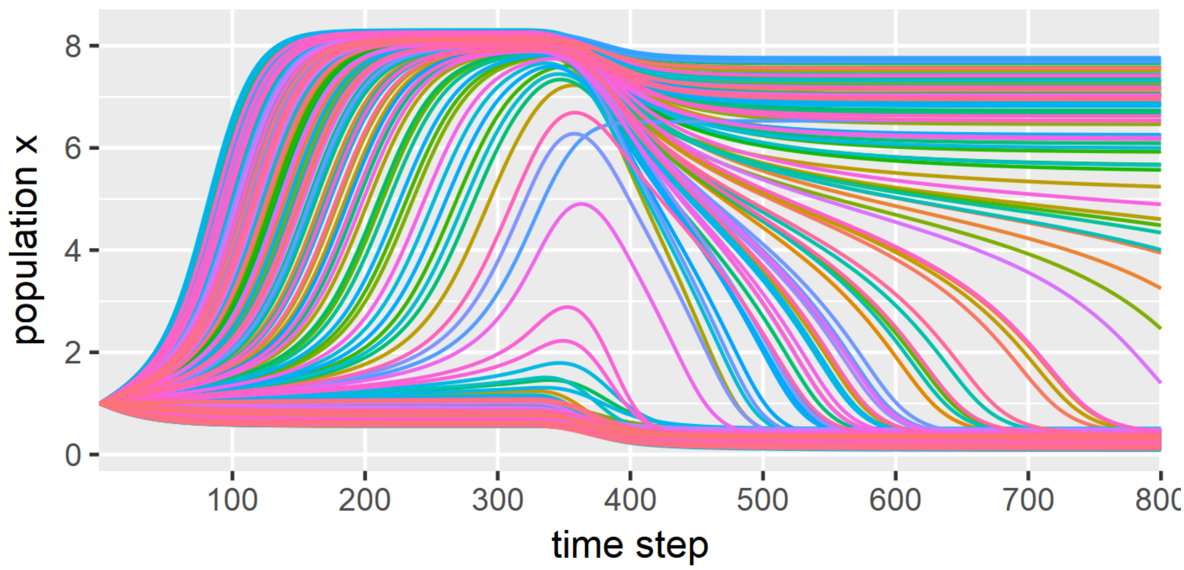



Figure 3.5: 200 experiments on the budworms model with varying $r$ and $r_{step}$

The results are exported into RStudio for plotting and analysis.

At first glance, three major groupings seem to appear - populations that stay above the cusp, population that stay below the cusp, and populations that fall over the cusp from outbreak to refuge levels. This makes intuitive sense, as the model has a bifurcation (generating two groups), and later introduces a parameter shift which further splits one of the groups in two. However, some time series cannot readily be categorized.

## 3.3. Number of Clusters

As many time series clustering methods are not parameter-free (that is, they require specification of certain parameters such as number of desired clusters, similarity search windows, etc.), the first step in clustering a novel set of time series is determining the number of clusters present in the data.

In Figure 3.6 the number of clusters $k$ found in the data is investigated using the silhouette widths. A higher silhouette width suggests better cluster validity (Rousseeuw, 1987). It appears that $k$ values of 2, 3 or 4 are broadly supported by the different methods, with some outliers.

As not all these methods are deterministic, but have stochastic influence through random seeding, unexpected trajectories should be considered with caution. The potentially significant effect of random seeds in the clustering process is demonstrated in Figure 3.7, where the same three clustering methods are performed five times each, with different random seeds. It appears that over multiple iterations, these stochastic methods also seem to favor $k$ values of 2,3 or 4 overall.

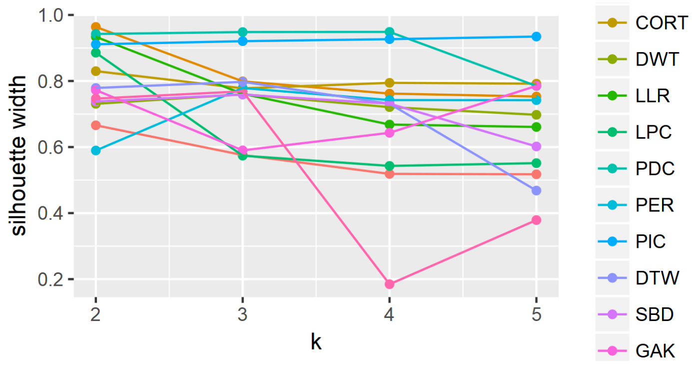



Figure 3.6: Silhouette widths for different clustering methods, budworms model

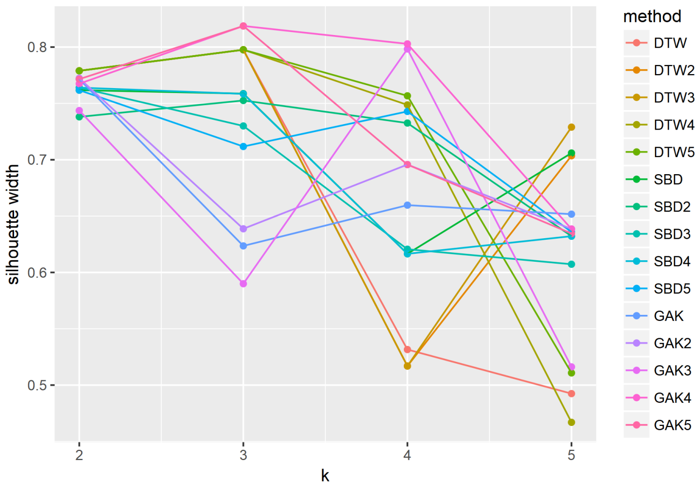



Figure 3.7: Influence of random seeding on silhouette widths, budworms model

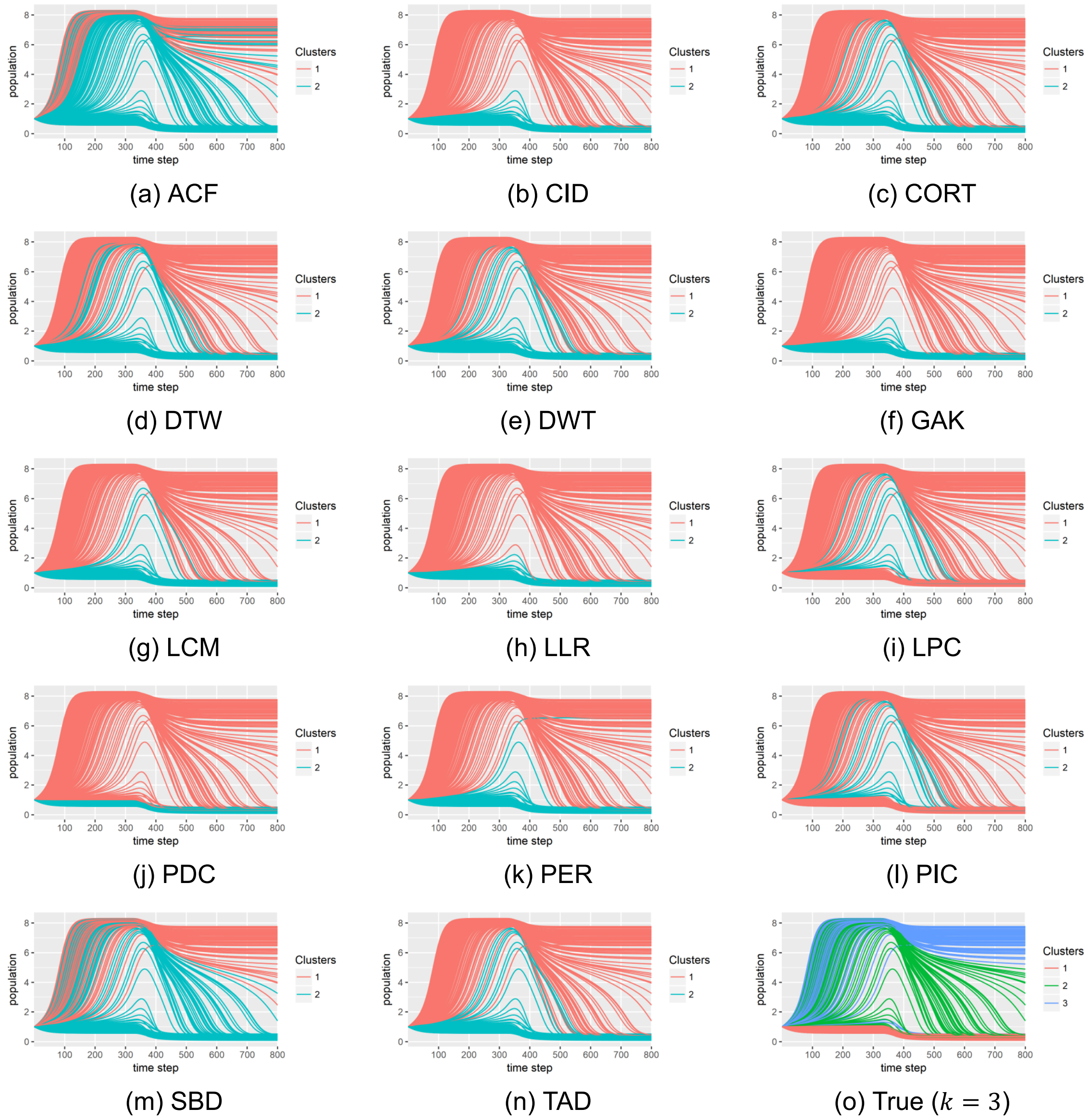
## 3.4. Clustering Solutions

Having determined that $k \in \{2, 3, 4\}$ is likely, clustering solutions for these alternatives can be generated and compared. For each implemented method, one clustering solution is generated and plotted for 2, 3 and 4 clusters. The cluster identifiers are matched to the true identifiers for $k = 3$ and to the LCM results for $k = \{2, 4\}$ using the Hungarian method (Kuhn, 2010). The true identifiers for $k = 3$ can be analytically determined from the model definition given earlier through bifurcation analysis. LCM is chosen as the reference solution for $k = \{2, 4\}$ because it consistently generated intuitive clusters over a variety of test models and cluster counts. As the matching is only for visualization purposes and does not affect the clustering solutions, any other method could also be chosen.

Visually, almost every proposed solution for $k = 2$ shown in Figure 3.8 propose reasonable clustering solutions. Most show very intuitive clusters, such as CID, GAK, PDC or TAD. Some other methods, such as ACF, PER or SBD, cannot intuitively be explained. In this context, an ”intuitive” cluster is understood to be one where all (or most) cluster members exhibit broadly similar behavior, there are no ”foreign” or ”non-matching” members in the cluster, and the clusters themselves appear to naturally divide the data into consistent subsets.

For $k = 3$ shown in Figure 3.9, some methods (CID, CORT, DTW, LCM, PER, SBD, TAD) show intuitive clustering solutions, while other methods (ACF, GAK, LPC, PDC, PIC) are not very insightful. A more precise comparison of each clustering solution with the true cluster members is presented later.

As the number of desired clusters grows beyond the true cluster count in Figure 3.10, some methods adapt better (CID, DTW, GAK, LCM, PER), others do not deliver intuitive outputs (ACF, PDC, PIC).




(a) ACF (b) CID (c) CORT
(d) DTW (e) DWT (f) GAK
(g) LCM (h) LLR (i) LPC
(j) PDC (k) PER (l) PIC
(m) SBD (n) TAD (o) True ($k = 3$)

Figure 3.8: Budworms clustering solutions for $k = 2$, including true clusters

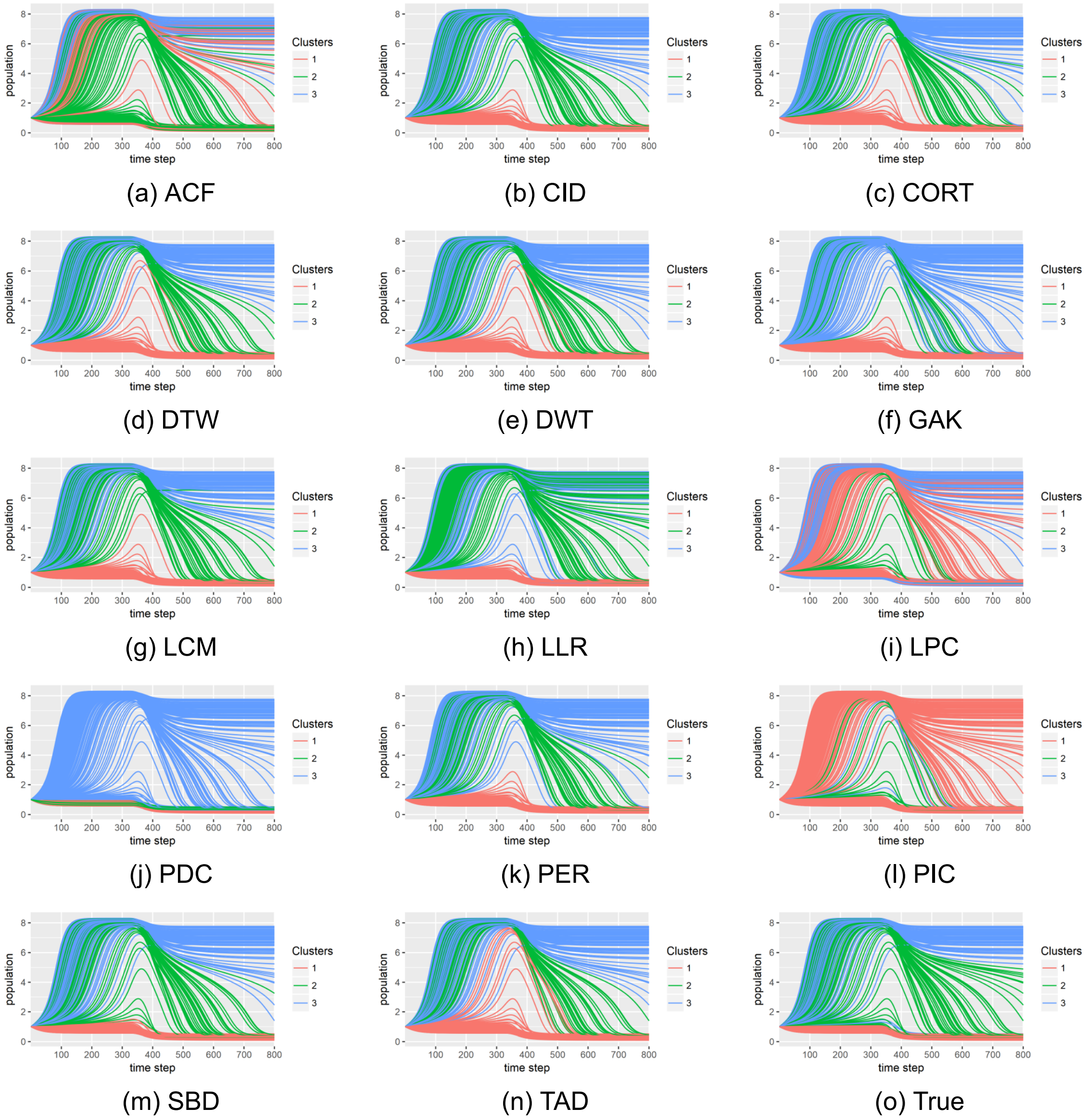



Figure 3.9: Budworms clustering solutions for $k = 3$

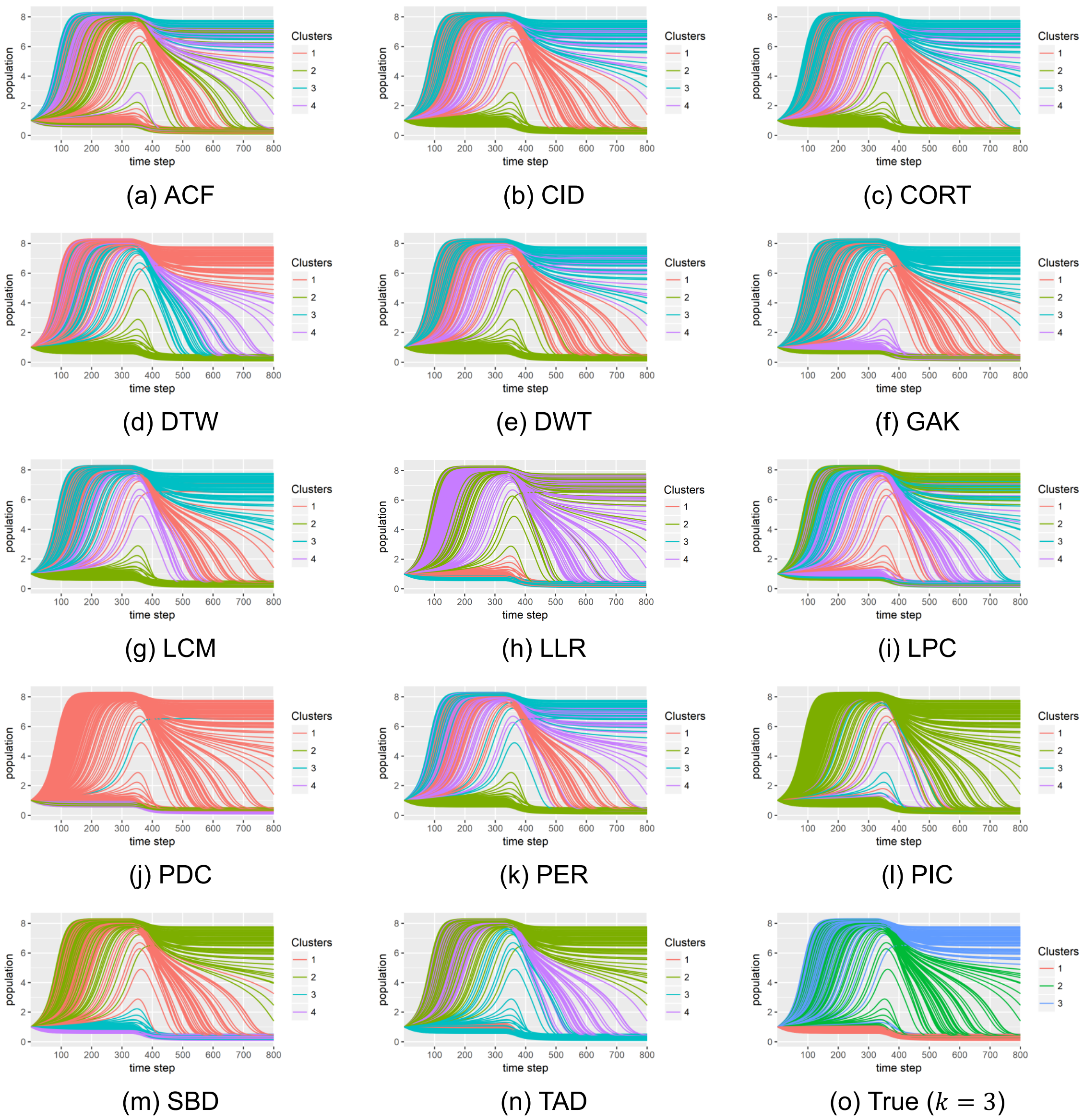


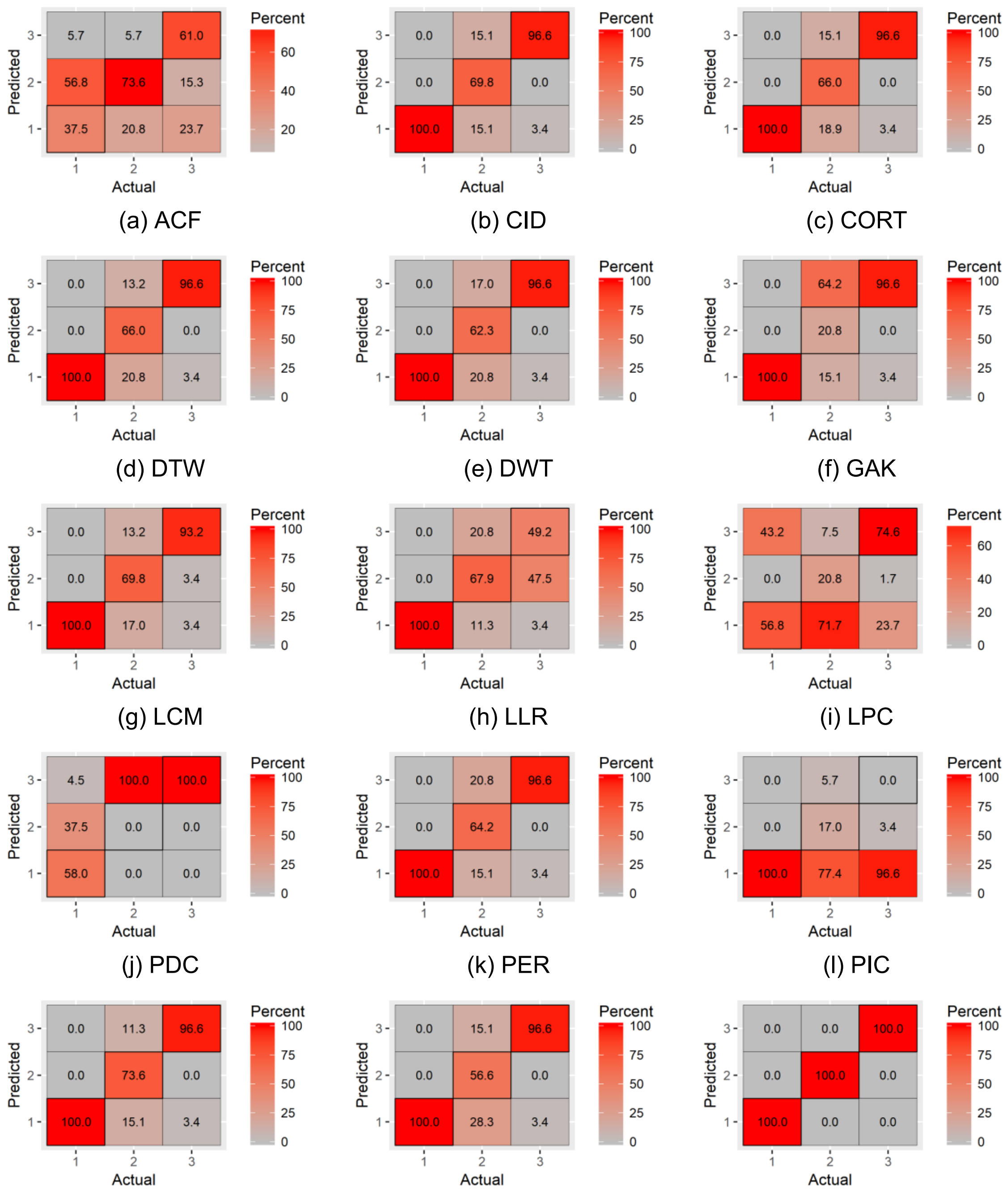

Figure 3.10: Budworms clustering solutions for $k = 4$, including true clusters

All clustering solutions presented above are influenced by a wide variety of factors, including random seeding, bounds and windows in the algorithms, and varying subsceptibility to time series length. Thus, it is difficult to draw general conclusions about ”right” and ”wrong” clustering methods. Nevertheless, for this specific data and the specific clustering parameters employed, some methods (CID, CORT, DTW, DWT, LCM, SBD) show intuitively sensible clusters for all ($k = 2, 3, 4$) proposed solutions.

## 3.5. Clustering Validation

The validity of a clustering solution cannot be objectively determined (Arbelaitz et al., 2013) without external knowledge. A number of cluster validity metrics exist which review the internal consistency of the clustering solutions. One such metric, the Silhouette width method (Rousseeuw, 1987) was used earlier to compare clustering solutions for different $k$ values.

For the presented budworms model, the true cluster number ($k = 3$) and the cluster members can be determined analytically and introduced as external knowledge. This allows validation of the proposed clustering solutions for $k = 3$ using confusion matrices, shown in Figure 4.10. For each clustering solution, every cluster is plotted against every cluster in a grid, with each grid cell containing the percentage of ”Actual” cluster members attributed to each ”Predicted” cluster. The closer a solution is to perfectly diagonal, the more accurate it is.




(a) ACF (b) CID (c) CORT (d) DTW (e) DWT (f) GAK (g) LCM (h) LLR (i) LPC (j) PDC (k) PER (l) PIC (m) SBD (n) TAD (o) True

Figure 3.11: Confusion matrices for clustering solutions vs. true cluster members, $k = 3$

## 3.6. Performance Comparison

Using confusion matrices, the objective performance of a clustering method can be evaluated. A good clustering solution (exemplified by Figure 4.10(o), which plots True vs. True clusters) will show high accuracy on the diagonal where cluster identifiers overlap.

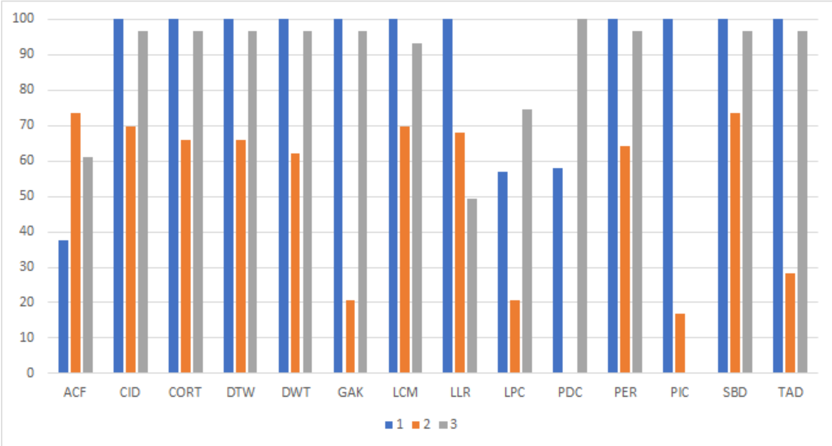



Figure 3.12: Clustering performance for each method, budworms model

Figure 3.12 plots the percentage of correctly identified cluster members for each method. For this specific combination of data and clustering parameters, CID, CORT, DTW, DWT, LCM, PER and SBD clearly surpass the other methods in terms of minimal and average performance.

While the various clustering methods are implemented in different packages and therefore likely with different levels of sophistication and efficiency, it is nonetheless interesting to compare their performance and speed. In Figure 3.13, each method's clustering accuracy for the $k = 3$ case and the time it took to complete the clustering are compared in a scatter plot.

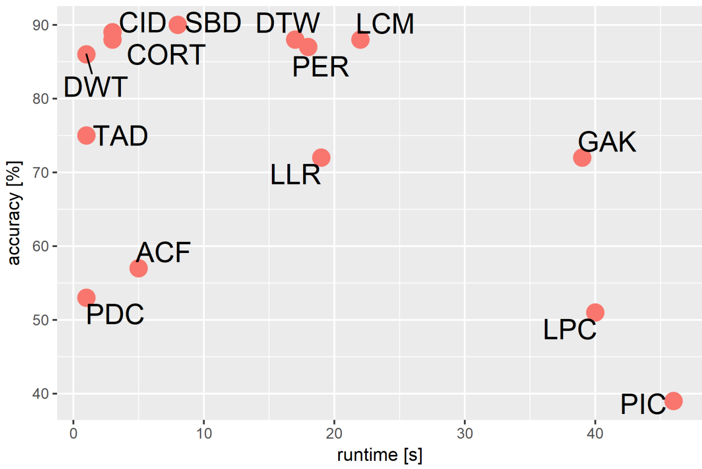



Figure 3.13: Accuracy and speed for each clustering method on the budworms data

In Figure 3.13, it becomes apparent that clustering performance and runtime are not necessarily a trade-off - the best methods also run quite quickly, and the slowest method (PIC) also performs the worst. This is a somewhat surprising result, but should be treated with caution, as clustering performand and speed are highly dependent on the dataset provided, the algorithm implementation, and other factors. As it stands, CID, SBD, DTW, LCM, DWT, CORT, and PER work accurately and quickly for this data set.

## 3.7. Subspace Induction with PRIM

Based on the cluster memberships identified by a particular clustering solution, it is possible to induce the (orthogonal) input subspaces which generate each cluster using a scenario discovery algorithm. PRIM is the *de facto* standard for this. The experiment inputs used to generate the data are plotted in Figure 3.14 as a scatterplot of the two varying inputs $r$ and $r_{step}$, grouped by cluster membership according to the CID clustering solution. For each cluster, an typical subspace an analyst might find with PRIM is overlaid in a corresponding color. CID is chosen because it is the most accurate non-stochastic choice - LCM and SBD can both perform better (as shown in Figure 3.13), but have seed-dependent clustering results. In contrast, the CID solution is reproducible at will.

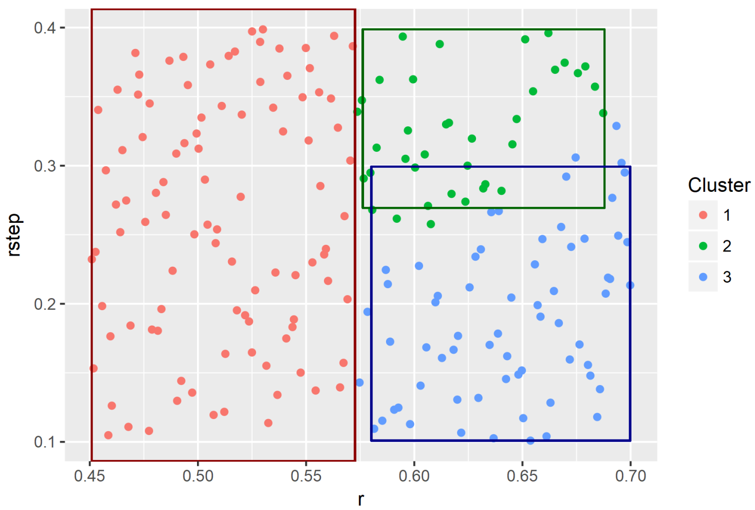



Figure 3.14: PRIM subspace boxes for CID-assigned cluster members, budworms model

In Figure 3.15, the PRIM subspaces derived from the CID clustering solution are overlaid on top of the true cluster memberships. Additionally, the true bifurcation regions analytically derived from the model are underlaid as color-coded polygons. A good overlap between analytical bifurcation regions and induced subspace boxes is observable, indicating that time series clustering may be useful for discovering behavior-based scenarios in complex system models.

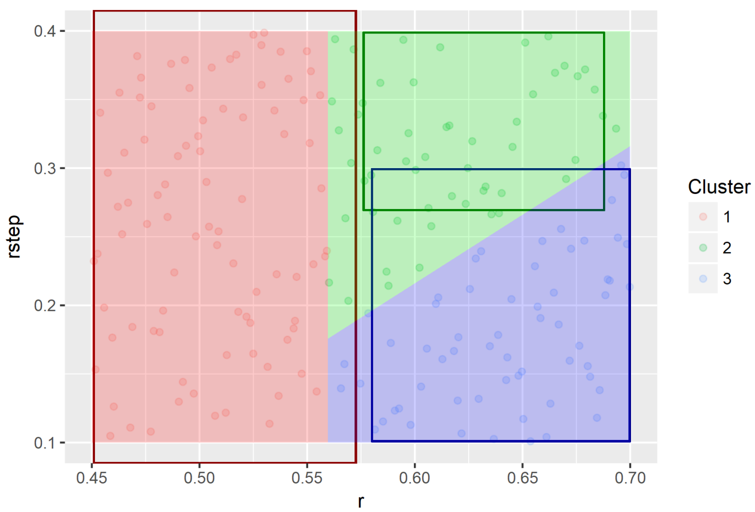



Figure 3.15: Exemplary PRIM subspace boxes for CID, budworms model, over true cluster members (dots) and bifurcation regions (polygons)

As a contrast to the highly accurate CID solution, a poor solution (here: LPC) can be drawn in a similar fashion. The colors are again matched using the Hungarian method, but almost no correlation between the analytical bifurcation regions (underlying polygons) and induced boxes can be observed in Figure 3.17. More importantly, the boxes significantly overlap, reducing their usefulness for policy analysis, since they cannot be clearly separated, as shown in Figure 3.16.

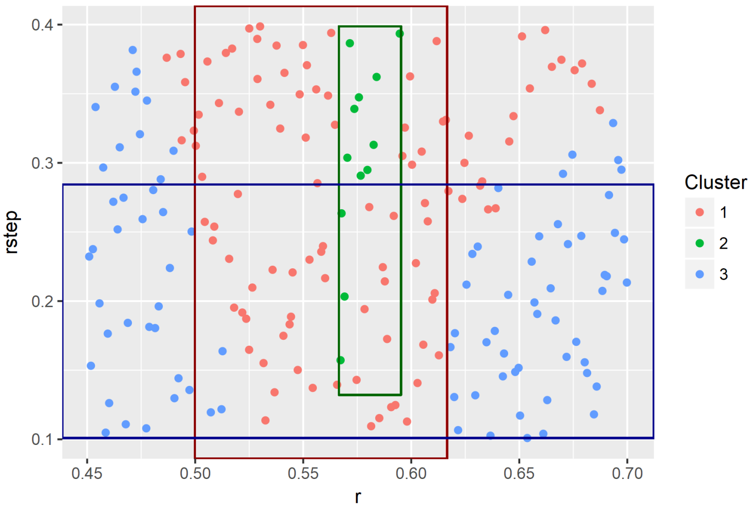



Figure 3.16: Exemplary PRIM subspace boxes for LPC-assigned cluster members, budworms model

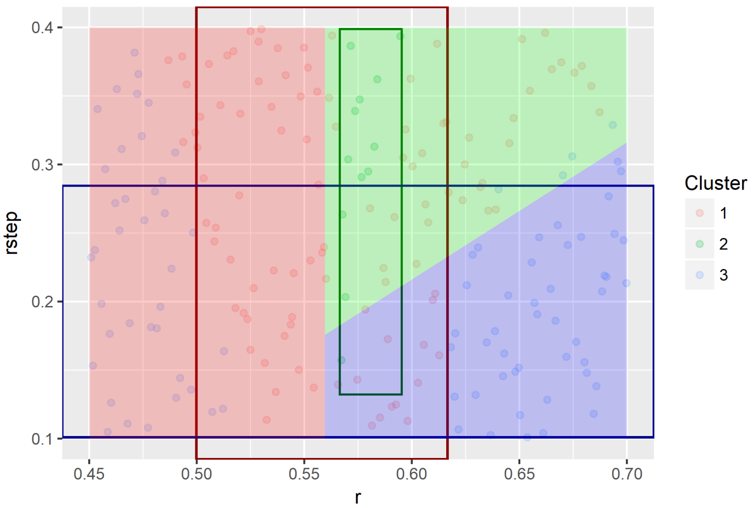



Figure 3.17: Poor performance of PRIM boxes for LPC, budworms model, over true bifurcation regions

## 3.8. Chapter Summary

The spruce budworms population model captures the dynamics of an insect outbreak. It is interesting for behavior-based scenario discovery because it exhibits a cusp bifurcation. With the right cluster count and clustering method, it is possible to find very good approximations of the analytical bifurcation regions in the input space by clustering the model's time series outputs. In other words, the bifurcation behavior of the model can be analyzed without considering the model itself. Methods CID, SBD, DTW, LCM, DWT, CORT, and PER cluster the generated data both quickly and correctly for finding bifurcation regions.

# 4 Brusselator

## 4.1. The Brusselator Chemical Reaction

A second interesting test case for nonlinear behavior is the Brusselator model. The name derives from the place of discovery, Brussels, and the fact that the model represents a chemical oscillator (Strogatz, 1994). It shows a Hopf bifurcation, alternating between linear and oscillating behavior depending on input parameter values, and can be reproduced in real life as a periodically oscillating chemical reaction, one example being the family of Belousov-Zhabotinski reactions (Petrov, Gáspár, Masere, and Showalter, 1993). The model is given as two differential equations representing rates of change in the concentrations of two suitable chemicals $X$ and $Y$, and two supplementary process chemicals $A$ and $B$ as parameters:

$$\dot{X} = A + X^2Y - BX - X$$
$$\dot{Y} = BX - X^2Y$$

The reaction's stable fixed point lies at $X = A$ and $Y = \frac{B}{A}$. However, if $B$ exceeds $1 + A^2$, the Hopf bifurcation occurs and the system transitions into an oscillating behavior, eventually approaching a limit cycle (stable oscillation pattern), shown in Figure 4.1. As with the budworms model, the Brusselator model can be instantiated in a way that a minor variation in one of the parameters triggers the bifurcation and resulting behavior change.

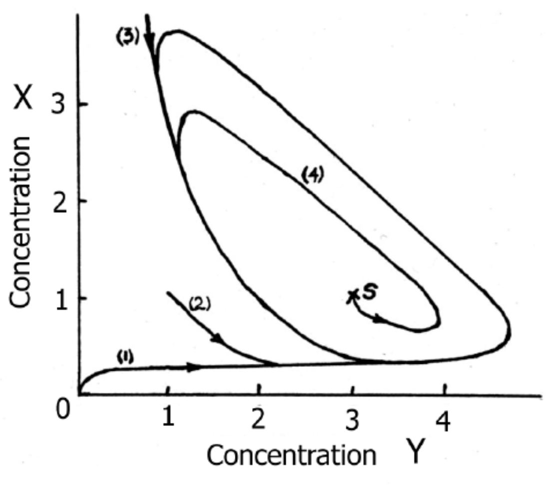



Figure 4.1: Brusselator limit cycle, with four exemplary approaches to the limit cycle

The two-equation form given above can also be translated into a Vensim model. As with the budworms model, the equations consist of minuends and subtrahends, which are split into separate in- and out-flows to respect system dynamics conventions. The resulting model is shown in Figure 4.2. The model is far more complex than the budworms example, showing five separate feedback loops across two stock-flow structures which each represent one dimension of flow. Input variables are again highlighted in green.

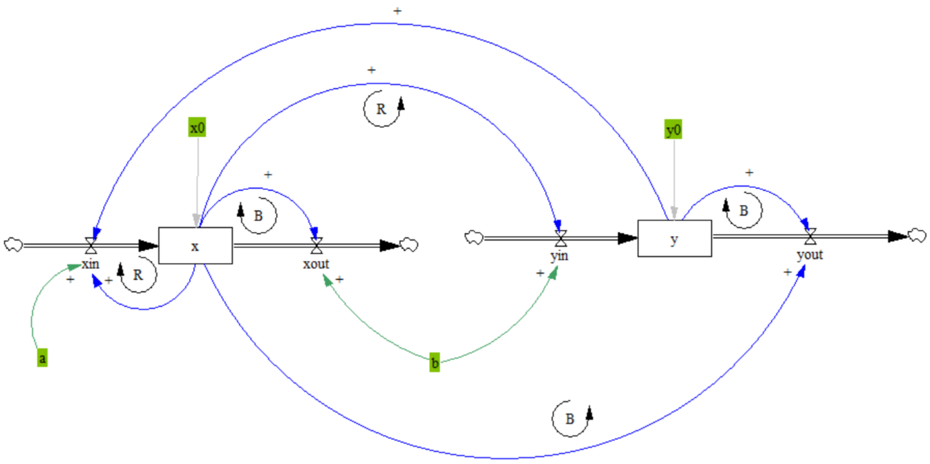



Figure 4.2: Brusselator system dynamics model in Vensim

## 4.2. Behavior Exploration

Using the EMA Workbench, I perform 100 experiments on the described Brusselator Vensim model and store the resulting outputs for analysis in RStudio. The inputs are specified in Table 4.1. $x_0$ and $y_0$ refer to the initial concentrations of the main reagents.

Table 4.1: Brusselator inputs

| variable | min | max | static |
|---|---|---|---|
| $a$ | 0.2 | 2 | |
| $b$ | 1 | 5 | |
| $x_0$ | | | 0.5 |
| $y_0$ | | | 0.5 |

An initial exploratory visualization of the density of reagent $Y$ in Figure 4.3 clearly shows the Hopf bifurcation occurring, as some outputs are linear, while others oscillate wildly. However, some outputs cannot easily be classified - they run apparently linearly, but at an angle, suggesting they in fact may be oscillating with a very low frequency.

## 4.3. Number of Clusters

Silhouette widths are again used to compare different cluster counts $k$ across the different methods.

Unlike before, no clearly dominant suggestions for $k$ are apparent apart from $k = 2$ (the ground truth). To explore a more varied clustering behavior, I therefore choose $k \in \{2, 4, 6\}$ as candidate solutions, as these values seem to create the most "dissent" between the different methods based on the positive and negative spikes apparent in Figure 4.4.

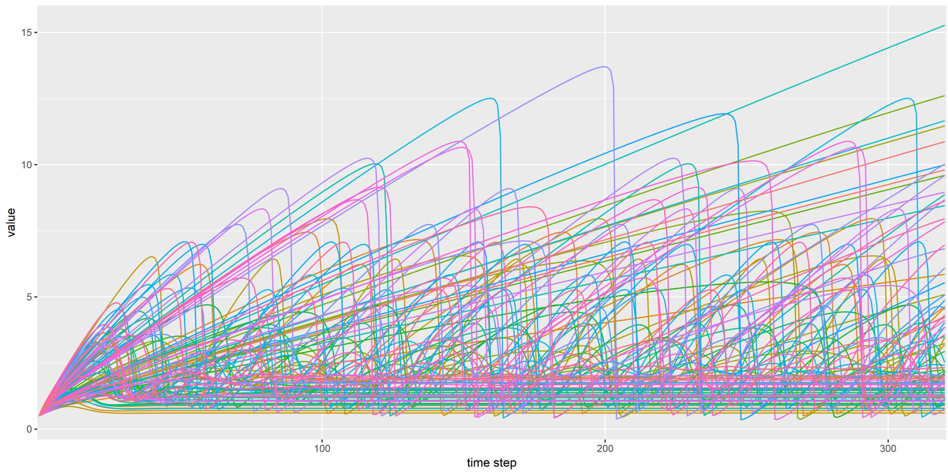



Figure 4.3: 100 experiments on the Brusselator model with varying $a$ and $b$

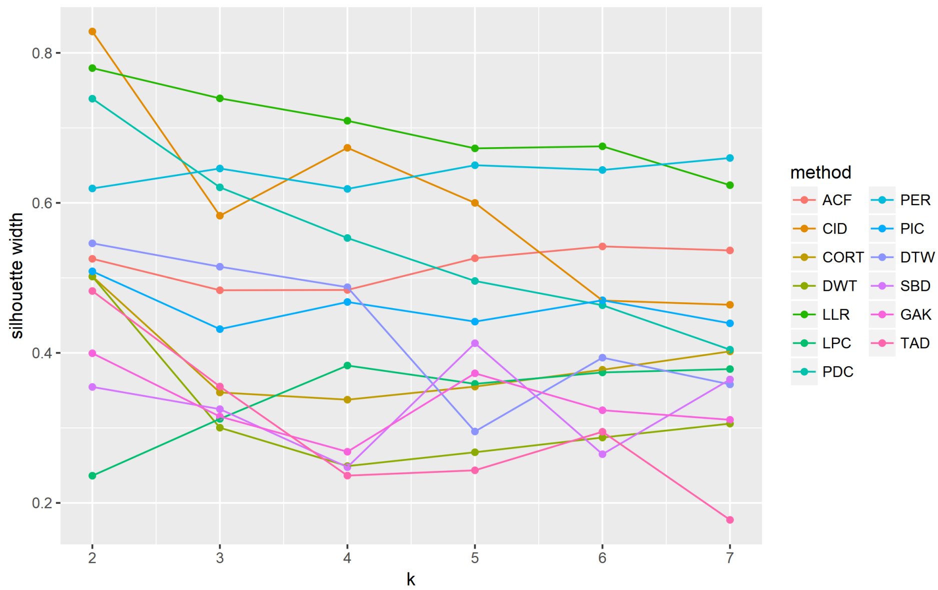
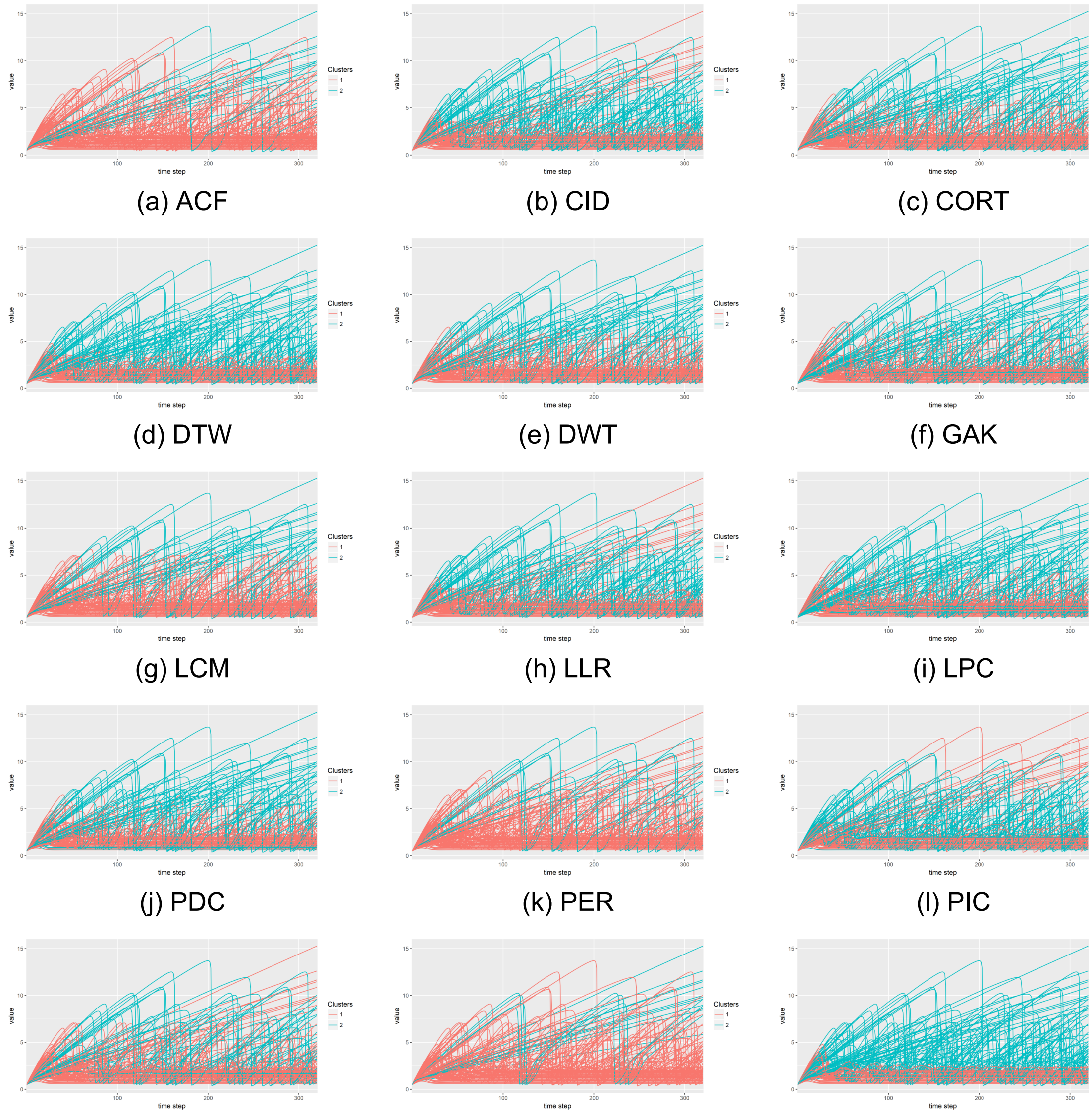



Figure 4.4: Silhouette widths for different clustering methods, Brusselator model

# 4.4. Clustering Solutions

Now that candidates for $k$ are available, I again compare clustering solutions methods for each $k$ option. For $k = 2$, the ground truth is used to Hungarian-match the cluster indices, for $k = 4$, I use the LCM solution as in the budworms model, and for $k = 6$, I use the CID solution, as the LCM clusterer would not converge for the given data and parameters.




(a) ACF (b) CID (c) CORT

(d) DTW (e) DWT (f) GAK

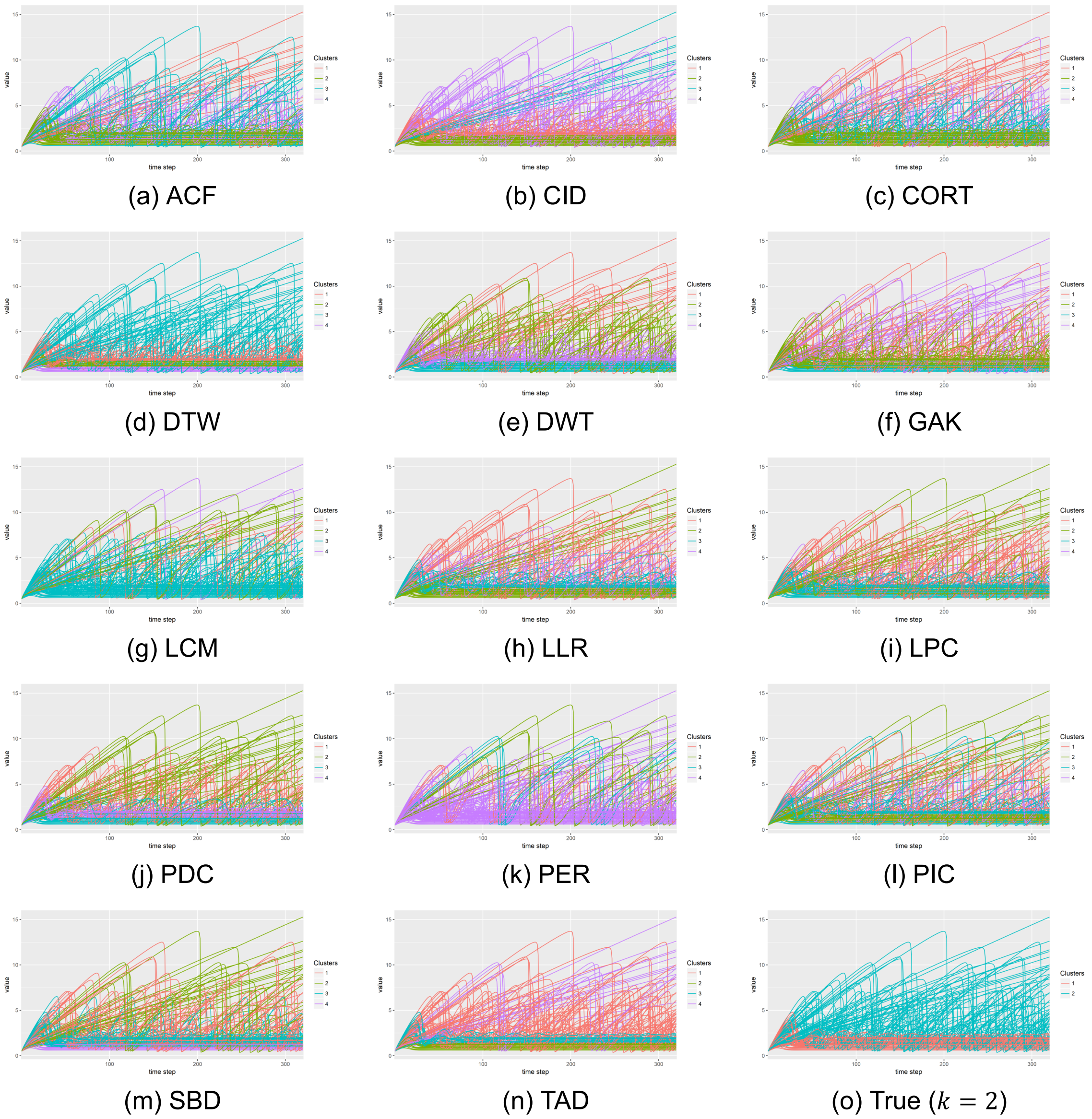
(g) LCM (h) LLR (i) LPC

(j) PDC (k) PER (l) PIC

(m) SBD (n) TAD (o) True ($k = 2$)

Figure 4.5: Different Brusselator clustering solutions for $k = 2$, including true clusters

While a simple linear-oscillating bifurcation seems like a straightforward clustering task, a wide variety of solutions is apparent from Figure 4.5. Low-amplitude oscillations seem likely to get lumped in with the linear time series. The fate of the low-frequency oscillations is notable - while e.g. CID and LLR cluster them with the linear outputs, LCM or DTW consider them closer to the oscillations since they reach upper-range values.



(a) ACF (b) CID (c) CORT

(d) DTW (e) DWT (f) GAK

(g) LCM (h) LLR (i) LPC

(j) PDC (k) PER (l) PIC

(m) SBD (n) TAD (o) True ($k = 2$)

Figure 4.6: Different Brusselator clustering solutions for $k = 4$, including true clusters

Despite the index-matched clusters, even a brief visual inspection shows how diverse the clustering solutions for $k = 4$ in Figure 4.6 are. A closer inspection reveals that the clusters do make intuitive sense, even though they are hard to visually distinguish. Using the CID solution (enlarged as Figure 4.7) as an example, the clusters can easily be qualitatively described:

- red cluster 1 holds low-amplitude oscillations
- green cluster 2 holds linear/stable time series
- blue cluster 3 holds quasi-linear but gaining outputs
- purple cluster 4 holds high-amplitude oscillating outputs

This does not mean that this clustering is correct, merely that it is internally and intuitively consistent. However, the fact that narratives for each cluster can easily be generated bodes well for scenario-based policy analysis, where narratives are often connected with model outputs (Bryant and Lempert, 2010).

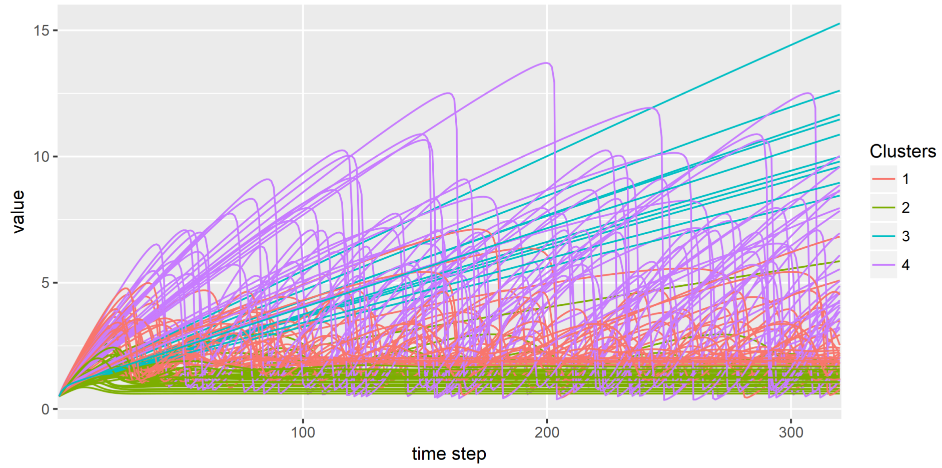
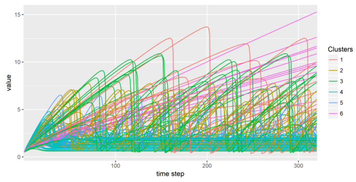
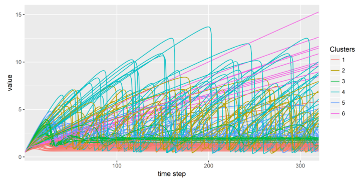


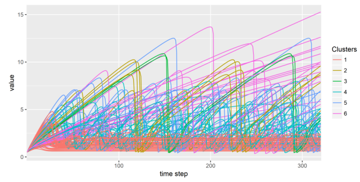

Figure 4.7: CID solution for Brusselator model, $k = 4$



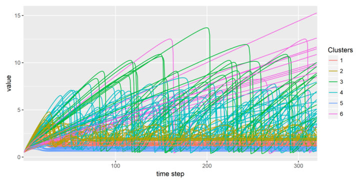
(a) ACF



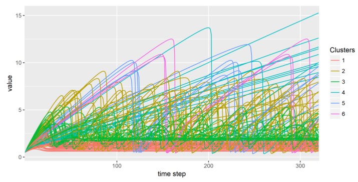
(b) CID



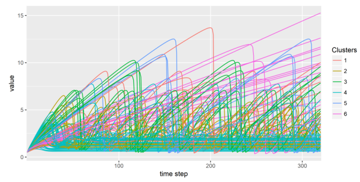
(c) CORT



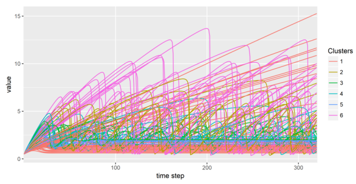
(d) DTW



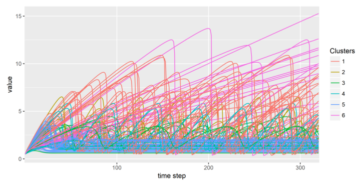
(e) DWT



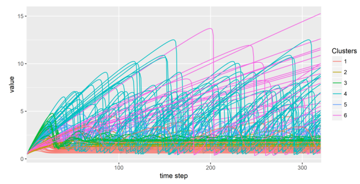
(f) GAK



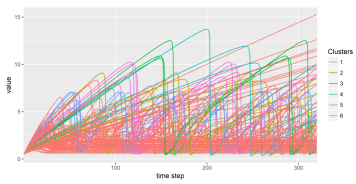
(g) LLR



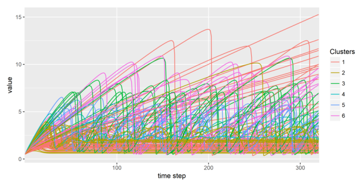
(h) LPC



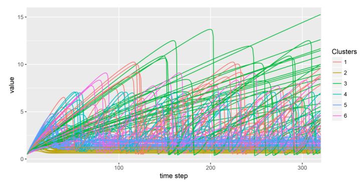
(i) PDC



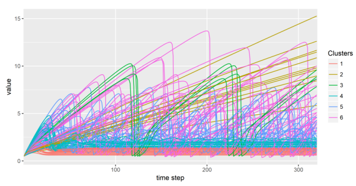
(j) PER



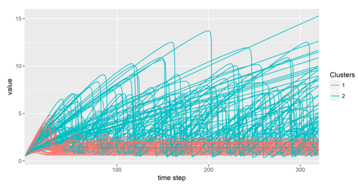
(k) PIC



(l) SBD



(m) SBD



(n) True ($k = 2$)

Figure 4.8: Different Brusselator clustering solutions for $k = 6$, including true clusters

In Figure 4.8, clustering solutions are presented for $k = 6$ and all methods except LCM. I could not get the LCM clustering algorithm to reliably converge, which I attribute to the specific implementation in Helske and Helske (2017) rather than a fundamental issue with latent class Markov modeling. As before, many of the clustering solutions can be qualitatively and intuitively described. I draw again upon the CID solution, shown enlarged in Figure 4.9):

- red cluster 1 holds linear outputs with little initial goal-seeking behavior
- gold cluster 2 holds medium-amplitude oscillations
- green cluster 3 holds linear outputs with noticable initial goal-seeking
- aquamarine cluster 4 holds high-amplitude oscillations
- blue cluster 5 holds low-amplitude oscillations
- pink cluster 6 holds quasi-linear but gaining outputs

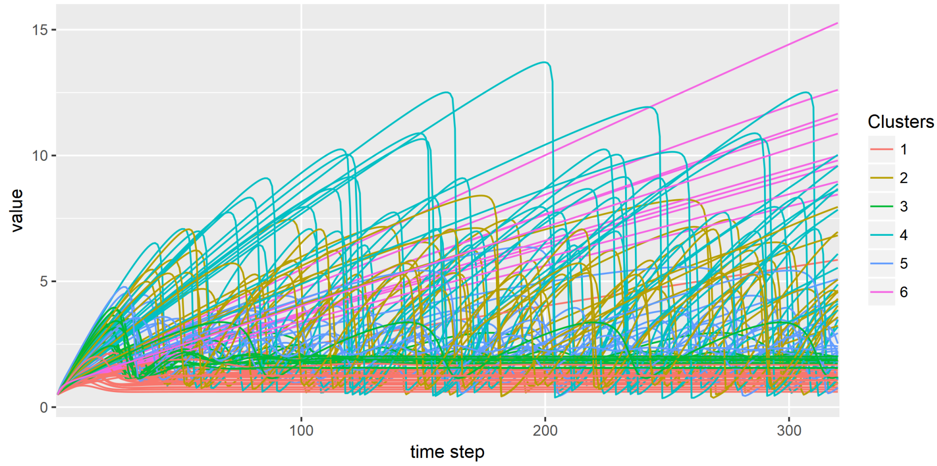



Figure 4.9: CID solution for Brusselator model, $k = 6$

## 4.5. Clustering Validation

Using the analytical bifurcation regions of the Brusselator model as external knowledge, the $k = 2$ clustering solutions can again be validated using confusion matrices.

From Figure 4.10, it becomes apparent that actual cluster 1 members are almost never attributed to cluster 2 (that is, linear time series are almost never grouped with oscillating ones), but some cluster 2 members are assigned to cluster 1 - oscillating time series, likely edge cases with low amplitudes and/or frequency, are considered similar to linear outputs.

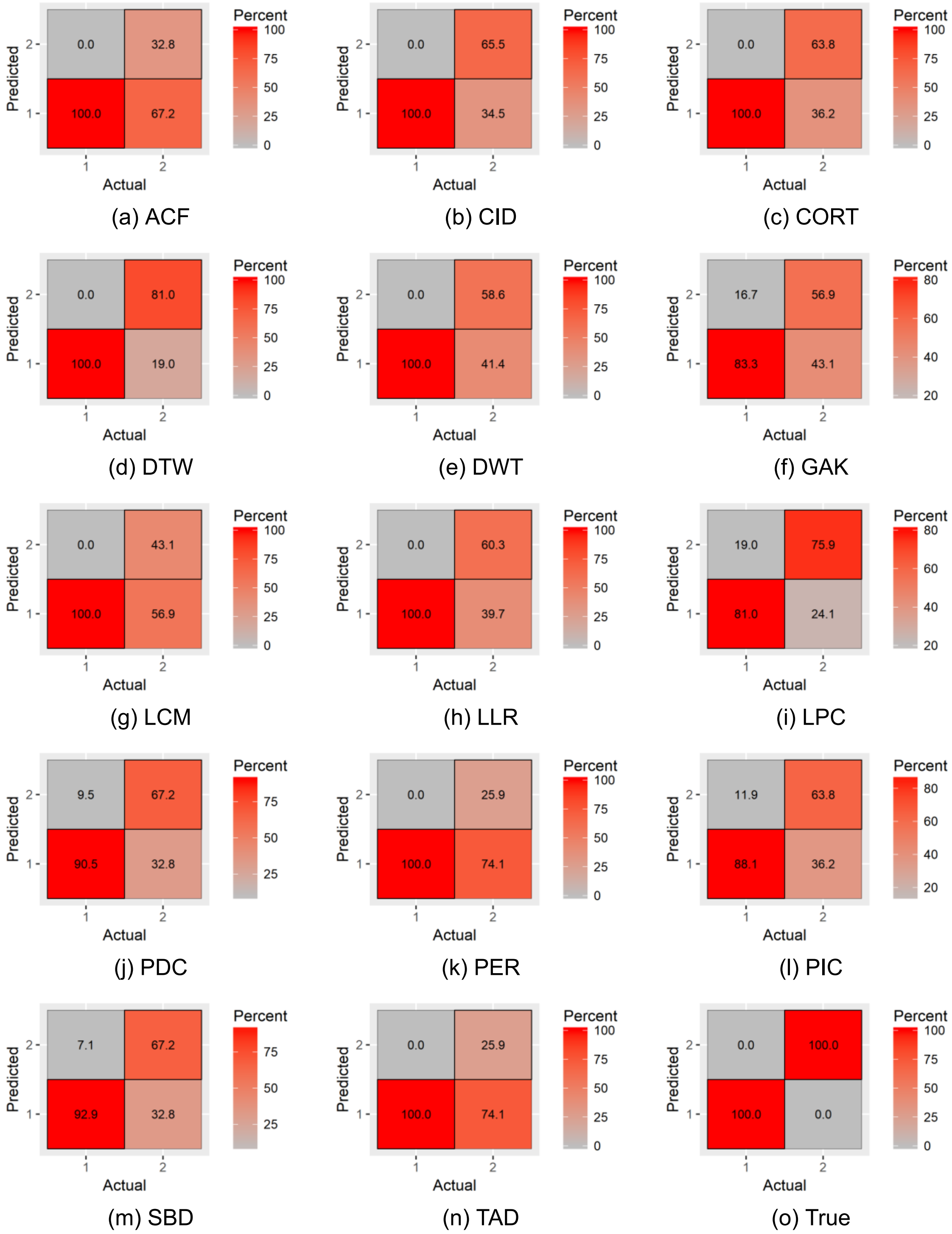



Figure 4.10: Confusion matrices for clustering solutions vs. true cluster members, $k = 3$

## 4.6. Performance Comparison

From the confusion matrices, clustering performance can be deduced and compared. Figure 4.11 plots the percentage of correctly identified cluster members for each method.

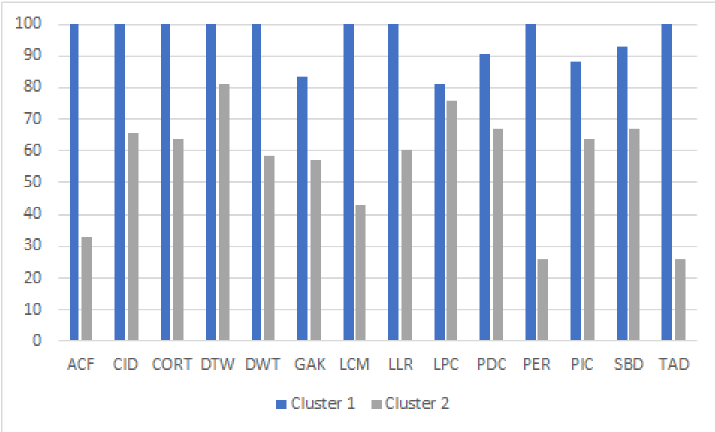



Figure 4.11: Clustering performance for each method, Brusselator model

For this specific combination of data and clustering parameters, it appears that CID, CORT, DTW, PIC and SBD perform reasonably well (>60% correct for both clusters).

In Figure 4.12, each method's clustering accuracy for the $k = 2$ case and the time it took to complete the clustering are compared in a scatter plot.

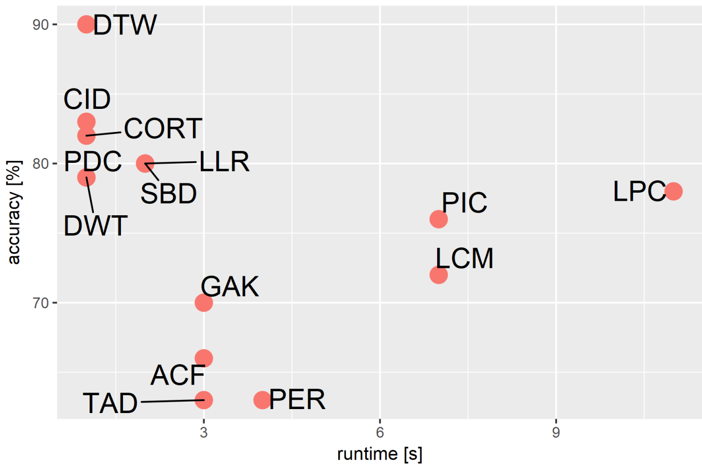



Figure 4.12: Accuracy and speed for each clustering method on the Brusselator data

Once again, there is no apparent trade-off between speed and accuracy, as the best-performing methods are also the fastest ones. I note once again that this insight should be treated with caution, as it is highly dependent on the dataset and implementation. For the Brusselator-generated data set, DTW, CID, CORT, PDC, DWT, SBD and LLR seem to work both quickly and accurately in finding bifurcation regions.

Interestingly, some methods perform well for both the budworms and Brusselator data sets. Specifically, I identify CID, CORT, DWT, DTW and SBD as high-performing methods for both cases, based on Figures 3.13 and 4.12.

## 4.7. Subspace Induction with PRIM

Subspace induction using the PRIM is interesting in the Brusselator case because bifurcation analysis reveals that the border between the two bifurcation regions is curved. In the budworms model, the borders were linear, which lends itself to the induction of orthogonal boxes.

In Figure 4.13, I plot the experiment inputs for the Brusselator model as a scatter plot, color the points by assigned DWT cluster membership, and overlay the induced PRIM boxes. I choose DWT for this because its bifurcates the input space along a similar (if slightly offset) curve to the true bifurcation regions.

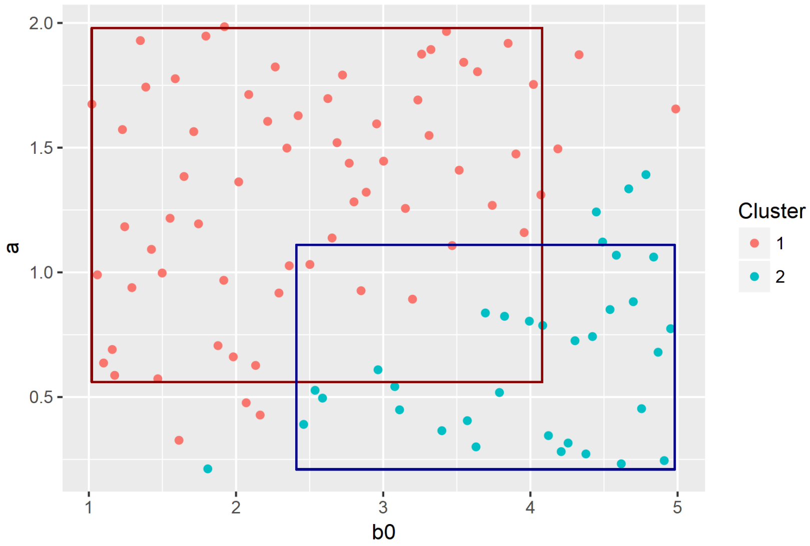



Figure 4.13: Exemplary PRIM subspace boxes for DWT-assigned cluster members, Brusselator model

In Figure 4.14, the PRIM subspaces derived from the DWT clustering solution are overlaid on top of the true cluster memberships. Additionally, the analytically derived Hopf bifurcation region is underlaid as a blue polygon.

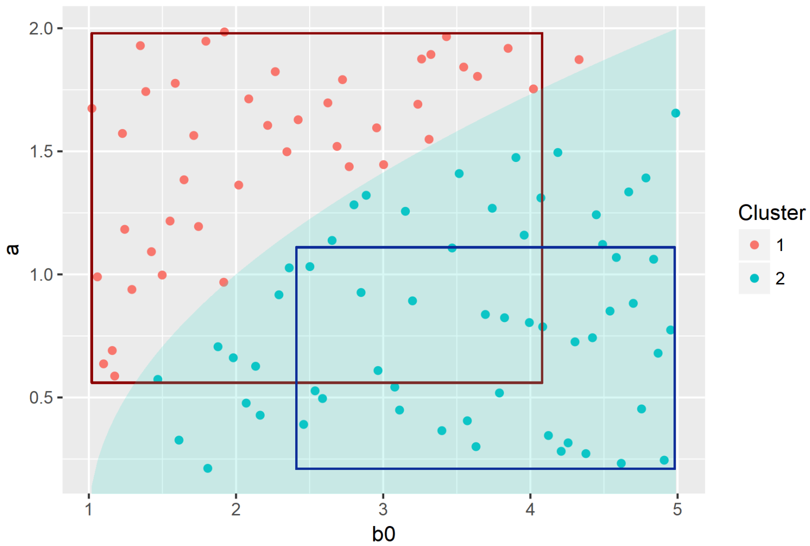



Figure 4.14: Exemplary PRIM subspace boxes for DWT, Brusselator model, over true cluster members and bifurcation regions

From Figure 4.14, it becomes apparent how PRIM reaches limitations when input subspaces are not orthogonal. Two flaws are noticeable. Firstly, the blue induced box for cluster 2 has poor coverage, almost half the relevant points are outside the box. However, increasing the box size would make it extend beyond the bifurcation region it is supposed to represent. Secondly, the red and blue induced boxes overlap quite significantly. This indicates that caution is required when using the box for policy analysis. The issues surrounding overlapping boxes will be explored in more detail later on.

## 4.8. Chapter Summary

The Brusselator is a model of a chemical reaction which shows a Hopf bifurcation - it can either have linear or oscillating behavior depending on input parameters. With the right choice of clustering method and cluster count, reasonable approximations of the analytical bifurcation regions in the input space can be found by clustering the model outputs. However, the non-orthogonal bifurcation regions are difficult to find with current rule induction methods. CID, CORT, DWT, DTW and SBD are the clustering methods which perform well for both the budworms and Brusselator models.

# 5
# Shale Gas

Both the budworms population and the Brusselator chemical reaction models are simple systems, with clearly observable nonlinearities, well-understood bifurcation behavior and generally low complexity. However, modern complex system models used in policy analysis are generally significantly larger and less well-defined (Davis et al., 2007). It is therefore necessary to test time series clustering as a method for finding decision-relevant input subspaces through time series clustering on such a complex model.

## 5.1. The Shale Gas Model

In 2014, Auping, de Jong, Pruyt, and Kwakkel introduced a comprehensive model of global energy developments, particularly the shale gas revolution, coupled with a global model of economic development and nation state stability. The model was also discussed or used in various other works, namely Auping, Pruyt, de Jong, and Kwakkel (2016); de Jong, Auping, and Govers (2014); Moorlag, Auping, and Pruyt (2014).

I consider this work a good representation of a modern complex system model as it:

- is widely scoped (large number of variables covering a diverse range of system aspects).
- includes significant feedback effects.
- features both parametric and structural uncertainty.
- generates a variety of behaviors over time.
- concerns a grand challenge of global proportions.

I was grateful to receive the main author's original data sets and models for this thesis. While the model itself is too complex to usefully explain here, the (highly aggregated) causal loop diagram of the main energy submodel, shown in Figure 5.1, may be indicative of the complexity considered.

A significant number of feedback loops are apparent. Behind each boxed variable lies a submodel of between perhaps 20 and 100 variables, and perhaps 5 to 30 feedback loops. Additionally, the model is subscripted across multiple regions and energy types, and also has a number of inputs that govern structural changes in the model itself.

## 5.2. Behavior Exploration

As the data was already supplied in the form of 2000 simulation runs (generated through a Vensim model and the EMA Workbench), initial exploration is as easy as visualizing the data over time. Specifically, I choose to visualize the future price of oil (*Oil Price [$/BBTU]* in the model), as it seems to be the indicator with the most interesting and varied behavior across experiments, making it an ideal candidate

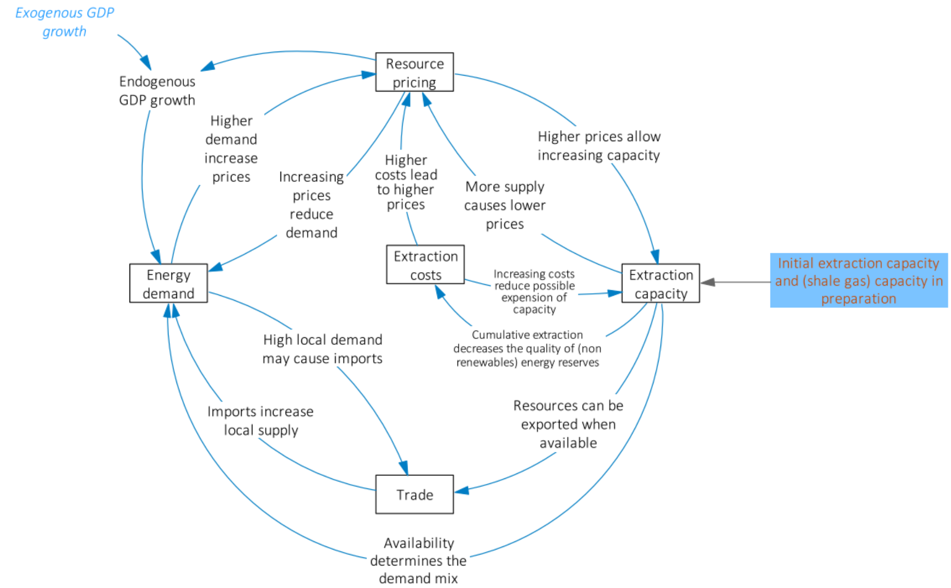



Figure 5.1: Causal loop diagram of Auping's shale gas model

for testing time series clustering. All work performed in this chapter will relate exclusively to this outcome of interest. To improve figure quality, I randomly select 500 of the 2000 stored experiments. As the experiments were performed under Latin Hypercube sampling, randomly selecting from amongst them should have no impact on the representativeness of Figure 5.2.

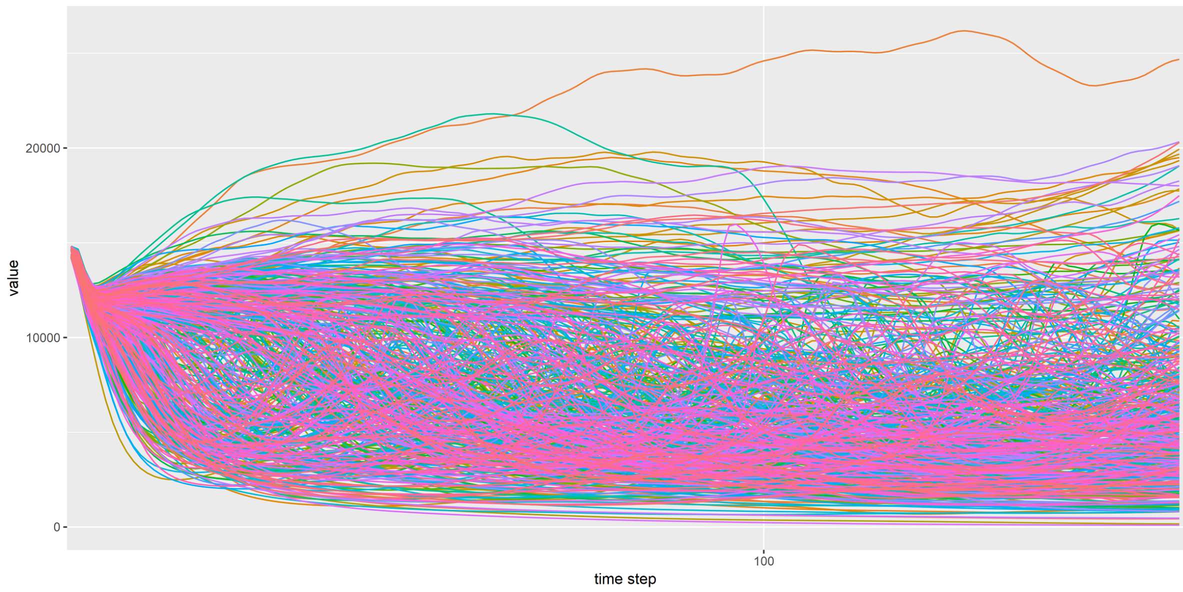



Figure 5.2: 500 randomly selected traces of the indicator "Oil Price [$/BBTU]"

Unlike the simple test cases presented before, no apparent clusters can be recognized. Likewise, no prototypical behaviors over time can be deduced. Line colors are randomly assigned.

## 5.3. Number of Clusters

To consider possible cluster counts, I again use the cluster silhouette width. For the five methods found to perform best in the previous chapters (CID, CORT, DWT, DTW and SBD), I examine the silhouette widths for every $k$ value between 2 and 50, with a step size of 2. This values are chosen somewhat arbitrarily, as I have no indication whatsoever regarding potentially suitable cluster counts. The only indication comes from the original paper, which shows that structural uncertainty switches are used. This might mean that $k$ values are likely to be a multiple of 2, as Vensim switches are generally binary. Since DTW and SBD are stochastically influenced clustering methods, I execute three separate runs for each one to reduce the influence of random seeding.

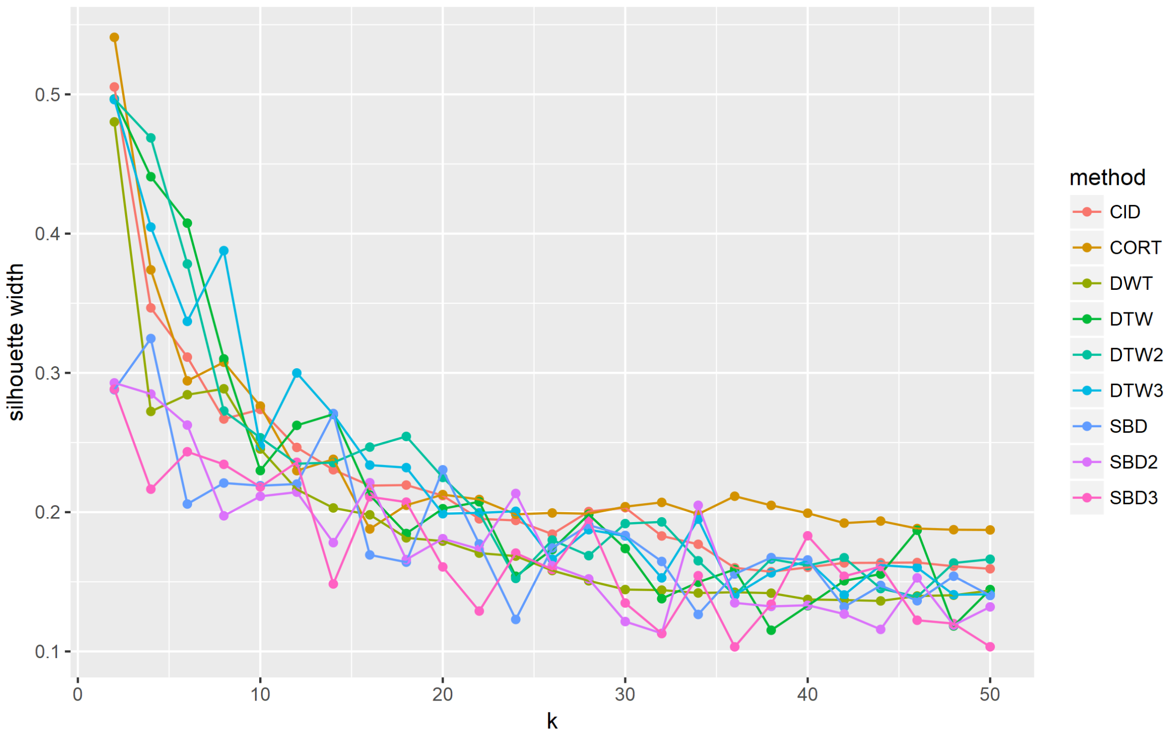



Figure 5.3: Silhouette widths for different clustering methods, shale gas model

As expected (or feared), no clear recommendation for $k$ emerges, apart from $k = 2$. However, a sharp elbow around $k \approx 12$ is apparent, with silhouette width rapidly stabilizing beyond that value. The stochastically influenced trajectories of DTW and SBD are also apparent. Other cluster validity indices might reveal more useful cluster count advice, but are not considered here for time reasons.

## 5.4. Clustering Solutions

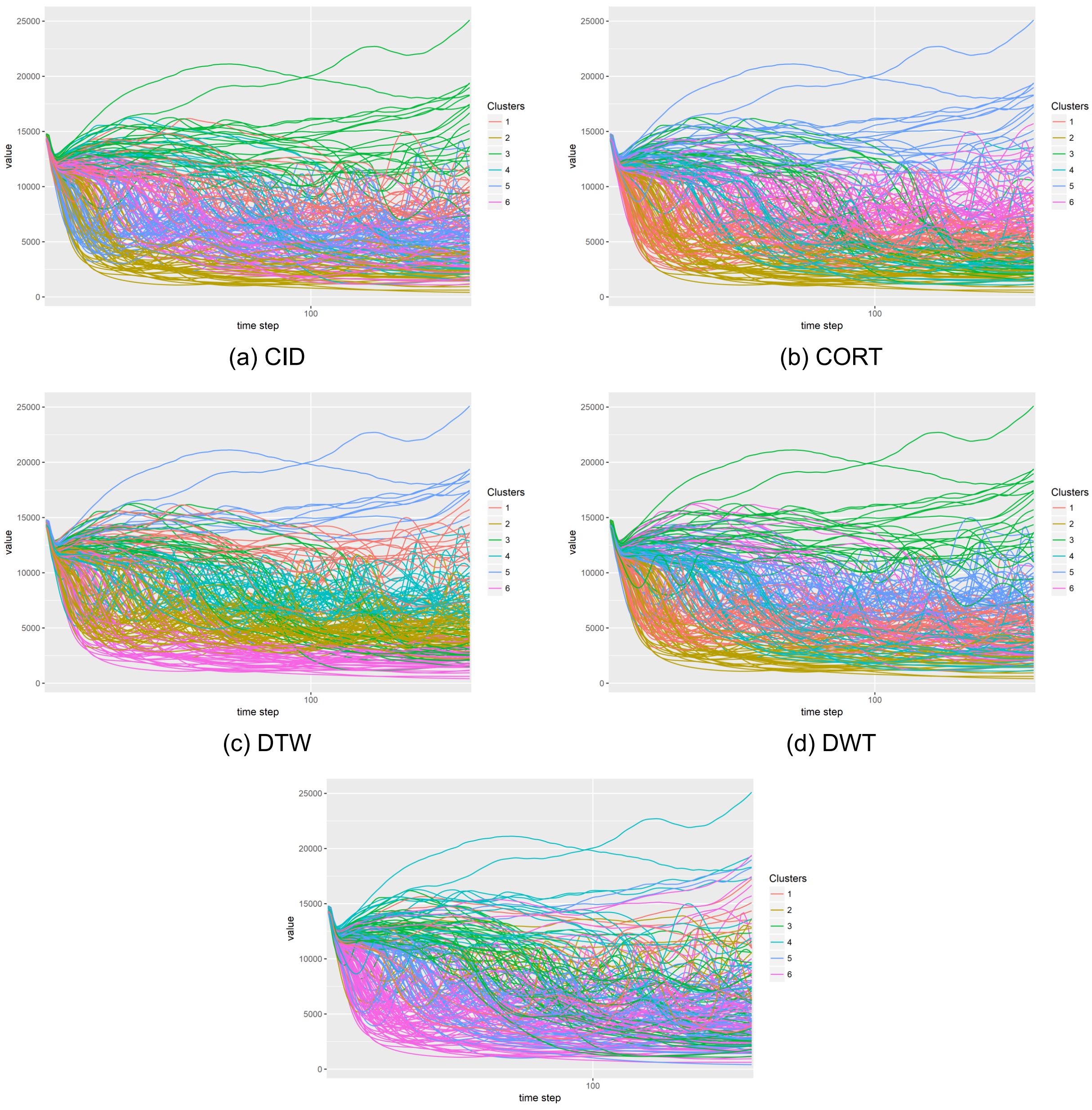
As the silhouette width does not identify a useful $k$ value, I (somewhat arbitrarily) use $k = 6$ in the following analysis. I feel this value will provide a good balance between generating meaningful clusters, containing enough time series to be representative of greater trends in the model, and being visually interpretable. To further improve figure interpretability, I use only 200 of the 2000 stored experiments for clustering exploration. I select experiments $[800 : 1000)$ for this, as they contain a nice diversity of behaviors, and two ”black swan” runs. I use the CID solution as reference for cluster index matching. Both the choice of $k$ and the reduction in considered runs should not influence the methodological application of time series clustering for scenario discovery, although the quality of the induced PRIM boxes may suffer if the input space is not populated densely enough.

Visual inspection of the clustering solutions indicates that every method seems to generate consistent clusters. However, it is difficult to clearly associate cluster members with one another. The two black swan runs are clearly visible, achieving values (read: global oil prices!) well above any other experiment.

By separating each figure into constituent clusters, a much clearer overview of cluster memberships can be achieved. The cluster colors are aligned with the color scheme used in Figure 5.4.

While the clustering solutions depicted in Figures 5.5, 5.6, 5.7, 5.8, and 5.9 all appear very different from each other, they seem to be internally consistent. Some details are noticeable. Two of the SBD clusters are much smaller in comparison, with less than a dozen members each. For DTW, three clusters are clearly smaller than the other three. However, DTW does place the two black swans identified earlier in a separate cluster together with other high-value time series. This may be an attractive solution for policy analysts who require special treatment of outliers. By contrast, SBD lumps the two black swans in with a number of other runs that do not intuitively appear similar.

Overall, it can be said that all five methods can usefully partition the provided time series into internally behaviorally consistent clusters, although those clusters differ wildly between methods. However, none of the clustering solutions appears intuitively better than the others.




(a) CID (b) CORT

(c) DTW (d) DWT

(e) SBD

Figure 5.4: Different clustering solutions for $k = 6$, shale gas model

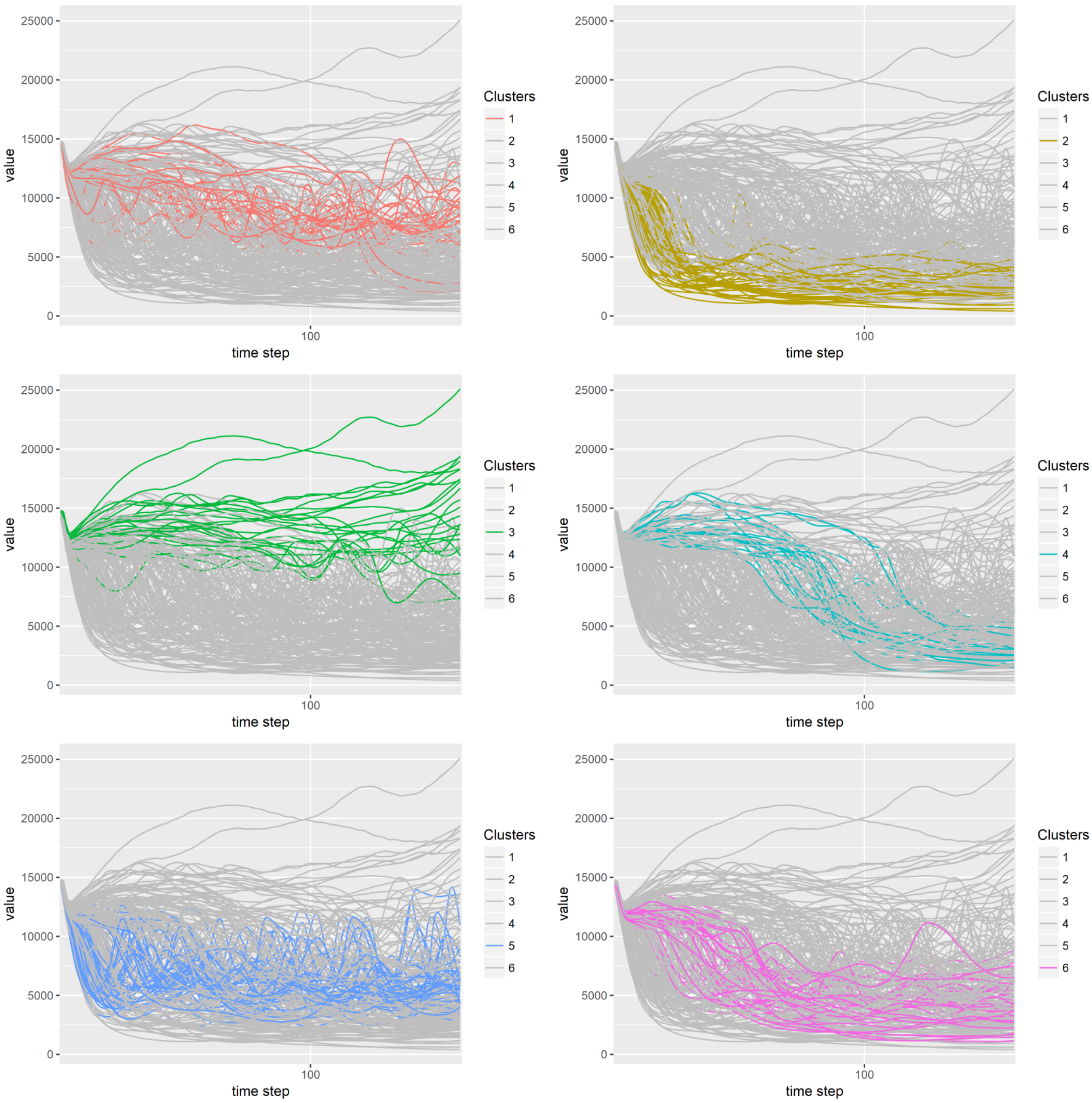



Figure 5.5: CID cluster members, shale gas model

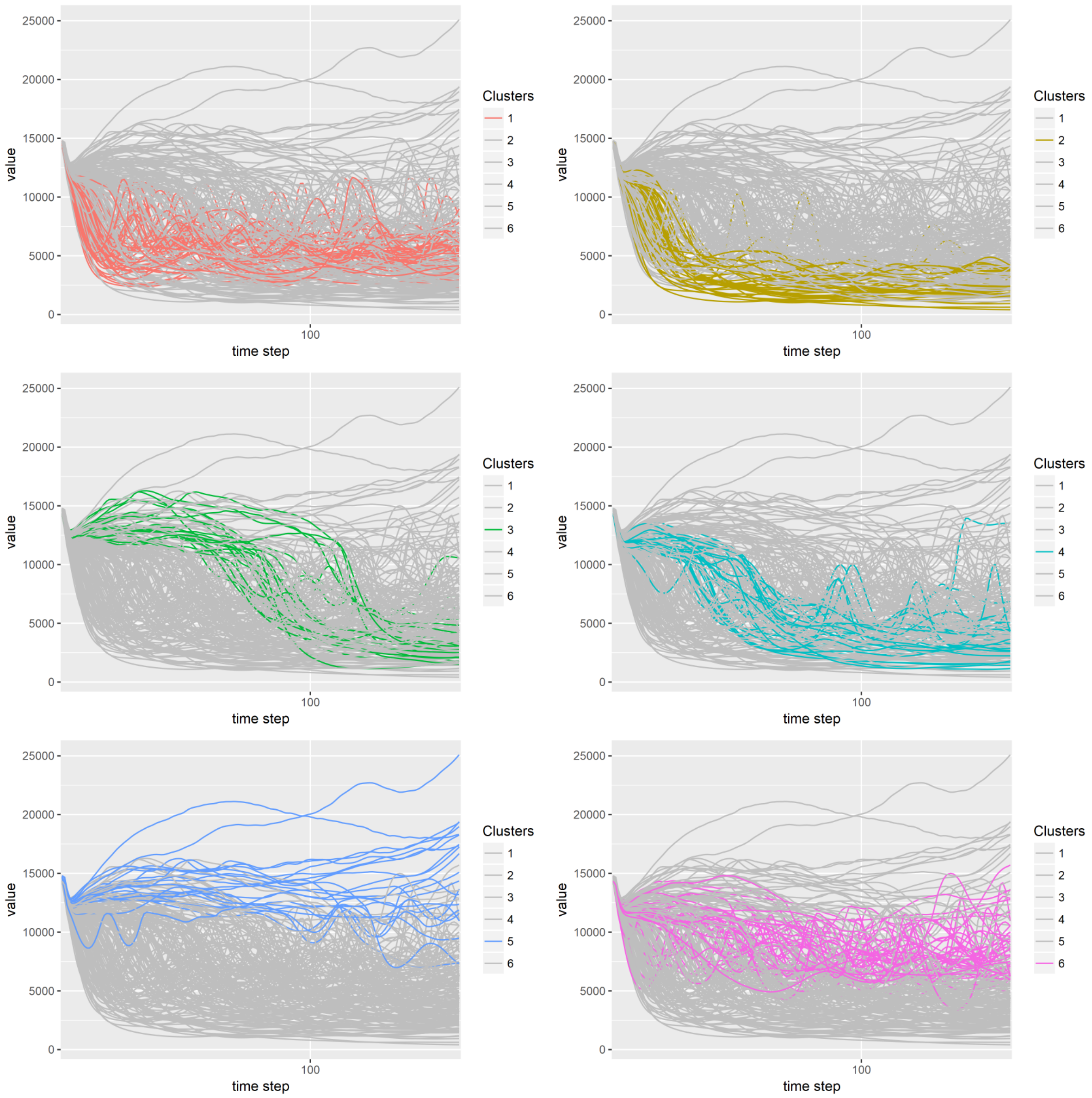



Figure 5.6: CORT cluster members, shale gas model

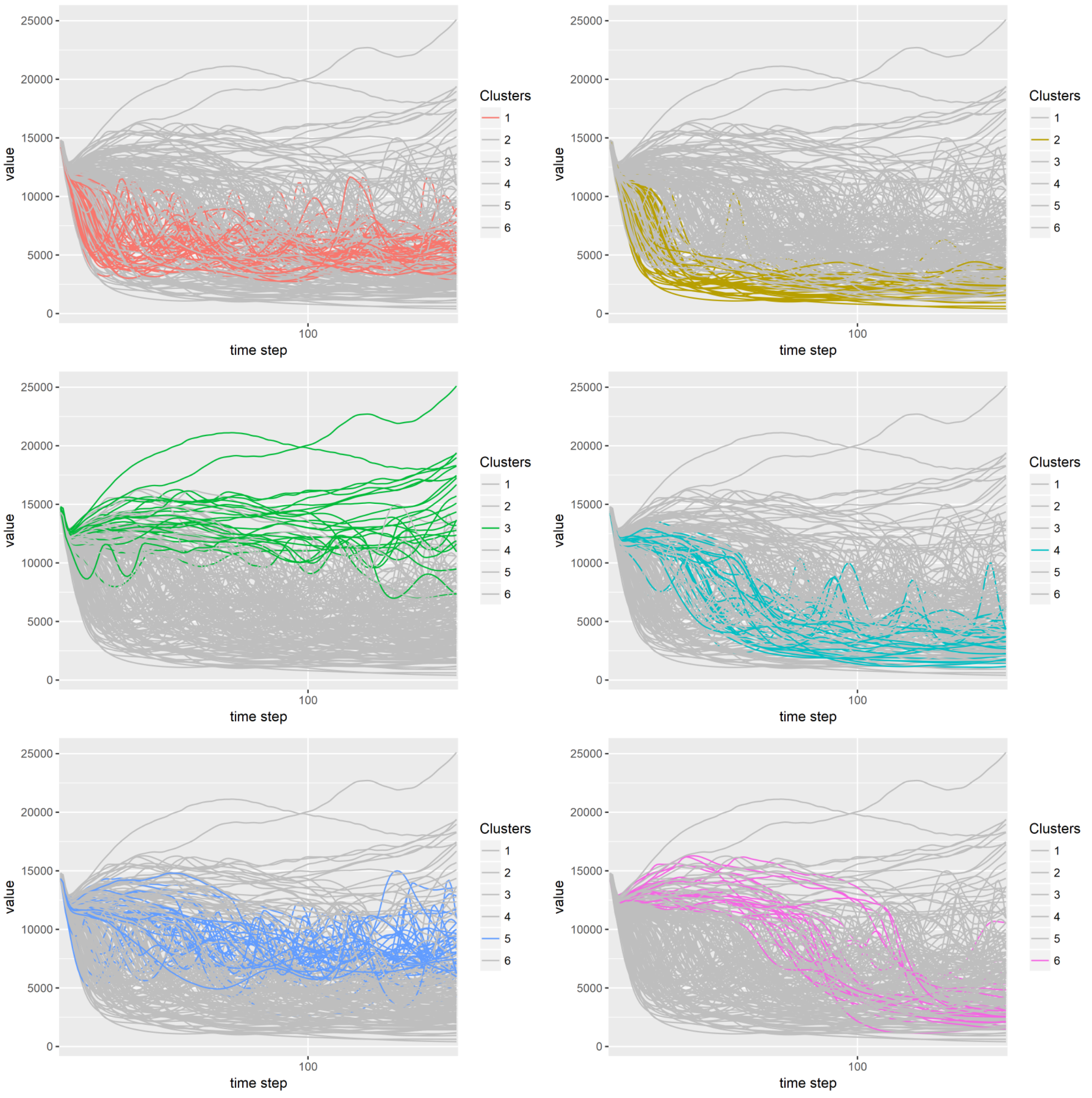



Figure 5.7: DWT cluster members, shale gas model

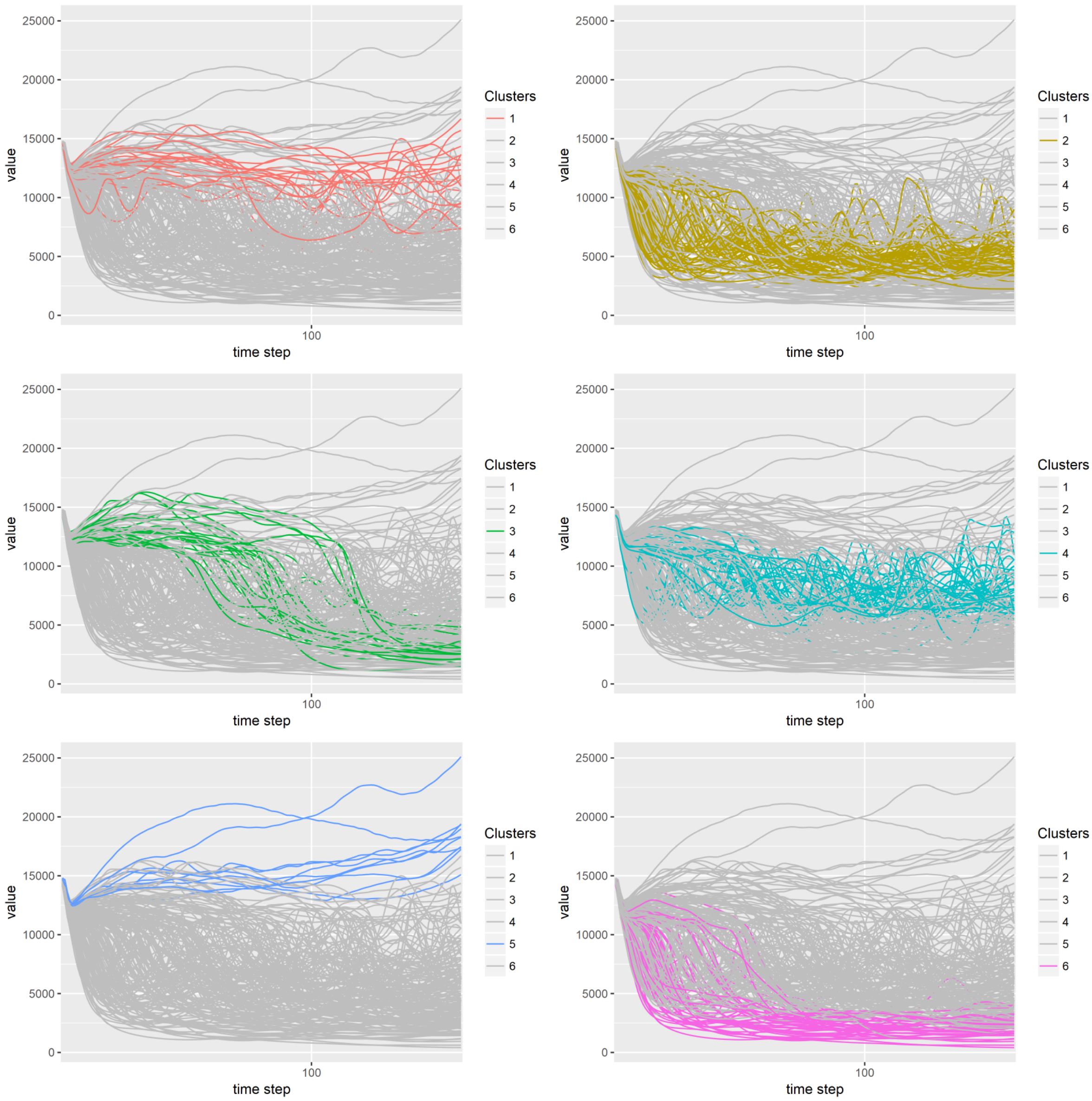



Figure 5.8: DTW cluster members, shale gas model

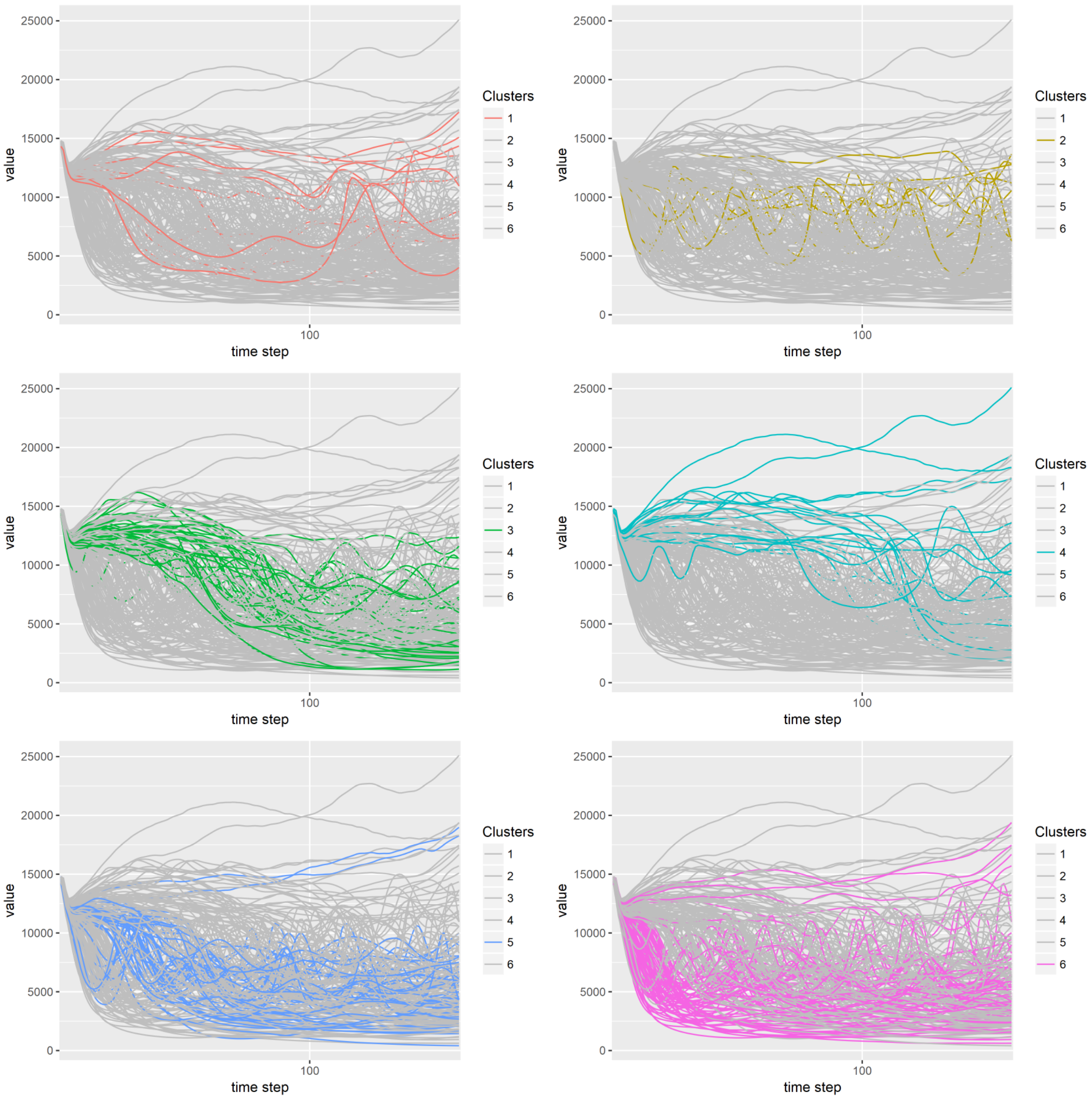



Figure 5.9: SBD cluster members, shale gas model

## 5.5. Subspaces Induction with PRIM

To demonstrate that PRIM boxes can also be induced for more complex models based on clustered time series, I re-run the CID clustering method for all 2000 experiments, shown in Figure 5.10. I choose CID here because the algorithm showed good accuracy in the two previously discussed models, runs very quickly (>3 minutes to cluster 2000 series of 160 data points) and produces clusters of roughly equal size. If the goal was to find true bifurcation regions, comparable cluster size could not be expected, but since 6 clusters are unlikely to represent true bifurcation regions, comparable cluster sizes may make analysis easier.

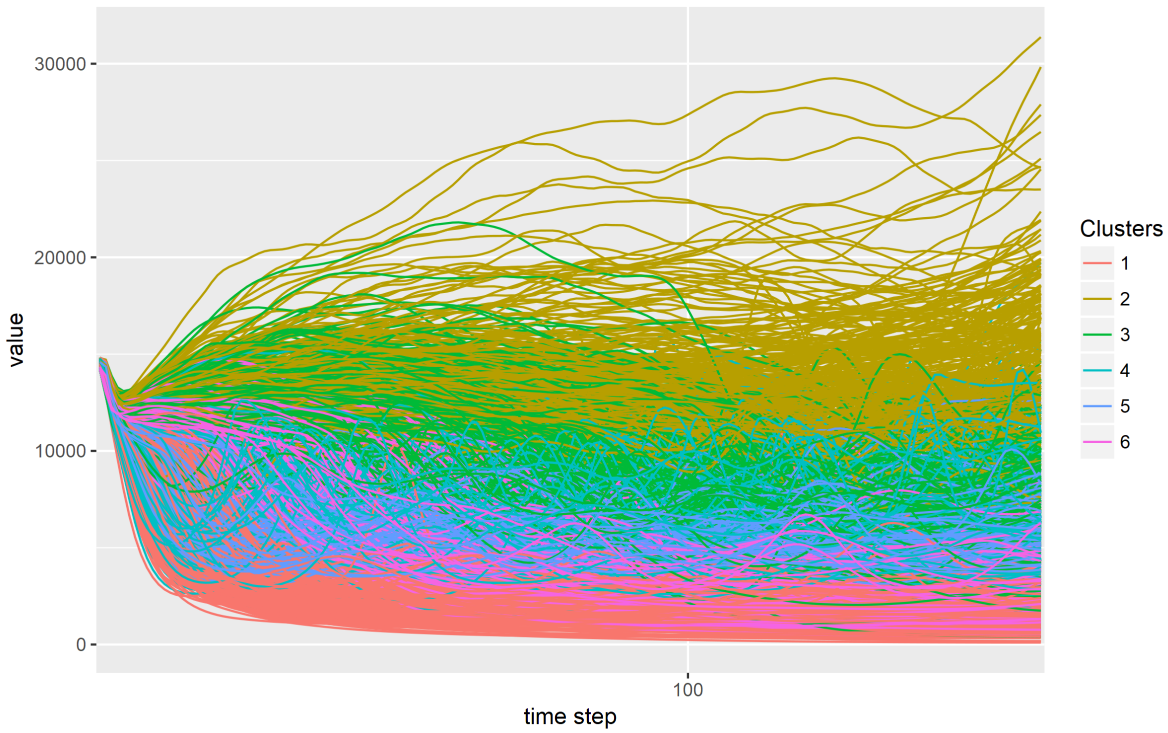



Figure 5.10: CID clustering solution, 2000 runs, shale gas model

Using extra runs increases the point density across the input space, and therefore should make PRIM easier to run. As the output space is now even more densely populated, I again split the plot into constituent time series, shown in Figure 5.11. As the clustering has been updated and differently-sized clustering solutions cannot be matched using the Hungarian method, the colors no longer correspond to the solution presented in Figure 5.5.

Based on the cluster constituents, cluster narratives can again be created. For Figure 5.11, these might be as follows:

- red cluster 1: oil price rapidly declines, and stays low.
- gold cluster 2: oil price continually increases.
- green cluster 3: oil price remains roughly constant, but can vary significantly temporally.
- aquamarine cluster 4: oil price declines somewhat, but shows significant oscillations.
- blue cluster 5: oil price declines noticeably, but rapidly reaches a new equilibrium.
- pink cluster 6: oil price slowly declines, eventually reaching quite low levels.

From a policy perspective, it might appear that behaviors 2 and 4 are the least desirable, as they are likely to upset the global economy quite significantly (increasing or rapidly fluctuating oil prices). It could be argued that behavior 1, a rapid drop in oil price, is also undesirable from an economic stability

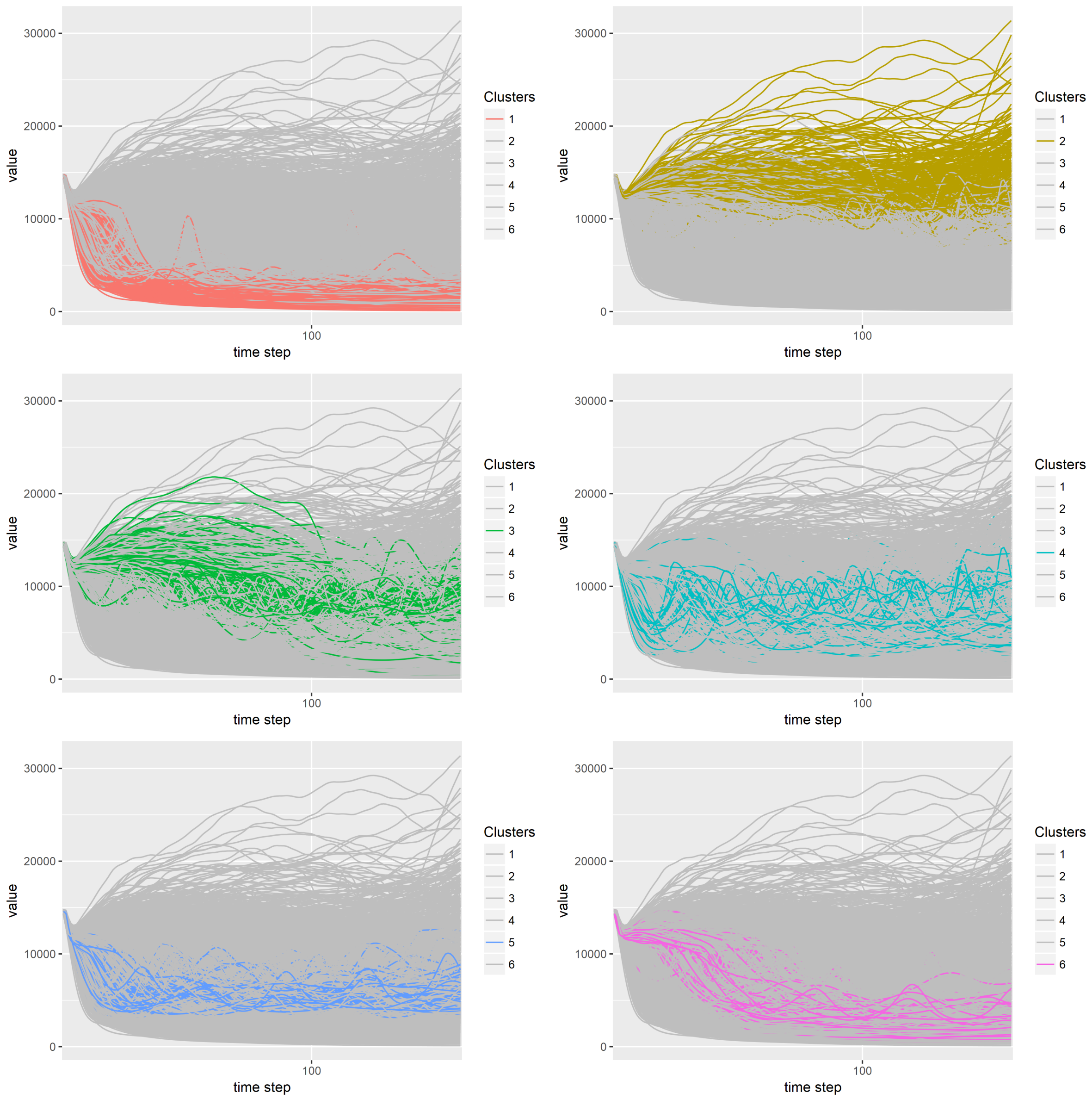



Figure 5.11: CID cluster constituents, 2000 runs, shale gas model

perspective. On the other hand, behavior 6 seems quite desirable - a gradual decline in oil prices to a very low level. This cursory analysis indicates that even for complex models, time series clustering can reveal patterns of interesting and policy-relevant behavior.

The derived clustering information is loaded back into EMA Workbench for rule induction with PRIM. As the shale gas model has almost 120 input variables, finding PRIM boxes is not straightforward. PRIM restricts dimensions iteratively in order of their effect on the objective function - the first few found dimensions are the most significantly restricting ones. However, these dimensions may not be same across all $k = 6$ clusters. Each cluster is likely to have somewhat different "predictors". Therefore, I find the six most important dimensions for each cluster, and then find the five most commonly shared dimensions across the clusters. These are the dimensions most likely to predict cluster membership. They are:

- *Switch prices or supply dominance in demand subtitution*
- *Initial unit costs oil*

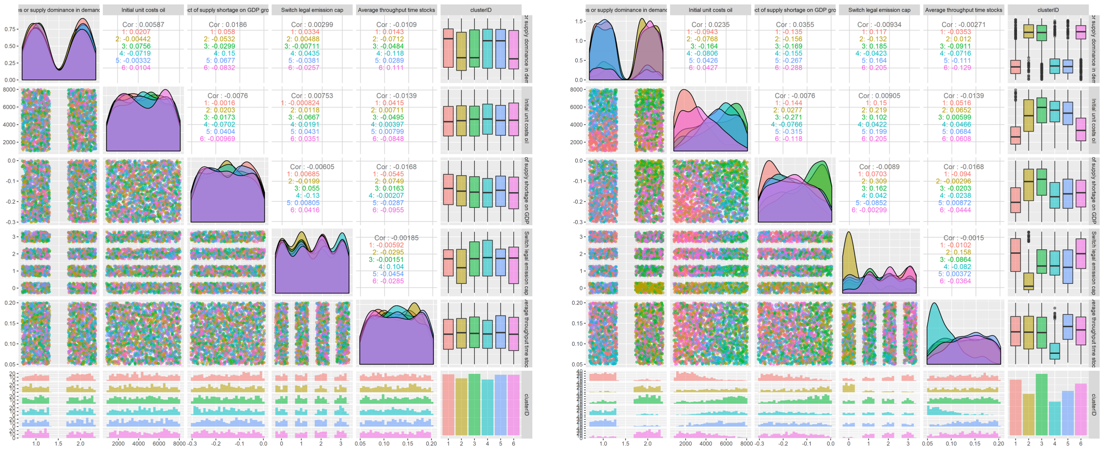
- *Effect of supply shortage on GDP growth*
- *Switch legal emission cap*
- *Average throughput time stocks*

It is not entirely surprising that two of these parameters are switches - structural uncertainty, commonly introduced through switches in Vensim, likely has more influence on behavior over time than parametric uncertainty. In Figure 5.12, pairings of these five descriptive inputs are shown, with inputs separated and shaded by assigned cluster, in a pairs plot (Emerson, Green, Schloerke, Crowley, Cook, Hofmann, and Wickham, 2013).


Figure 5.12: Pairs plot of five most predictive model inputs, for CID and $k = 6$, 2000 runs, shale gas model

Figure 5.12 is rather complex. In the following, I will first go over the plot in a general fashion, and then highlight some interesting details. The plot pairs the five most predictive input parameters given earlier against each other, creating a 5x5 grid. As this grid would be diagonally symmetrical, the upper half of the grid is used to display correlation values, while the lower half shows scatter plots of the two variables in question. The cells on the diagonal are used to display density estimates. A sixth row and column add information about the CID-assigned cluster memberships of the time series belonging to each set of input parameters. In the bottom-most row, histograms for each cluster and input parameter are given, and in the right-most column, box plots for each cluster and input parameter show median, quartiles and potential outliers. Finally, the bottom-right corner shows how many data points are assigned to each cluster.

Non-uniform distributions are interesting things to look for in pairs plots, as they hold predictive power. While a uniform distribution means a value is equally likely to occur anywhere over a range, a weighted



(a) Random assigned clusters (b) CID-assigned clusters

Figure 5.13: Time series output clustering can separate inputs: Randomly and CID-assigned clusters

(or non-uniform) distribution indicates that the likelihood of the value occurring is different over the range. As the displayed data was created using uniform sampling, and only then clustered, any non-uniformity must be introduced by the clustering! Figure 5.13 contrasts a "null hypothesis" clustering solution, where cluster memberships are randomly assigned, and the CID clustering solution.

For the diagonal cells in Figure 5.12, non-uniformity might be represented by a spike or dip in the density estimate of a certain cluster. For the scatter plots in the lower triangle, color "biases" towards a side or corner of the scatter plot indicate non-uniformity as well. For the histograms at the bottom, non-uniform distributions are easy to identify by unequally-sized columns (bin members). Similarly, outliers in the box plots are identified by looking for boxes shifted vertically relative to the other boxes.

In the following, I will use index-1 notation to refer to specific cells in this 6x6 grid (that is, the left-upper cell is [1,1], and the right-bottom cell is [6,6]). In the diagonal density estimate plots, the [1,1] cell is notable for its equal distribution of clusters, but unequal distribution within each cluster. Since this is a categorical switch variable, the distribution "valley" is not surprising, but the distributions for each cluster are noteworthy because they align very well with the two switch values (1 and 2). None of the clusters are divided evenly across the two categorical values possible. This indicates that this switch has a significant influence on cluster assignment. The box plot in [1,6] supports this insight. Cells [2,2] and [3,3] are not particularly noteworthy, although they do show some small peaks which may indicate non-uniform distribution. The histograms in [6,2] and [6,3] support this possibility - they display clear non-uniformity by being shifted to one end of the input range. Cells [4,4] and [5,5] are very interesting because they show significant non-uniform peaks for clusters 2 (gold) and 4 (aquamarine), respectively. This indicates that the associated variables (*switch legal emissions cap* and *average stock throughput time*) are highly predictive for these two clusters. This gels well with the cluster constituent plots in Figure 5.11 - cluster 2 (gold) predicts high oil prices, which makes sense if there is no global cap on emissions. Similarly, a low stock throughput time, as predicted by cluster 4 (aquamarine) will naturally lead to rapid oscillations in oil prices, as stocks empty and re-fill at a quick pace. This shows good correspondence between model behavior and induced input subspaces!

It is difficult to gain any real insights from visual inspection of the scatter plots, although the non-uniform distributions of some of the clusters are clearly evident. It appears that many clusters appear to favor one end or corner of the respective scatter plots, indicating that the clustering can indeed separate distinct inputs. In Figure 5.14 below, I show and discuss the PRIM-induced rule boxes for each cluster in each scatter plot, allowing better characterization of cluster alignment within each scatter plot.

While the histograms and box plots are mostly supporting material for the analysis of the pairwise scatter plots, their individual consideration is worthwhile, as they very clearly highlight single-variable-induced non-uniformities. In histogram [6,1], the uniformly distributed sampling of the switch is apparent, but also the predictive power, as none of the clusters show evenly remotely comparable histograms for the two possible switch values. This again indicates that this switch has a significant influence on the assigned cluster, and therefore model behavior. In histogram [6,2], non-uniform distributions are clearly visible for every cluster except 2 (gold) and 5 (blue). This again indicates that for some clusters, the initial oil cost is usefully predictive. Histogram [6,3] shows moderate non-uniformity, while cell [6,4] is very interesting because it is almost perfectly uniform except for cluster 2 (gold), for which it is highly non-uniform. The effect of a missing emissions cap on oil prices was already discussed above. In histogram [6,5], the spike for cluster 4 (aquamarine) observed in cell [5,5] is also reflected. The box plots, which are essentially different visualizations of the same data, show a similar pattern of noteworthiness as the histograms, so I will not discuss them further. None of the direct correlations show a value above 0.315, which is rather low. This indicates that no combination of two input parameters can explain cluster membership well. This is not surprising for a complex model with over 100 input parameters.

I now re-run PRIM for each CID cluster. In the pairs plot in Figure 5.14, I overlay the PRIM-induced boxes over the pairs plot. Colors are slightly faded to make the boxes easier to interpret. The boxes are also slightly jittered where necessary to improve visibility when overlaid. The chosen PRIM boxes are those which contain the most of the five best predictors identified above. Note that none of the PRIM boxes are fully specified, as can be seen from Table 5.1, where I list the induced subspace rules. This means that for every box, one or more of the five most predictive inputs was not predictive *at all*. This is somewhat surprising. It appears that variations in model behavior are not always caused by the same parameters.

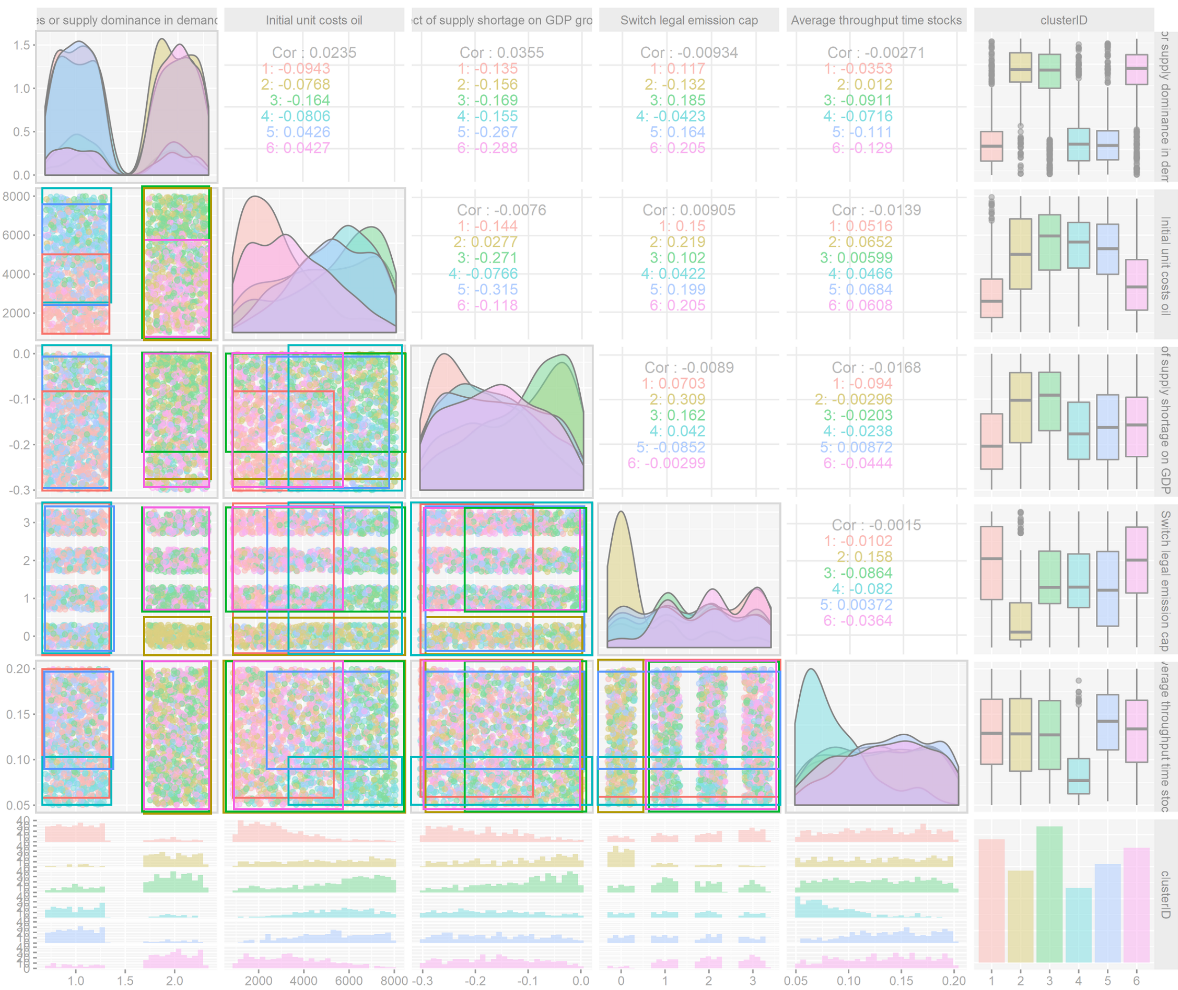



Figure 5.14: PRIM-induced subspaces over pairs plot of five most predictive model inputs, for CID and $k = 6$, 2000 runs, shale gas model

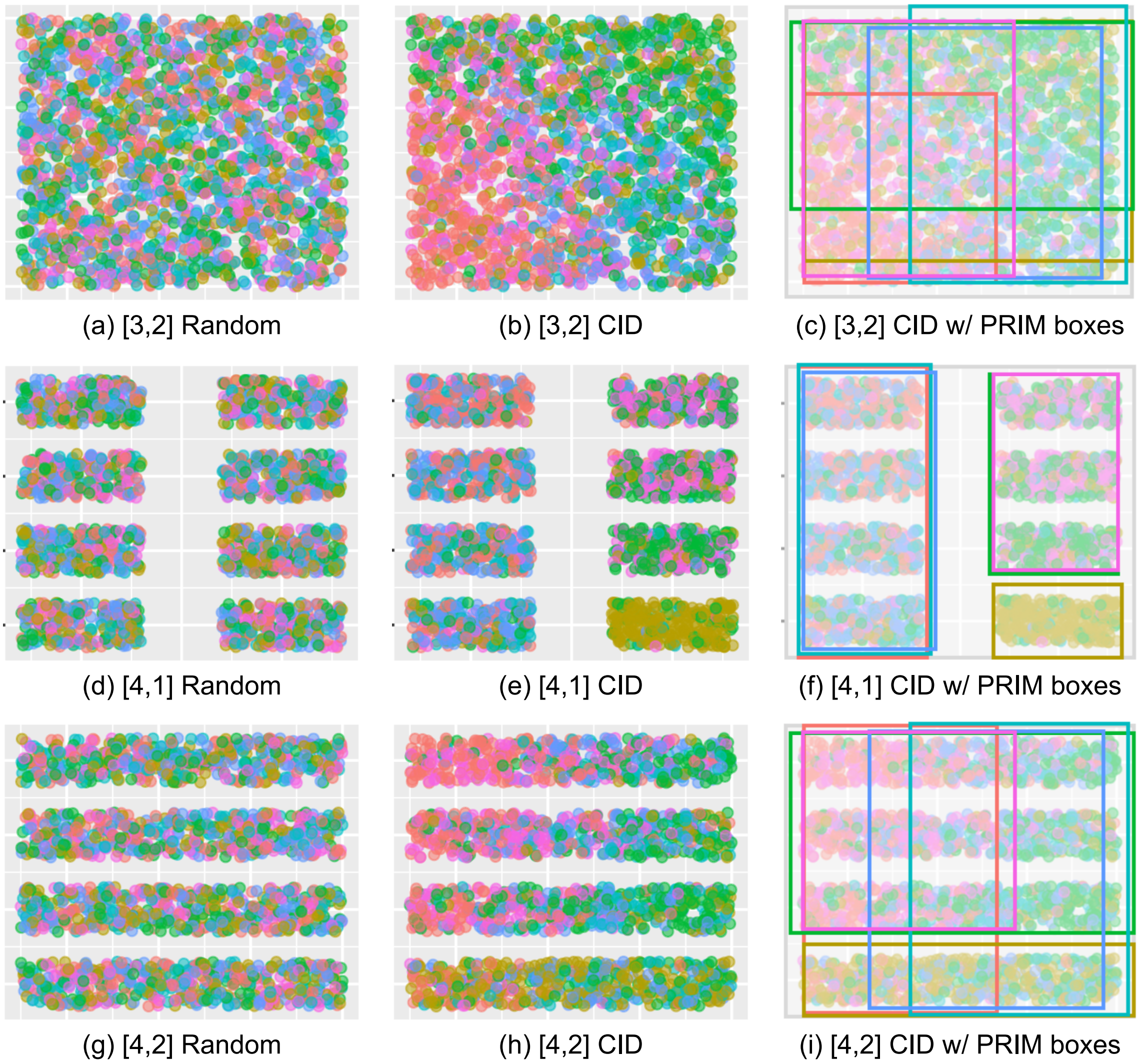



Figure 5.15: Magnification of interesting pairs plot cells, for random and CID-assigned cluster memberships, and CID-based PRIM boxes

In Figure 5.15, I provide magnified views of a few grid cells of interest - [3,2], [4,1], and [4,2]. For each cell, I show the "null hypothesis" random clustering and CID-assigned clustering from 5.13, and the induced PRIM boxes from 5.14.

## 5.6. Subspaces Analysis

It is difficult to assess the usefulness of the induced boxes directly from the pairs plot, as the five-dimensional boxes can only be shown as two-dimensional shadows. Ideally, a clear correlation would be seen between the boxes and the underlying data points - a box should tightly surround a large number of identically colored data points, but almost no other points. However, if this is not that case, that does not mean the box is poorly fitted, it may simply be separated in a different dimension. Thus, overlapping boxes in the pairs plot indicate poor predictive power of the variables in question, rather than poorly drawn ("loose") boxes. An exception can be seen in cell [4,1], where the cluster 2 (gold) box tightly surrounds a single subset of inputs for the two switch variables - no emissions cap and supply dominance, a combination which apparently predicts high oil prices very well. In cells [5,1] and [5,2], a clear separation between boxes 4 and 5 (aquamarine and blue) is observable and supported by the histograms in cell [6,5]. Both clusters are distributed non-uniformly, but at opposite ends of the input space. This distribution again relates to the *switch stock throughput time* - low throughput time causes fast oscillations (cluster 4), high throughput time predicts cluster 5 instead. From the plots in Figure 5.11, it is apparent how similar the two clusters are, apart from their oscillation frequency.

Table 5.1: PRIM-induced rules for CID clusters, $k = 6$

| | Parameter Rules | | | | |
|---|---|---|---|---|---|
| **Cluster** | Switch price/supply dominance | Initial unit costs oil | Effect of supply shortage on GDP growth | Switch legal emissions cap | Average throughput time stocks |
| 1 | 1 | [1002, 4945] | [-0.299, -0.088] | {0,1,2,3} | [0.057,0.2] |
| 2 | 2 | [-Inf, Inf] | [-0.27,0] | 0 | [-Inf, Inf] |
| 3 | 2 | [-Inf, Inf] | [-0.22, 0] | {1,2,3} | [-Inf, Inf] |
| 4 | 1 | [3368, 7997] | [-Inf, Inf] | {0,1,2,3} | [0.05, 0.11] |
| 5 | 1 | [2293, 7753] | [-0.299, -0.015] | {0,1,2,3} | [0.084, 0.2] |
| 6 | 2 | [1002, 5819] | [-0.282, -0.028] | {1,2,3} | [-Inf, Inf] |

Table 5.1 lists the specific parameter rules for each box. while many of the rules are quite broad, some do contain useful information for both modelling and policy analysis. The clustering for the *switch legal emissions cap*, which has four levels to indicate no emissions cap and three levels of increasingly severe caps, shows no differentiation between the three levels of emissions caps. It could therefore be argued that this switch is overspecified in the model (it could be reduced to just {0,1}), or not fully integrated, since the different levels of emissions cap apparently have no effect on model behavior. Similarly, it transpires that if there is no emissions cap at all, oil prices will almost invariably skyrocket. This is not entirely surprising, but still represent tangible policy advice - if a global emissions cap is not implemented, oil prices will likely drastically increase over time.

Table 5.2: PRIM box attributes for CID clusters

| **Targeted Cluster** | **Cluster Size** | **Total in Box** | **1 in Box** | **2 in Box** | **3 in Box** | **4 in Box** | **5 in Box** | **6 in Box** | **Cov** | **Dens** |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 386 | 391 | 254 | 1 | 1 | 41 | 69 | 25 | 0.658 | 0.649 |
| 2 | 287 | 218 | 1 | 176 | 37 | 3 | 0 | 1 | 0.613 | 0.807 |
| 3 | 426 | 564 | 16 | 61 | 253 | 17 | 13 | 204 | 0.594 | 0.449 |
| 4 | 234 | 263 | 36 | 11 | 21 | 144 | 49 | 2 | 0.615 | 0.548 |
| 5 | 308 | 291 | 164 | 16 | 65 | 73 | 223 | 38 | 0.724 | 0.766 |
| 6 | 359 | 423 | 34 | 14 | 106 | 15 | 19 | 235 | 0.655 | 0.555 |

While the pairs plot presented in Figure 5.12 indicates that the applied time series clustering does indeed reveal interesting correlations between model inputs and output behaviors, analysis of the induced PRIM boxes highlights some apparent limitations. In Table 5.2, the relevant statistics for each cluster's box are collected. For each targeted cluster, I provide the number of points attributed to that cluster, the total number of points within the box targeting that cluster, as well as the number of cluster members in the box for each cluster. The latter is a novel meta-box analysis. I also give coverage (cluster members in box divided by total cluster members) and density (cluster members in box divided by

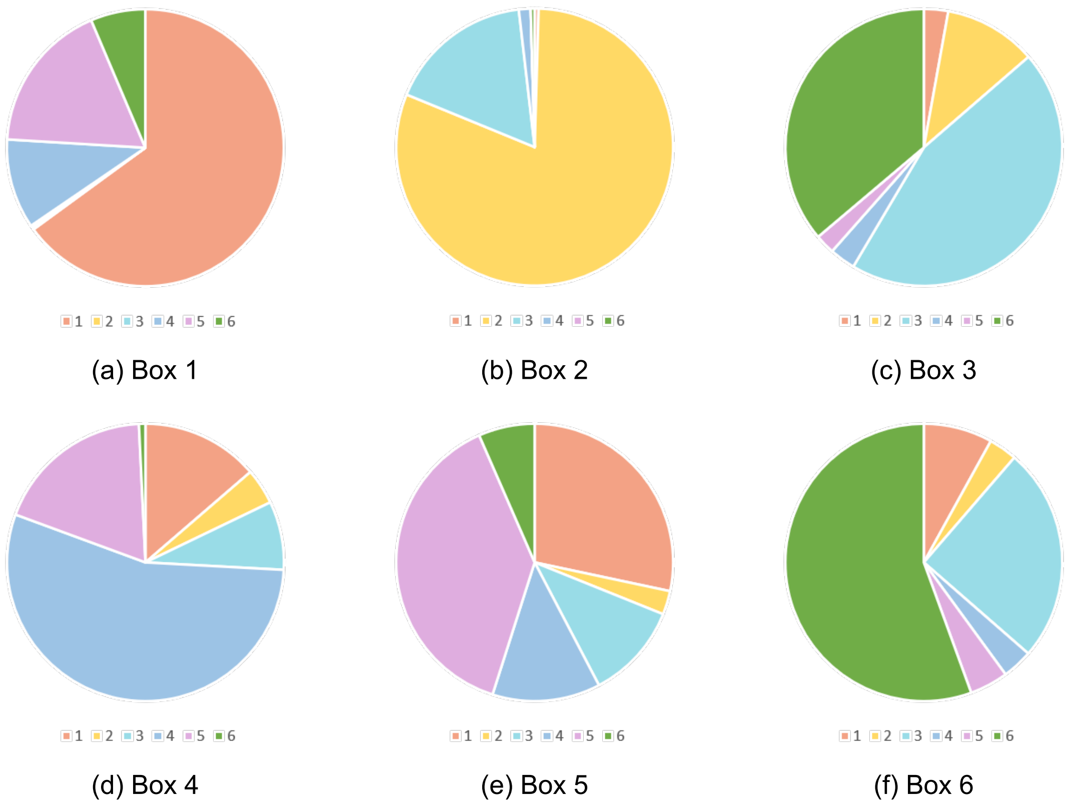


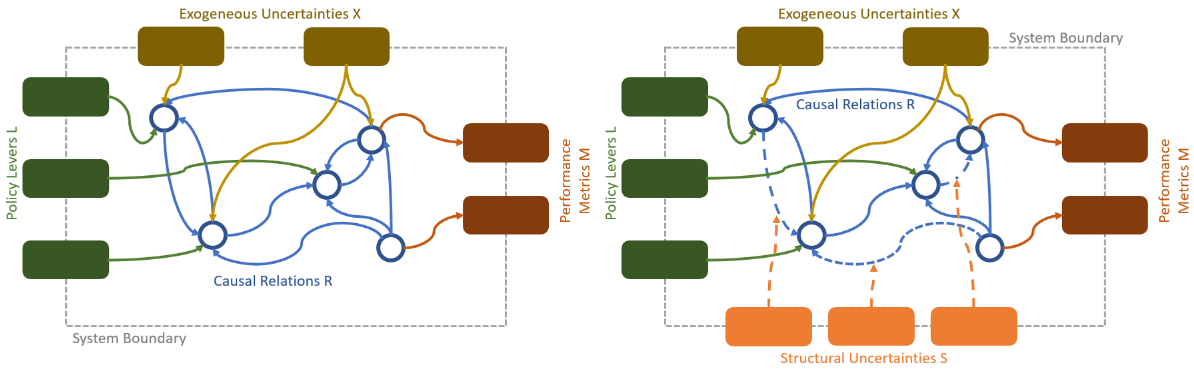

Figure 5.16: Assigned clusters of inputs inside each induced PRIM box

total points in box) for the targeted cluster for each box. Both are lower than an analyst would typically like to see. Values above 0.8 for both are considered desirable. The fact that this was not achieved, despite up to five restricted dimensions, indicates that these boxes may not be particularly useful for model-based decision support, as they do not allowed finely targeted policies and interventions - many relevant cases are excluded, and many irrelevant cases included. This is emphasized by Figure 5.16, which shows pie charts for each induced box and the assigned clusters of all points included in that box. Ideally, these charts would show a single color. In reality, it turns out that most boxes include points belonging to all six clusters.

The underlying causes for this unsatisfying performance are not clear. I believe there are two main issues at play. Firstly, I partition the 2000 model outputs into 6 clusters. It is likely that this conflates distinct model behaviors, reducing the explanatory power of the resulting boxes. Furthermore, the inherently orthogonal approach of PRIM restricting input dimensions makes it difficult to account for interactions between multiple input parameters - even though such second-order effects are to be expected in complex system models.

## 5.7. Suitability of Input Dimensions for PRIM

In this analysis, I use the five most predictive input parameters with no regard for their function in the model. Strictly speaking, this is erroneous, as one of the inputs is used to introduce structural uncertainty (*switch price/supply dominance in demand substitution*) through a parametric uncertainty. This is accomplished through conditional switches in the system dynamics model itself. In a policy context, it does not make much sense to apply rule induction to this parameter, as the uncertainty surrounding it can inherently never be reduced - it will never clearly be determinable which switch setting reflects reality and therefore is in effect. If this model was being used for real policy analysis, the parametric uncertainties should be separated into truly reducible (often parametric) uncertainties, and irreducible uncertainties, which are often of structural nature, but proxied through parametric inputs.




(a) XLRM (Lempert et al., 2003) XLRMS (proposed)

Figure 5.17: XLRM framework could be extended with structural uncertainties S, creating XLRMS

The fact that these are not separated may be a result of the XLRM framework (Lempert et al., 2003) used as the foundation of the EMA Workbench (Kwakkel, 2017), which separates model inputs merely into policy levers (controllable inputs) and exogeneous uncertainties (uncontrollable inputs) and leaves no room for reducible and irreducible uncertainties to be separated. Arguably, the XLRM framework only accommodates shallow uncertainty, as there is no room for changes in the model's causal relations, or parameters which may affect these. Thus, an "XLRMS" framework as shown in Figure 5.17, with Structural Uncertainties (S) added in orange affecting the causal relations in the model itself, can be proposed.

The introduction of structural uncertainties as a distinct set of model inputs would also open a novel model analysis method which might be called structural sensitivity analysis. This could help explore the sensitivity of model outcomes to purely structural questions, rather than reducibly uncertain parametric inputs. This could highlight if/when a model is overly sensitive to a single structural uncertainty question, and improve understanding of the behavior of deeply (structurally) uncertain models. An example of this can be seen in the presented shale gas model, where the structural uncertainty switch for *price/supply dominance in demand substitution* clearly has a significant impact on model behavior, which may or may not be substantiated. As the focus of this thesis is methodological exploration of time series clustering, I do not explore this further and leave it to future research.

## 5.8. Chapter Summary

The shale gas model by Auping et al. (2014) is a complex system model forecasting possible global energy and political dynamics. It features hundreds of variables, dozens of feedback loops, and both parametric and structural uncertainties. Accordingly, the relation of model inputs to outputs, as is performed in scenario discovery, is not trivial. Firstly, it turns out that cluster silhouette widths cannot recommend a "likely" cluster count $k$. Secondly, no clustering method intuitively outperforms the others, although SBD and DTW create very unevenly sized clusters. CID, CORT and DWT create roughly evenly sized clusters, I therefore recommend these methods for future investigation. Faced with these two analytical uncertainties, I choose to study the CID clustering solution for $k = 6$ clusters and 2000 experiments. As can be seen in Figure 5.13, the clustering solution indeed separates the inputs based on output behavior, introducing output-predictive non-uniformities for each cluster's input space without any investigation of the model structure - a true "black box" input-output analysis.

For each cluster, I induce the PRIM rule box that generates (most of) the cluster's outputs along the five (of 120) most common and predictive input dimensions. While the resulting boxes are reasonably explanatory, their statistical attributes are not sufficient for rigorous policy analysis, as they generally contain too many inputs belonging to other clusters. This is likely due to the complexity of the analyzed model, but also the choice of cluster count $k$ and box orthogonality limitations imposed by the scenario discovery algorithm PRIM.

The usefulness of the found rules is questionable, as one of the restricted dimensions is an irreducible structural uncertainty. Arguably, such uncertainties should not be considered in scenario discovery. This argument can be extended to a potentially useful methodological separation of exogenous uncertainties into reducible (often parametric) uncertainties and irreducible (often structural) uncertainties within an extended XLRM framework.

# 6

# Subspaces Analysis

In conventional scenario discovery, the goal is often to find a single subspace of interest within the input space which generates a set of outputs of interest through some system model. Outputs are defined as interesting through some external criterion applied to the output values. In the previous chapters, I presented a method of using time series analysis to cluster output time series, and find the corresponding input subspaces for each cluster. This allows the separation of outputs by intrinsic behavior rather than extrinsic criterion. This is advantageous because externally introduced thresholds or other conditions may conflate distinct model behaviors. Additionally, these external criteria are often subjective and may introduce bias into the analysis.

My new approach includes a conceptual transition from searching for a single input subspace of interest to finding multiple (likely behaviorally) distinct subspaces within the input space, and opens up a variety of new analysis possibilities. In the following, I will explore three such avenues, namely the concept of subspace separability, the re-imagination of rule induction as a two-phase, local and global optimization problem, and the use of clustering in determining model reference behavior, which is useful both for communication and robust decision making.

## 6.1. Subspace Separability

The creation of multiple distinct subspaces within a model's input space allows for analysis not only of the subspaces themselves, as is commonly done in conventional scenario discovery, but also of the relations and interactions between the subspaces - a meta-analysis of the subspaces, so to speak.

One policy analysis-relevant example of such an analysis is the investigation of subspace separability. Each input subspace generates a distinct model behavior in its associated outputs. These outputs are numerically different, but behaviorally identical or comparable. Therefore, they are likely vulnerable or sensitive to the same intervention policies. Thus, a distinct policy to ensure desirable outcomes can be designed for each input subspace. However, this necessitates that the subspaces do not overlap - they must be well-separated. If this is not the case (e.g. when subspaces overlap), policies may inadvertently target outputs of a different behavior, with unforeseeable effects. Thus, separability can be seen as a desirable global criterion of the induced subspaces.

For the shale gas model presented earlier, Table 6.1 shows the shared members of every two-box combination. The box rules are those given in Table 5.2 for the CID clustering solution with $k = 6$ of the future oil price scenarios generated by the shale gas model. The diagonally symmetrical table gives the number of data points that could be attributed to (at least) two subspaces. As the data is sampled using Latin Hypercube, the number of shared data points is proportional to the shared volume of the two boxes, or their overlap.

While there are only four overlaps between boxes, only one box (for cluster 2) is perfectly isolated. Cross-referencing box pairs with many shared points (e.g. 1 & 5, or 3 & 6) with the constituent plots

Table 6.1: Shared overall members for CID-based PRIM boxes

| Boxes | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **1** | 391 | 0 | 0 | 65 | 208 | 0 |
| **2** | 0 | 218 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 564 | 0 | 0 | 324 |
| **4** | 65 | 0 | 0 | 263 | 100 | 0 |
| **5** | 208 | 0 | 0 | 100 | 291 | 0 |
| **6** | 0 | 0 | 324 | 0 | 0 | 423 |

in Figure 5.11 may explain why this is the case, as these pairs show broadly similar output behaviors. I posited earlier that input subspaces (and, by extension, their boxes) are linked to distinct output behaviors - it follows that if two clusters show similar output behavior, their input subspaces will likely overlap significantly.

Figure 6.1 shows a network graph of the six found subspaces for the shale gas model. Node size denotes the number of data points in each subspace, and the edges are weighted by shared data points between two subspaces. The isolation of cluster 2 is visible, as well as the strong overlap between subspaces 3 and 6.

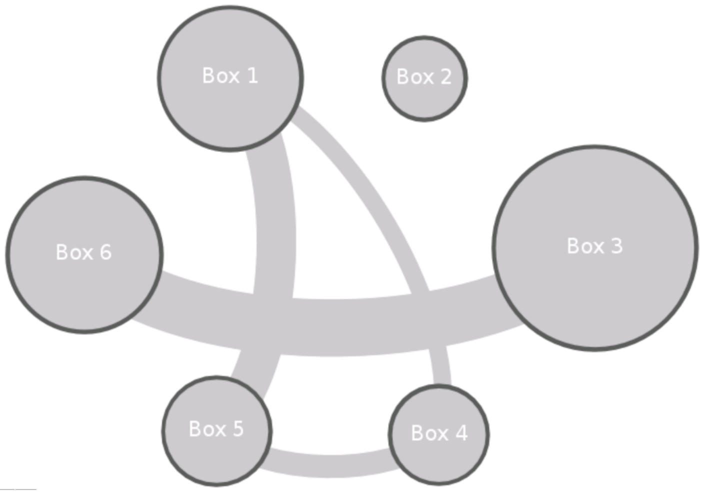



Figure 6.1: Network graph of shale gas model input subspaces, CID solution

Subspace separability is a novel concept in scenario discovery, as the induction of multiple unrelated subspaces in the input space has to my knowledge never been performed. Although PRIM does foresee finding multiple boxes of interest within the input space, these are considered to be subsets of the overall subspace of interest, rather than distinct subspaces of interest in their own right (Bryant and Lempert, 2010). Furthermore, PRIM advocates removing data points associated with previously found boxes, which would certainly be counterproductive here.

## 6.2. Rule Induction using Local and Global Criteria

Since subspace separability has easily definable metrics (shared data points, or shared hyperspace volume) and a definitive goal (zero box overlap), it could also be used during rule induction using a many-objective algorithm, as is presented by Kwakkel (2018). However, this would require a re-imagination of rule induction as an iterative two-step optimization process. In a first step, individual subspaces would be found using established "local" subspace criteria. In a second step, "global" criteria of these subspaces would be assessed. Based on this assessment, the parameters of the local subspace induction could be modified, and another cycle could be started.

In the first step in this new form of scenario discovery, a many-objective optimization algorithm would seek to induce rules for a single subspace using well-established metrics (Friedman and Fisher, 1999) such as:

- Coverage (points of interest inside subspace / total points of interest)
- Density (points of interest inside subspace / total points inside subspace)
- Interpretability (1 / number of restricted dimensions)

In this first phase, the optimizer would seek to maximize these three criteria. This algorithm would have to be executed for every subspace in turn, and would likely be governed by some definable parameters, much like PRIM has a settable step size, box threshold, etc.

In a second phase, global criteria of the various subspaces found in the first step would be assessed. These global criteria might be conceptualized as:

- Separability (1 / overlap between subspaces)
- Coverage (points inside subspaces / points in input space)
- Population (number of found subspaces)
- Validity (explanatory power of found subspaces)

Separability, defined through box overlap, was already introduced above. It should be maximized in order to ensure subspaces are well-separated.

Coverage would need to be introduced to ensure that subspaces do not just shrink to high-density and high-interpretability shapes, but adequately cover the entire input space. This maximization is likely to lead to reduced density and interpretability within the subspaces, but is beneficial from a policy analysis standpoint because it removes ”blind spots” from the input space for which no defined subspace (and therefore no policy response) is known.

Less subspaces are easier to understand and plan for than many subspaces - the more the input space is partitioned, the harder it becomes to conduct useful policy analysis. Thus, the number of subspaces - the population - should be minimized.

However, as the number of subspaces is reduced, more and more distinct model behaviors (and policy vulnerabilities) are lumped together into increasingly larger subspaces. By extension, these subspaces become less useful and explanatory of model behavior. Thus, the validity of the found subspaces should also be assessed, and maximized. This counteracts the proposed population minimization, and allows for exploration of possible trade-offs between the two criteria.

Based on the assessment of these four global criteria, the parameters of the local subspace induction algorithms could be adapted. If time series clustering were to be used, these parameters could include the desired cluster count $k$, the used clustering algorithm, and the parameters governing those algorithms. This could be accomplished through any number of optimization techniques, including genetic algorithms or hill-climbing methods. Based on the updated assigned clusters found through time series clustering, the subspaces for each cluster could then again be induced locally, and then again assessed using the global criteria. Through this iterative process, well-separated and highly predictive subspaces within the input space could be found automatically. A conceptual flow chart of such an algorithm is shown in Figure 6.2.

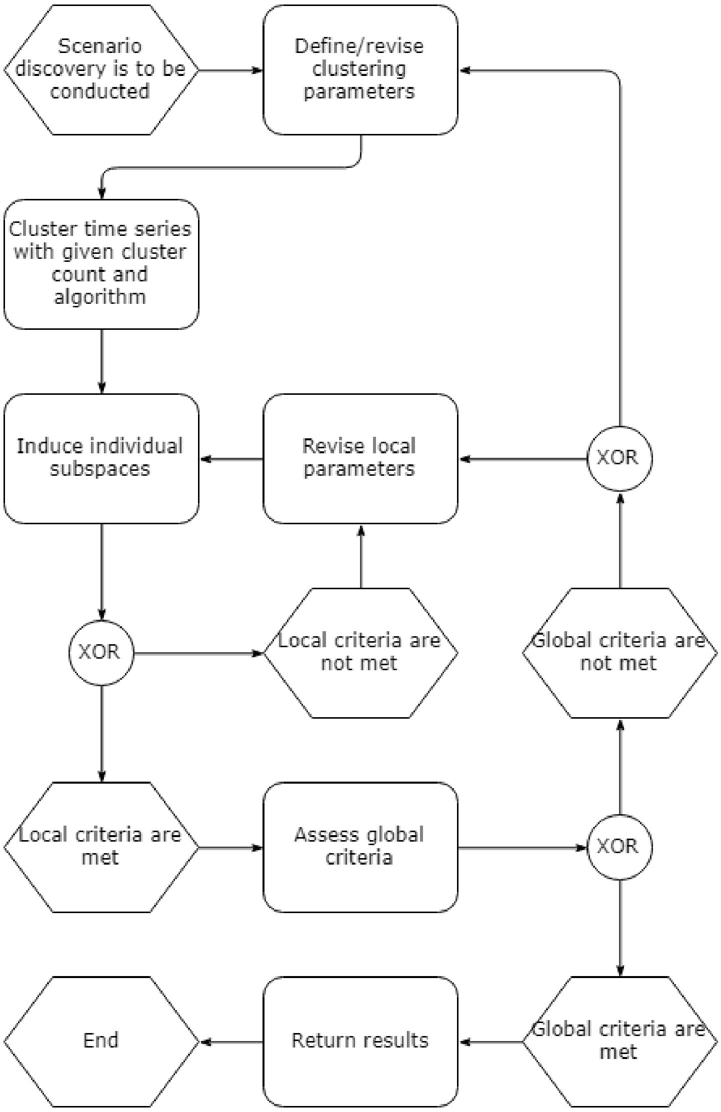



Figure 6.2: Conceptual flow chart of iterative, two-stage rule induction using local and global criteria

## 6.3. Model Reference Behavior

The communication of inherent model characteristics such as typical behavior modes is often difficult. Analysts are confronted with the problem of manually identifying ”typical” or ”descriptive” model behaviors from a collection of potentially thousands of experiment outputs using intuition or external criteria (Auping et al., 2016). Time series clustering offers an alternative approach. Since partitional clustering is often conducted around medoids (Cryer, 2008), and these medoids (sometimes called centroids) are inherently the most dissimilar time series in the set of outputs, they could also be described as the most behaviorally diverse outputs. These medoids are easily determined by finding the time series in each cluster which is closest to all other cluster members. Thus, partitional time series clustering offers a way to objectively find and plot the $k$ most diverse time series outputs of a model. Figure 6.3 provides an example using the CID clustering method and $k = 6$ on the shale gas model data. The six cluster centroids are specifically highlighted (and the other time series faded out) to emphasize their different behaviors. The legend is omitted to maximize plot space.

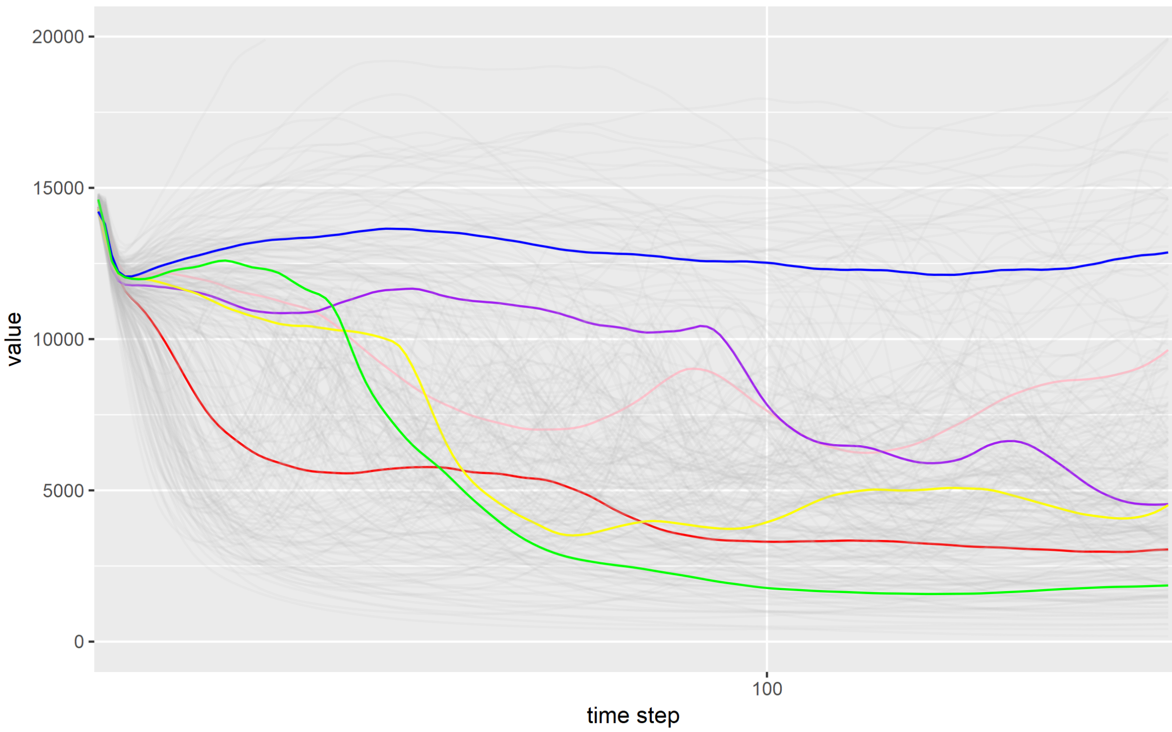



Figure 6.3: CID cluster centroids representing $k = 6$ distinct model behaviors

The problem of finding ”diverse” or ”representative” model outputs is also relevant for some robust decision making methods, which require the provision of reference or base line scenarios (Kwakkel, Haasnoot, and Walker, 2015; Watkins and McKinney, 1995). Kwakkel and Jaxa-Rozen (2018) specifically highlight the need for a more systematic method of finding reference scenarios. It stands to reason that these should be as behaviorally diverse as possible in order to cover a wide range of model reference dynamics. Thus, time series clustering could be employed to identify these reference scenarios. As each cluster has one centroid, any number of reference scenarios can be generated by changing the number of clusters the generated time series outputs are partitioned into. Some clustering methods, such as TADPole (Begum et al., 2015), do not calculate the distance between every output time series, and are thus significantly faster for large data sets. This may make them especially suitable for finding reference scenarios in the large model output data sets required to capture complex model behaviors.

## 6.4. Chapter Summary

The conceptual transition from finding a single input subspace of interest, as it is done in conventional scenario discovery, to finding multiple input subspaces requires a re-characterization of rule induction as an iterative two-step optimization problem. While subspaces can be found individually through local optimization towards established local criteria such as coverage, density and interpretability, global criteria are necessary to ensure that global performance is satisfactory. These global criteria may include subspace separability, input space coverage, number of subspaces, and validity of the subspaces, and can also be addressed using a many-objective optimizer. These local and global optimizations must be conducted sequentially and iteratively until a satisfying partition of the input space into subspaces is found.

Furthermore, time series clustering can be used to identify model reference behaviors. Specifically, the centroids of each time series cluster can be seen as the most distant (and therefore diverse) time series outputs in an output data set. They can therefore be used to explain potential model behavior modes visually. Additionally, they can also be used as reference scenarios for robust decision making algorithms.

# 7

# Conclusions, Reflections, Future Work

Scenario discovery is a method of relating a complex system model's outputs to its inputs, the goal being to find the generative input space that usefully predicts an output meeting some external criteria of interest (Kwakkel and Jaxa-Rozen, 2016). This extrinsic assessment of model outputs requires expert knowledge and introduces subjective judgement into the scenario discovery process. To overcome this, I proposed a new approach based on intrinsic model dynamics. In *behavior-based scenario discovery*, model outputs are first partitionally clustered into subsets based on their behavior over time. Then, the generative input subspaces for each cluster are determined using the Patient Rule Induction Method (Bryant and Lempert, 2010). I demonstrated this approach using three increasingly complex system dynamics models. In all three cases, time series clustering proved useful in partitioning the model input space into separate subspaces associated with distinct output behaviors over time. As behavior-based scenario discovery only analyzes model in- and outputs without consideration of model structure, it is a true black box analysis technique. However, the resulting induced rule boxes are heavily dependent on the chosen clustering parameters, including clustering method and number of clusters. For complex models with many input variables, the found boxes may not be perfectly predictive. This is also partially due to limitations imposed by the employed rule induction method.

The transition from searching for a single input subspace of interest to partitioning the entire input space into multiple behaviorally distinct subspaces introduces new requirements for scenario discovery. The analysis of meta-heuristics of the different input subspaces, such as separability or global coverage, becomes possible. These meta-heuristics could be used to automate behavior-based scenario discovery through an iterative, two-stage many-objective optimization algorithm with both local and global optimization metrics. Furthermore, cluster attributes such as centroids could be used for visualization of exemplary model behaviors, or to generate reference scenarios for many-objective robust decion making algorithms.

## 7.1. Research Questions and Conclusions

In the following, I will address the research questions I posed in the first chapter. I will also recap other critical insights that appeared during the research on this topic.

> (1) How can time series clustering be applied to the outputs of simulation models to partition the outputs into subsets with similar dynamics?

Based on the three system dynamics models used as test cases in this thesis, I recommend the following approach to apply time series clustering for scenario discovery:

1. Generate a large number of time series model outputs using a suitable sampling method. I used Latin Hypercube Sampling (Kwakkel, 2017).
2. Identify a suitable number of partitional clusters in the data using a cluster validity index. I used silhouette widths (Rousseeuw, 1987) with mixed success.

3. Apply a suitable time series clustering algorithm to the data, partitioning the time series into a suitable number of clusters. I found Complexity-Invariant Distance (Batista et al., 2011), Combined Temporal Correlation and Raw Values (Chouakria and Nagabhushan, 2007), and Discrete Wavelet Transform (Zhang et al., 2006) to offer the best balance of speed, intuitive behavioral partitioning and cluster size.
4. For each cluster, induce the generative rule box using a suitable rule induction method and the assigned clusters for each input-output pairing as condition. I used the Patient Rule Induction Method (Bryant and Lempert, 2010) implemented in the EMA Workbench (Kwakkel, 2017) in Python.

Note that a number of time series clustering methods exist. Of the fourteen methods explored in this thesis, the only one I cannot recommend is latent class Markov model clustering. However, this is likely due to issues with the implementation provided by the seqHMM package (Helske and Helske, 2017), which regularly failed to converge, rather than a fundamental issue with the method itself. In general, all packages I used in both R and Python showed a high degree of maturity and usability. I have no reservations recommending any of them (apart from seqHMM) for further use.

The approach as described above closely mirrors the process employed in conventional scenario discovery (Lempert et al., 2008). Therefore, behavior-based scenario discovery could be employed not just as a standalone analysis, as I used it, but also as the analytical core of robust decision making frameworks such as Many-Objective Robust Decision Making (Kasprzyk et al., 2013).

> (2) How can the generative input subspaces for multiple subsets of model outputs be induced?

The input subspaces for each subset (or cluster) of outputs can easily be induced using established rule induction method such as the Patient Rule Induction Method or Classification and Regression Trees (Lempert et al., 2008). The rule induction method must be executed once for each cluster. Unlike in conventional PRIM, data points are not removed from the input space once they are assigned to a box, as this might distort the induction process.

> (3) How does the transition from searching for a single input subspace of interest to searching for multiple distinct subspaces conceptually affect the scenario discovery process?

The main conceptual difference between searching for a single subspace of interest and searching for multiple behaviorally distinct subspaces is the potential for subspace meta-analysis. Not only can the subspaces be investigated using established metrics, but the relations between the subspaces can also be analysed. The most policy-relevant new criterion here is the notion of subspace separability, which describes how multiple input subspaces overlap. As this overlap can reduce the effectiveness of policy interventions, subspace separability should be maximized. For this purpose, a novel two-phase many-objective optimization algorithm could be designed, which would optimize for both local (for every subspace individually) and global (across subspaces and the input space) criteria.

> (M) In complex system models, how can the input subspaces associated with decision-relevant system behaviors over time be found?

As demonstrated in this thesis, partitional time series clustering, coupled with the Patient Rule Induction Method, is a suitable method for finding the input subspaces associated with decision-relevant system behaviors. The consideration of model dynamics over time introduces a novel, intrinsic analysis and separation of model behaviors. However, it is not a panacea for identifying and separating policy vulnerabilities. Behavior-based scenario discovery is highly sensitive to choices in cluster count, clustering method, and other parameters. Therefore, it should always be used in concert with other analysis techniques such as sensitivity analysis. Furthermore, the introduction of global subspace criteria in addition to established local criteria may require a revision of automated rule induction methods.

Overall, behavior-based scenario discovery integrates well into the three-step process of scenario discovery introduced earlier as a combination of high-resolution sampling, time series clustering-based determination of outputs of interest, and PRIM-based rule induction. Table 7.1 shows an overview of current options for each step in scenario discovery. Scenario discovery can be seen as a catalogue

Table 7.1: Options for scenario discovery steps

| **Step 1: Generation of Model Outputs** | **Step 2: Identification of Outputs of Interest** | **Step 3: Induction of Subspace Rules** |
|---|---|---|
| Low-resolution sampling (Davis et al., 2007) | External criteria (Hamarat et al., 2013) | Patient Rule Induction Method (Bryant and Lempert, 2010) |
| High-resolution sampling (Halim et al., 2016) | Directed search (Pruyt et al., 2011) | Classification and Regression Trees (Bryant and Lempert, 2010) |
| Adaptive sampling (Islam and Pruyt, 2016) | Time series clustering (presented here) | Many-objective evolutionary algorithm (Kwakkel, 2018) |

of options, rather than a fixed procedure, from which the analyst must choose the right method(s) for each step. The citations given refer to examples of implementations.

A number of other insights appeared during this research effort. In the Brusselator and shale gas model analyses, the need for a non-orthogonal rule induction method became increasingly obvious. I originally intended to include an attempt at implementing such a non-orthogonal method using convex hulls in this thesis, but could not make substantial progress in time. This remains a glaring research gap and weakness in the model-based decision support field.

In the shale gas model, irreducible and reducible uncertainties are lumped together in the model definition. This is a consequence of the XLRM framework (Lempert et al., 2003) underlying the EMA Workbench, which does not foresee structural uncertainty or separate reducible and irreducible uncertainties. Unfortunately, this also hinders policy analysis in general, and scenario discovery in particular, as rules may be induced across dimensions which are irreducibly uncertain. I believe this could be remedied by extending both the EMA Workbench and the XLRM framework in such a way that reducible and irreducible uncertainties can be separated. This would also enable new model analysis methods such as the structural sensitivity analysis.

While I focused my research on time series clustering for scenario discovery, two other applications of partitional clustering in model-based decision support did appear, both related to cluster centroids. These centroids, sometimes called medoids, are the most representative and central members of every time series cluster. By extension, they are also the most distance time series from one another, and the most behaviorally diverse. As such, they are useful both for highlighting potential model behavior modes, and as reference scenarios in robust decision making algorithms. The latter fills a research gap indicated by Kwakkel and Jaxa-Rozen (2018).

## 7.2. Reflection and Critique

This thesis is not based on any established research framework or methodology. This is not because I purposefully avoided doing so, but because of the broad nature of my original line of research - ”how could we use time series clustering in scenario discovery?”. As there was no definitive goal to work towards, I was never really sure in which direction I was taking my research, or rather, my research was taking me. In retrospect, I should have more clearly defined an achievable and fully answerable research question early on in my literature review. This would have helped me structure my time and efforts in a more productive manner. However, the openness of my line of research also allowed me to explore completely new issues that came up, such as subspace separability, without being too concerned whether I was still doing what I was supposed to do. As it stands, I certainly could have benefited from a clearer goal to work towards in the final stages of work, especially when I got sidetracked by interesting but not fully relevant questions that came up unexpectedly. One example is the use of polytopes and convex hulls to do rule induction. I was unsatisfied with the orthogonal boxes induced by PRIM, and spent quite some time trying to improve rule induction, even though it was only tangentially related to my main line of research. In the end, I have no nothing to show for this effort. While it was a useful learning experience, I feel this time could have been better spent on material that did make it into this thesis.

Early in my research I decided to focus on system dynamics models. Such models are very useful from an analysis perspective because their behavior is deterministic, so even nonlinearities such as bifurcations can be easily identified and mapped. However, many modern simulation studies use stochastic methods such as agent-based or discrete event models. While I did design my proposed method to work on black box models, and therefore believe it should be usable with any modelling paradigm, I cannot make any definitive statements in this regard.

Finally, my initial two case studies were so simple that the number of bifurcation regions could easily be identified through time series clustering of the outputs. However, the third case was so complex that my previously used method of determining likely cluster counts, the cluster silhouette width, failed completely. This raises some interesting questions. Can bifurcation regions even be usefully mapped in highly complex models? How many separate ”policy regions” in an input space can stakeholders and analysts cope with? Might other cluster validity indices provide clearer information on distinct model behaviors? I cannot answer any of these questions, even though they are critically important to the re-characterization of scenario discovery from searching for a single input subspace of interest, to searching for multiple behaviorally distinct subspaces.

## 7.3. Avenues of Future Work

A number of possible future lines of research have emerged from this thesis. I will briefly elaborate on them here.

While scenario discovery is currently a single-stage many-objective optimization problem, this concerns only a single subspace of interest, and the optimization criteria are formulated as such. Once multiple subspaces become relevant, both subspace-specific local and inter-subspace global criteria become relevant for rule induction. Accordingly, automated scenario discovery should be re-formulated to consider both local and global criteria. I presented a possible algorithmic approach earlier.

Since the discovery goal is no longer a single subspace of interest, but a partitioning of the entire input space, the number of subspaces it should be partitioned into becomes extremely influential for policy design. Accordingly, more research should be done on now to best determine this number. One approach might be to examine further cluster validity indices. Furthermore, multivariate time series clustering could be explored. All work I presented here concerns univariate time series - sequences which stand alone, and can be compared on a one-to-one basis. However, models often have multiple key performance indicators. Accordingly, each experiment generates multiple time series, which are linked together by virtue of stemming from the same experiment. In multivariate time series clustering, sets of time series are compared on a set-to-set basis, potentially improving discrimination of model behaviors. Finally, adaptive sampling (Islam and Pruyt, 2016) might be useful in building distinct behavior clusters directly through input sampling, rather than first using random sampling such as Latin Hypercube and only then clustering the outputs.

The Patient Rule Induction Method is currently the gold standard for rule induction in scenario discovery. However, the orthogonal approach to inducing rules severely limits the quality of the induced boxes, especially for complex models. The development of non-orthogonal rule induction methods could greatly benefit (behavior-based) scenario discovery, especially where nonlinear models are concerned.

For the shale gas model, I briefly mentioned how the conflation of reducible and irreducible uncertainties in the model input space reduces the usefulness of rule induction. While rule induction should only be performed on reducibly uncertain parameters, the current implementation in the EMA Workbench leaves no room for this separation apart from somewhat tedious manual removal. If reducible and irreducible uncertainties were clearly separated, the usefulness of input subspaces found through scenario discovery could be improved, and novel analysis methods such as structural uncertainty sensitivity analysis could be introduced. This separation can easily be extended to the XLRM framework, which lacks any consideration for irreducible/structural uncertainties. It should be extended accordingly to better capture deep uncertainty.

Finally, the usage of time series clustering for identifying reference scenarios in robust decision making should be explored. There is currently no established method of identifying such reference or baseline scenarios, even though the outcomes of optimization algorithms based on them may significantly depend on the choices made. A comparison of robust decision making algorithm outputs using analyst- and clustering-supplied reference scenarios could prove valuable in this regard.

# References

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013. ISSN 00313203. 10.1016/j.patcog.2012.07.021.

Willem .L. Auping, Sijbren de Jong, Erik Pruyt, and Jan H. Kwakkel. The Geopolitical Impact of Shale Gas: The Modelling Approach. In *32nd International Conference of the System Dynamics Society*, 2014.

Willem L. Auping, Erik Pruyt, Sijbren de Jong, and Jan H. Kwakkel. The geopolitical impact of the shale revolution: Exploring consequences on energy prices and rentier states. *Energy Policy*, 98: 390–399, 2016. ISSN 03014215. 10.1016/j.enpol.2016.08.032. URL `http://dx.doi.org/10.1016/j.enpol.2016.08.032`.

Steve Bankes. Exploratory Modeling for Policy Analysis. *Operations Research*, 41(3):435–449, 1993. ISSN 0030-364X. 10.1287/opre.41.3.435. URL `http://pubsonline.informs.org/doi/abs/10.1287/opre.41.3.435`.

Gustavo E A P A Batista, Xiaoyue Wang, and Eamonn J Keogh. A Complexity-Invariant Distance Measure for Time Series. *SIAM International Conference on Data Mining*, pages 699–710, 2011. ISSN 1384-5810. http://dx.doi.org/10.1137/1.9781611972818.60. URL `http://epubs.siam.org/doi/abs/10.1137/1.9781611972818.60`.

Nurjahan Begum, Liudmila Ulanova, Jun Wang, and Eamonn Keogh. Accelerating Dynamic Time Warping Clustering with a Novel Admissible Pruning Strategy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, pages 49–58, 2015. ISBN 9781450336642. 10.1145/2783258.2783286. URL `http://dl.acm.org/citation.cfm?doid=2783258.2783286`.

Lena Börjeson, Mattias Höjer, Karl Henrik Dreborg, Tomas Ekvall, and Göran Finnveden. Scenario types and techniques: Towards a user's guide. *Futures*, 38(7):723–739, 2006. ISSN 00163287. 10.1016/j.futures.2005.12.002.

Andreas M Brandmaier. pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software*, 67(5):1–23, 2015. ISSN 1548-7660. 10.18637/jss.v067.i05. URL `http://www.jstatsoft.org/v67/i05/`.

W. Braun. The System Archetypes. *The Systems Modeling Workbook*, pages 1–26, 2002.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC Press, 1984. ISBN 978-0412048418.

Casey Brown, Yonas Ghile, Mikaela Laverty, and Ke Li. Decision scaling: Linking bottom-up vulnerability analysis with climate projections in the water sector. *Water Resources Research*, 48(9): 1–12, 2012. ISSN 00431397. 10.1029/2011WR011212.

Benjamin P. Bryant and Robert J. Lempert. Thinking inside the box: A participatory, computer-assisted approach to scenario discovery. *Technological Forecasting and Social Change*, 77(1): 34–49, 2010. ISSN 00401625. 10.1016/j.techfore.2009.08.002. URL `http://dx.doi.org/10.1016/j.techfore.2009.08.002`.

J. Cariboni, D. Gatelli, R. Liska, and A. Saltelli. The role of sensitivity analysis in ecological modelling. *Ecological Modelling*, 203(1-2):167–182, 2007. ISSN 03043800. 10.1016/j.ecolmodel.2005.10.045.

Ahlame Douzal Chouakria and Panduranga Naidu Nagabhushan. Adaptive dissimilarity index for measuring time series proximity. *Advances in Data Analysis and Classification*, 1(1):5–21, 2007. ISSN 18625355. 10.1007/s11634-006-0004-6.

Jonathan D Cryer. *Time Series Analysis*. Wadsworth, 2008. ISBN 9780387759586.

Marco Cuturi. Fast Global Alignment Kernels. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 929–936, 2011. ISBN 9781450306195. URL `http://www.iip.ist.i.kyoto-u.ac.jp/member/cuturi/Papers/cuturi11fast.pdf`.

S. Dalal, B. Han, R. Lempert, A. Jaycocks, and A. Hackbarth. Improving scenario discovery using orthogonal rotations. *Environmental Modelling and Software*, 48:49–64, 2013. ISSN 13648152. 10.1016/j.envsoft.2013.05.013. URL `http://dx.doi.org/10.1016/j.envsoft.2013.05.013`.

Paul K. Davis, Steven C. Bankes, and Michael Egner. *Enhancing Strategic Planning with Massive Scenario Generation - Theory and Experiments*. 2007. ISBN 9780833040176.

Sijbren de Jong, Willem L. Auping, and Joris Govers. *The Geopolitics of Shale Gas: The Implications of the US Shale Gas Revolution on Intrastate Stability within Traditional Oil- and Natural Gas-Exporting Countries in the EU Neighborhood*. The Hague Centre for Strategic Studies, The Hague, 2014.

David Casado de Lucas. *Classification Techniques for Time Series and Functional Data*. PhD thesis, Universidad Carlos III de Madrid, 2010.

George T Duncan, W I Gorr, and Janusz Szczypula. Forecasting analogous time series. *International Series in Operations Research and Management Science*, (412):195–214, 2001. 10.1007/978-0-306-47630-3_10.

John W. Emerson, Walton A. Green, Barret Schloerke, Jason Crowley, Dianne Cook, Heike Hofmann, and Hadley Wickham. The generalized pairs plot. *Journal of Computational and Graphical Statistics*, 22(1):79–91, 2013. ISSN 10618600. 10.1080/10618600.2012.694762.

David N. Ford. A Behavioral Approach to Feedback Loop Dominance. In *Proceedings of the 1998 System Dynamics Conference*, volume 1, page 33, 1998. ISBN 5628.

Jerome H. Friedman and Nicholas I. Fisher. Bump Hunting in High-Dimensional Data. *Statics and Computing*, 9:123–143, 1999. ISSN 0960-3174. 10.1023/A:1008894516817. URL `http://link.springer.com/article/10.1023/A:1008894516817`.

Pedro Galeano and Daniel Pella. Multivariate Analysis in Vector Time Series Pedro Galeano and Daniel Pella. *Resenhas, the Journal of the Institute of Mathematics and Statistics of the University of Sao Paolo*, 4:383–403, 2000.

M. D. Gerst, P. Wang, and M. E. Borsuk. Discovering plausible energy and economic futures under global change using multidimensional scenario discovery. *Environmental Modelling and Software*, 44:76–86, 2013. ISSN 13648152. 10.1016/j.envsoft.2012.09.001. URL `http://dx.doi.org/10.1016/j.envsoft.2012.09.001`.

Sebastiaan Greeven, Oscar Kraan, Émile J.L. Chappin, and Jan H. Kwakkel. The emergence of climate change mitigation action by society: An agent-based scenario discovery study. *JASSS*, 19(3), 2016. ISSN 14607425. 10.18564/jasss.3134.

David G. Groves and Robert J. Lempert. A new analytic method for finding policy-relevant scenarios. *Global Environmental Change*, 17(1):73–85, 2007. ISSN 09593780. 10.1016/j.gloenvcha.2006.11.006.

Céline Guivarch, Robert Lempert, and Evelina Trutnevyte. Scenario techniques for energy and environmental research: An overview of recent developments to broaden the capacity to deal with complexity and uncertainty. *Environmental Modelling and Software*, 97:201–210, 2017. ISSN 13648152. 10.1016/j.envsoft.2017.07.017.

M Haasnoot, J Schellekens, J J Beersma, H Middelkoop, and J. C.J. Kwadijk. Transient scenarios for robust climate change adaptation illustrated for water management in the Netherlands. *Environmental Research Letters*, 10(10), 2015. ISSN 17489326. 10.1088/1748-9326/10/10/105008.

Marjolijn Haasnoot, Hans Middelkoop, Astrid Offermans, Eelco van Beek, and Willem P.A. van Deursen. Exploring pathways for sustainable water management in river deltas in a changing environment. *Climatic Change*, 115(3-4):795–819, 2012. ISSN 01650009. 10.1007/s10584-012-0444-2.

Marjolijn Haasnoot, Jan H. Kwakkel, Warren E. Walker, and Judith ter Maat. Dynamic adaptive policy pathways: A method for crafting robust decisions for a deeply uncertain world. *Global Environmental Change*, 23(2):485–498, 2013. ISSN 09593780. 10.1016/j.gloenvcha.2012.12.006. URL `http://dx.doi.org/10.1016/j.gloenvcha.2012.12.006`.

Ronald A. Halim, Jan H. Kwakkel, and Lóránt A. Tavasszy. A scenario discovery study of the impact of uncertainties in the global container transport system on European ports. *Futures*, 81(October): 148–160, 2016. ISSN 00163287. 10.1016/j.futures.2015.09.004. URL `http://dx.doi.org/10.1016/j.futures.2015.09.004`.

Caner Hamarat, Jan H Kwakkel, and Erik Pruyt. Technological Forecasting & Social Change Adaptive Robust Design under deep uncertainty. *Technological Forecasting & Social Change*, 80(3):408–418, 2013. ISSN 0040-1625. 10.1016/j.techfore.2012.10.004. URL `http://dx.doi.org/`

10.1016/j.techfore.2012.10.004.

Satu Helske and Jouni Helske. Mixture Hidden Markov Models for Sequence Data: The seqHMM Package in R. 2017. URL http://arxiv.org/abs/1704.00543.

K.W. Hipel and Y. Ben-Haim. Decision making in an uncertain world: information-gap modeling in water resources management. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 29(4):506–517, 1999. ISSN 1094-6977. 10.1109/5326.798765. URL http://ieeexplore.ieee.org/ielx5/5326/17357/00798765.pdf?tp={&}arnumber=798765{&}isnumber=17357.

Kurt Hornik and Walter Böhm. clue: Cluster ensembles. R package version 0.3-53., 2017.

Tushith Islam and Erik Pruyt. Scenario generation using adaptive sampling: The case of resource scarcity. *Environmental Modelling and Software*, 79:285–299, 2016. ISSN 13648152. 10.1016/j.envsoft.2015.09.014. URL http://dx.doi.org/10.1016/j.envsoft.2015.09.014.

Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. Distance measures for effective clustering of ARIMA time-series. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 273–280, 2001. ISBN 0-7695-1119-8. 10.1109/ICDM.2001.989529. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=989529.

Joseph R. Kasprzyk, Shanthi Nataraj, Patrick M. Reed, and Robert J. Lempert. Many objective robust decision making for complex environmental systems undergoing change. *Environmental Modelling and Software*, 42:55–71, 2013. ISSN 13648152. 10.1016/j.envsoft.2012.12.007. URL http://dx.doi.org/10.1016/j.envsoft.2012.12.007.

Harold W. Kuhn. The Hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, volume 2, pages 29–47. 2010. ISBN 9783540682745. 10.1007/978-3-540-68279-0_2. URL http://doi.wiley.com/10.1002/nav.3800020109.

Jaap C. J. Kwadijk, Marjolijn Haasnoot, Jan P. M. Mulder, Marco M. C. Hoogvliet, Ad B. M. Jeuken, Rob A. A. van der Krogt, Niels G. C. van Oostrom, Harry A. Schelfhout, Emiel H. van Velzen, Harold van Waveren, and Marcel J. M. de Wit. Using adaptation tipping points to prepare for climate change and sea level rise: a case study in the Netherlands. *Wiley Interdisciplinary Reviews: Climate Change*, 1(5):729–740, 2010. ISSN 17577780. 10.1002/wcc.64. URL http://doi.wiley.com/10.1002/wcc.64.

Jan H. Kwakkel. The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. *Environmental Modelling and Software*, 96:239–250, 2017. ISSN 13648152. 10.1016/j.envsoft.2017.06.054. URL http://dx.doi.org/10.1016/j.envsoft.2017.06.054.

Jan H. Kwakkel. A generalized many-objective optimization approach for scenario discovery. In *Proceedings of the 8th International Congress on Environmental Modelling and Software*, 2018.

Jan H. Kwakkel and Marc Jaxa-Rozen. Improving scenario discovery for handling heterogeneous uncertainties and multinomial classified outcomes. *Environmental Modelling and Software*, 79:311–321, 2016. ISSN 13648152. 10.1016/j.envsoft.2015.11.020. URL http://dx.doi.org/10.1016/j.envsoft.2015.11.020.

Jan H. Kwakkel and Marc Jaxa-Rozen. Comparing many-objective robust decision making and multi-objective robust optimization for the lake problem. *Geophysical Research Abstracts*, 20, 2018.

Jan H. Kwakkel and Erik Pruyt. Exploratory Modeling and Analysis, an approach for model-based foresight under deep uncertainty. *Technological Forecasting and Social Change*, 80(3):419–431, 2013. ISSN 00401625. 10.1016/j.techfore.2012.10.005. URL http://dx.doi.org/10.1016/j.techfore.2012.10.005.

Jan H. Kwakkel, Willem L. Auping, and Erik Pruyt. Dynamic scenario discovery under deep uncertainty: The future of copper. *Technological Forecasting and Social Change*, 80(4):789–800, 2013. ISSN 00401625. 10.1016/j.techfore.2012.09.012. URL http://dx.doi.org/10.1016/j.techfore.2012.09.012.

Jan H. Kwakkel, Marjolijn Haasnoot, and Warren E. Walker. Developing dynamic adaptive policy pathways: a computer-assisted approach for developing adaptive strategies for a deeply uncertain world. *Climatic Change*, 132(3):373–386, 2015. ISSN 01650009. 10.1007/s10584-014-1210-4.

Jan H. Kwakkel, Marjolijn Haasnoot, and Warren E. Walker. Comparing Robust Decision-Making and Dynamic Adaptive Policy Pathways for model-based decision support under deep uncertainty. *Environmental Modelling and Software*, 86:168–183, 2016. ISSN 13648152. 10.1016/j.envsoft

.2016.09.017. URL http://dx.doi.org/10.1016/j.envsoft.2016.09.017.

Robert J. Lempert, Steven W. Popper, and Steven C. Bankes. *Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis*. 2003. ISBN 0833034855. 10.1016/j.techfore.2003.09.006.

Robert J. Lempert, David G. Groves, Steven W. Popper, and Steven C. Bankes. A General, Analytic Method for Generating Robust Strategies and Narrative Scenarios. *Management Science*, 52(4):514–528, 2006. ISSN 0025-1909. 10.1287/mnsc.1050.0472. URL http://pubsonline.informs.org/doi/abs/10.1287/mnsc.1050.0472.

Robert J. Lempert, Benjamin P. Bryant, and Steven C. Bankes. Comparing Algorithms for Scenario Discovery. 2008.

Donald Ludwig, Dixon Jones, and Crawford Holling. Qualitative Analysis of Insect Outbreak Systems : The Spruce Budworm and Forest. *The Journal of Animal Ecology*, 47(1):315–332, 1978.

Donella H. Meadows. The Limits to Growth. *The Club of Rome*, page 211, 1972. ISSN 1093-474X. 10.1111/j.1752-1688.1972.tb05230.x. URL http://doi.wiley.com/10.1111/j.1752-1688.1972.tb05230.x.

Donella H. Meadows. *Thinking in Systems: A Primer*. Taylor and Francis, 2012. ISBN 9781844077267.

Pablo Montero and José Vilar. TSclust: An R Package for Time Series Clustering. *JSS Journal of Statistical Software*, 62(1):1–43, 2014. ISSN 1548-7660. 10.18637/jss.v062.i01. URL http://www.jstatsoft.org/.

Douglas C Montgomery. *Time Series Analysis and Forecasting*. McGraw-Hill, 2016. ISBN 978-3-319-28723-2. 10.1007/978-3-319-28725-6. URL http://link.springer.com/10.1007/978-3-319-28725-6.

Ruben Moorlag, Willem L. Auping, and Erik Pruyt. Exploring the effects of shale gas development on natural gas markets: a multi-method approach. In *32nd International Conference of the System Dynamics Society*, 2014.

John Paparrizos and Luis Gravano. k-Shape: Efficient and Accurate Clustering of Time Series. *ACM SIGMOD*, pages 1855–1870, 2015. ISSN 07308078. 10.1145/2723372.2737793. URL http://dl.acm.org/citation.cfm?id=2723372.2737793.

Andrew M. Parker, Sinduja V. Srinivasan, Robert J. Lempert, and Sandra H. Berry. Evaluating simulation-derived scenarios for effective decision support. *Technological Forecasting and Social Change*, 91:64–77, 2015. ISSN 00401625. 10.1016/j.techfore.2014.01.010. URL http://dx.doi.org/10.1016/j.techfore.2014.01.010.

Valery Petrov, Vilmos Gáspár, Jonathan Masere, and Kenneth Showalter. Controlling chaos in the Belousov—Zhabotinsky reaction. *Nature*, 361(6409):240–243, 1993. ISSN 0028-0836. 10.1038/361240a0. URL http://www.nature.com/doifinder/10.1038/361240a0.

D Piccolo. A distance measure for classifying ARMA models. *Journal of Time Series Analysis*, 11(2):153–163, 1990. ISSN 01439782. 10.1111/j.1467-9892.1990.tb00048.x.

Erik Pruyt, Thomas Logtens, and Govert W. Gijsbers. Exploring Demographic Shifts: Aging and Migration Exploratory Group Model Specification & Simulation. pages 1–27, 2011.

Chotirat Ann Ratanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das. Mining Time Series Data. *Data Mining and Knowledge Discovery Handbook*, pages 1049–1077, 2010. ISSN 14337851. 10.1007/978-0-387-09823-4_56. URL http://dx.doi.org/10.1007/978-0-387-09823-4{_}56.

Horst W J Rittel and Melvin M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, 1973. ISSN 00322687. 10.1007/BF01405730.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C):53–65, 1987. ISSN 03770427. 10.1016/0377-0427(87)90125-7.

Julie Rozenberg, Stéphane Hallegatte, Adrien Vogt-Schilb, Olivier Sassi, Céline Guivarch, Henri Waisman, and Jean Charles Hourcade. Climate policies as a hedge against the uncertainty on future oil supply. *Climatic Change*, 101(3):663–668, 2010. ISSN 01650009. 10.1007/s10584-010-9868-8.

Andrea Saltelli and Paola Annoni. How to avoid a perfunctory sensitivity analysis. *Environmental Modelling and Software*, 25(12):1508–1517, 2010. ISSN 13648152. 10.1016/j.envsoft.2010.04.012. URL http://dx.doi.org/10.1016/j.envsoft.2010.04.012.

Alexis Sarda-Espinosa. Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package, 2017. URL https://cran.r-project.org/web/packages/dtwclust/

vignettes/dtwclust.pdf.

Steven H. Strogatz. Nonlinear Dynamics and Chaos, 1994.

B. C. Trindade, P. M. Reed, J. D. Herman, H. B. Zeff, and G. W. Characklis. Reducing regional drought vulnerabilities and multi-city robustness conflicts using many-objective optimization under deep uncertainty. *Advances in Water Resources*, 104:195–209, 2017. ISSN 03091708. 10.1016/j.advwatres.2017.03.023. URL http://dx.doi.org/10.1016/j.advwatres.2017.03.023.

José A Vilar and Sonia Pértega. Discriminant and cluster analysis for Gaussian stationary processes: Local linear fitting approach. In *Journal of Nonparametric Statistics*, volume 16, pages 443–462, 2004. 10.1080/10485250410001656453.

T. Warren Liao. Clustering of time series data - A survey. *Pattern Recognition*, 38(11):1857–1874, 2005. ISSN 00313203. 10.1016/j.patcog.2005.01.025.

D. W. Watkins and D. C. McKinney. Robust optimization for incorporating risk and uncertainty in sustainable water resources planning. *IAHS Publications-Series of Proceedings and Reports-Intern Assoc Hydrological Sciences*, 231:225–232, 1995.

Saskia E Werners, Stefan Pfenninger, Erik van Slobbe, Marjolijn Haasnoot, Jan H Kwakkel, and Rob J Swart. Thresholds, tipping and turning points for sustainability under climate change, 2013. ISSN 18773435.

Norbert Wiener. *Cybernetics or Communication and Control in the Animal and the Machine*. The MIT Press, Cambridge, MA, 1961.

Gönenc Yücel and Yaman Barlas. Automated parameter specification in dynamic feedback models based on behavior pattern features. *System Dynamics Review*, 27(2):195–215, 2011. 10.1002/sdr.457.

Hui Zhang, Tu Bao Ho, Yang Zhang, and Mao-song Lin. Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform. *Informatica*, 30:305–319, 2006. ISSN 03505596.

# List of Figures

# List of Tables

# A

# Software and Packages

I used a wide variety of computer tools for this research. To aid reproducibility, the following tables list all used modelling and analysis software, packages and versions. As all these packages are subject to change, I can only guarantee reproducibility for these specific versions. Note that many of these packages have their own dependencies, which are not listed below. I only list time series clustering and analysis-specific packages, and will not mention common tools for visualization, data storage or other similarly unspecific purposes. All work was conducted on a personal computer using the Windows 10 64bit Professional operating system.

## A.1. Programs

| **Program** | **Usage** | **Version** |
|---|---|---|
| Python | General programming language | 3.5.1.0 |
| Anaconda Navigator | Python package manager and distribution for data science | 1.7 |
| R | Mathematics and statistics programming language | 3.5.0 |
| RStudio | R interface | 1.1.4.47 |
| Vensim | System dynamics modelling software | 7.3 |

## A.2. Python Packages

| **Package** | **Usage** | **Version** |
|---|---|---|
| ema-workbench | Exploratory modelling and analysis toolkit | 1.1.4 |
| feather-format | Standardized data transfer between Python and R | 0.3.1 |
| scipy | Data science and analysis | 1.0.0 |

## A.3. R Packages

| Package | Usage | Version |
|---|---|---|
| clue | Cluster ensemble functions | 0.3-55 |
| clv | Cluster validation functions | 0.3-2.1 |
| dtw | Dynamic time warping algorithms | 1.18-1 |
| dtwclust | Time series clustering with dynamic time warping algorithms | 5.3.1 |
| feather | Standardized data transfer between R and Python | 0.3.1 |
| infotheo | Information-theoretic measures and functions | 1.2.0 |
| pdc | Permutation distribution clustering of time series | 1.0.3 |
| seqHMM | Data analysis with hidden Markov models | 1.0.8-1 |
| tidyverse | Data science and analysis ecosystem | 1.2.1 |
| TSclust | Time series clustering functions | 1.2.4 |

# B

# Online Code Repository

I am making all code used to conduct this research publicly available through my GitHub account, `http://github.com/steipatr/`. The repository is structured around the three cases described in this thesis. For each case, I provide the relevant Vensim model (or completed data set, for the shale gas case), and the associated R and Jupyter code. Furthermore, I provide an R file containing the custom functions I developed.

As the analysis was conducted in both R and Python, it is necessary at times to transfer data between the two environments. In an effort to ensure repository documentation, the intended order of execution for the various analysis steps is described online for each case separately. Similarly, data import and export requirements are indicated within each script at the appropriate point.