Anomaly Classification Through Automated Shape Grammar Representation

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Mechanical Engineering

Mark E. Whiting

B.Des., Industrial Design, Royal Melbourne Institute of Technology
M.S., Industrial Design, Korea Advanced Institute of Science and Technology

Carnegie Mellon University
Pittsburgh, PA

August, 2017

Acknowledgements

This thesis is dedicated to:

- my wife, Mani, who has supported me despite my time consuming hobbies;
- my parents, Mary and Bernard, and my sisters, Ruth and Bridget, who put up with me doing research instead of visiting them;
- my advisors, Jonathan Cagan, and Philip R. LeDuc, who have been the driving force behind my work and have allowed me to follow the winding path I preferred during my PhD;
- my committee, Manuela M. Veloso and Yongjie Jessica Zhang;
- Don Norman and Michael Bernstein, who have both, at different times, driven me to rethink how I do research;
- my many colleagues, especially Ardon Z. Shorr, Chris McComb, Kosa Goucher-Lambert, Kyle Justus, Paul Egan, Daniel Clymer, Sun-Joong Kim, and many others;
- the CMU community, especially the students of the Mechanical Engineering Department and my colleagues in the Graduate Student Assembly;
- my data partners at UPMC, Joe Mettenburg and Enrico Novelli and their teams and patients;
- and to my sources of funding, the National Science Foundation, under grants CMMI-1160840 and CMMI-1233864, and the Pittsburgh Infrastructure and Technology Alliance.

Abstract

Statistical learning offers a trove of opportunities for problems where a large amount of data is available but falls short when data are limited. For example, in medicine, statistical learning has been used to outperform dermatologists in diagnosing melanoma visually from millions of photos of skin lesions. However, many other medical applications of this kind of learning are made impossible due to the lack of sufficient learning data, for example, performing similar diagnosis of soft tissue tumors within the body based on radiological imagery of blood vessel development. A key challenge underlying this situation is that many statistical learning approaches utilize unstructured data representations such as strings of text or raw images, that don't intrinsically incorporate structural information.

Shape grammar is a way of using visual rules to define the underlying structure of geometric data, pioneered by the design community. Shape grammar rules are replacement rules in which the left side of the rule is a search pattern and the right side is a replacement pattern which can replace the left side where it is found. Traditionally shape grammars have been assembled by hand through observation, making it slow to use them and limiting their use with complex data. This work introduces a way to automate the generation of shape grammars and a technique to use grammars for classification in situations with limited data.

A method for automatically inducing grammars from graph based data using a simple recursive algorithm, providing non-probabilistic rulesets, is introduced. The algorithm uses iterative data segmentation to establish multi scale shape rules, and can do so with a single dataset. Additionally, this automatic grammar induction algorithm has been extended to apply to high dimensional data in a non-visual domain, for example, graphs like social networks. We validated our method by comparing our results to grammars made of historic buildings and products and found it performed comparably grammars made by humans.

The induction method was extended by introducing a classification approach based on mapping grammar rule occurrences to dimensions in a high dimensional vector space. With this representation data samples can be analyzed and quickly classified, without the need for data intensive statistical learning. We validated this method by performing sensitivity tests on key graph augmentations and found that our method was comparably sensitive and significantly faster at learning than related existing methods at detecting graph differences across cases.

The automated grammar technique and the grammar based classification technique were used together to classify magnetic resonance imaging (MRI) of the brain of 17 individuals and showed that our methods could detect a variety of vasculature borne condition indicators with short and long-term health implications. Through this study we demonstrate that automated grammar based representations can be used for efficient classification of anomalies in abstract domains such as design and biological tissue analysis.

Table of Contents

Tables

Figures

Introduction

Statistical based models and symbolic based models are two broad categories in which a majority of work in making artificially intelligent systems can be placed (Minsky, 1991). Symbolic models hinge on the idea of formally extracting structured information from the real world, while statistical approaches, often categorized as machine learning, use observed patterns' repetition to make probabilistic guesses about the nature of the underlying information. Both methods have significant strengths, but fall prey to critical weaknesses too. Machine learning's reliance on probabilistic models generally means that it requires more data to make inferences, and that it can have difficulty in making sense of rare occurrences in data. On the other hand, symbolic approaches require a significant amount of manual structuring of data to help an automated system make sense of the world. In this work, we explore one way to ally these approaches through an automated data structuring preprocess, which makes statistical learning more accessible for small datasets that have been structurally represented.

Many domain specific data structuring techniques exist, however, one that has found prevalence in a diversity of fields is the concept of grammars, formalisms relating information's structure through rules about types of information (Wand and Weber, 1993). The most straightforward demonstration of this is considering the grammar of the English language, where almost all sentences adhere to the general structure of a *subject*, followed by a *verb*, and terminated by an *object*. Each of these types of term or phrase represents a diversity of words or combinations of words that afford rich and expressive communication about the entirety of the human experience. The concept of grammar has also been applied through shape grammar in design (Stiny and Gips, 1972), graph grammar in engineering (Rozenberg, 1997), as well as more abstract formalisms such as pattern languages used first in architecture (Alexander, Ishikawa and Silverstein, 1977) and later influencing modern programming paradigms (Beck and Cunningham, 1987). The shared underlying notion is that by defining a vocabulary

around well-structured rules, a virtually unlimited number of potential outcomes can be generalized, and studied with significant efficiency. For instance, as we start to read a sentence we can make sophisticated guesses about how it will end, to the point that we have a strong tendency to interrupt one another during verbal discourse.

A caveat with grammar representations is that many of the more abstract uses of them, such as in design, have not been adequately automated (Gips, 1999). Processing shape and graph grammars manually is laborious, and it seems likely this has led to reduced applications of these methods outside of academia.

Thesis

Grammar based representations can be automated and used for efficient classification of anomalies in abstract domains such as design and biological image analysis.

To evaluate this thesis, Chapter 3 introduces a general purpose recursive grammar induction algorithm, demonstrating that grammar based representations can be automatically induced for graph based data. Chapter 4 establishes a rule frequency based representation to leverage grammars in classification, while needing less training data than standard methods require, showing efficient classification of anomalies in design. Chapter 5 applies this method in detecting conditions indicated by brain vasculature in MRI of sickle cell anemia patients, demonstrating grammar methods being used to aid in biological image analysis.

This dissertation aims to introduce a generalizable approach for using grammatical formalism with data from the real world, in order to create useful abstractions that reflect critical structure, thereby making other types of inference more accessible. This approach relies on data that has some structural integrity, and data that can be segmented or isolated in such a way that that integrity can be readily observed.

Contributions

To achieve the desired goal, several key steps have been taken which constitute 4 main contributions of this work, distributed through chapters 3, 4, and 5.

Automated grammar induction

Chapter 3 introduces a flexible and generalizable method for extracting grammars from data which can have a graph structure. The method is presented in the context of shape grammars and graph grammars, as means for abstracting rules about designs from visual information or from structured data about a design. Existing work has automated grammar induction for linear graphs, for statistically derivable rules, but has not provided a general purpose approach, as this work does.

Grammar induction heuristics

Chapter 3 also introduces a set of 4 heuristics to evaluate a grammar induction process, based on historical perspectives on the use of grammar to represent style in design. The heuristics measure the accuracy, variability, repeatability and conciseness of a given grammar, as a result of an induction process from a body of data. With this method, grammar induction processes can be quantitatively compared. Prior work has introduced measures for grammar effectiveness in specific cases, but have lacked a holistic approach allowing for comparison of induced grammars between cases, which is enabled with the introduced heuristics.

Grammar based classification

In Chapter 4, a method for utilizing grammar rule frequency as a feature for classification is introduced. With this approach, a small amount of training data can be used to learn robust differences between

differing classes of data. When combined with the automated grammar induction methods introduced in Chapter 3, the method can be conducted with an almost completely automated end to end pipeline. The method is validated against a statistical learning baseline for anomaly detection where it reaches comparable levels of accuracy with significantly less training data. The method is also validated in distinguishing design differences by categorizing spline networks representing a variety of cars into their respective automotive form factor categories.

Applying automated grammar based classification to brain vasculature for diagnosis

In chapter 5, the methods from Chapters 3 and 4 are applied to time of flight (TOF) MRI scans of 17 sickle cell anemia patients to show sensitivity in detecting indicative rules correlated with current conditions and predicted health outcomes. A pipeline for converting TOF MRI into a graph which can be used to build a grammar is utilized. The classification approach is then conducted, producing a rule frequency based representation of each patient. Patient conditions are correlated with rule frequency distributions to provide a reliable predictive model with only the 17 brain scans.

Thesis overview

The disstertation includes a general related work section in Chapter 2 which introduces the key concepts used in the rest of the work and positions the work in the context of existing related attempts.

Chapter 3 introduces the core automated grammar induction algorithm used by the rest of the work.

Chapter 4 applies the grammar induction technique for classification, which is then further expanded in Chapter 5 with a specific application in medical image analysis.

Chapter 6 discusses the limitations of the current work, questions yet to be answered in this line of work, and the broader implications of utilizing intermediary structured representations in design and in other contexts where data is scarce and artificial intelligence features would be salient.

Finally, Chapter 7 reiterates the contributions of the work and concludes the dissertation.

Related Work

This chapter gives context to the variety of topics discussed in the rest of the dissertation. First background on shape and graph grammar methods will be provided, describing how those methods have been used, and where they have fallen short. Next details on inducing structural knowledge will be outlined, based on work in the field such as machine translation, computational design, and graph analysis. Following this, a discussion of computational methods around extracting information from frequency based representations of information will be introduced, and finally, the medical imaging techniques this work leverages will be outlined.

Grammars

A shape grammar (Gips, 1975; Stiny, 1980) is a set of rules that define shape transformations. In aggregate, these rules create a system of transformations which can be used together to create particular visual outputs. These are generally constructed in a planned way in order to demonstrate a design (Stiny and Gips, 1972), or are constructed retroactively from a design, with the intention of analyzing it in some way (Pugliese and Cagan, 2002; McCormack, Cagan and Vogel, 2004; Trescak, Rodriguez and Esteva, 2009).

The convention is to visualize the rules in a shape grammar as a current state on the left hand side of the rule (LHS), which can be transformed into an alternative state on the right hand side of the rule (RHS)(Stiny, 1980). This simple representation affords significant flexibility because grammar rules can often be applied repeatedly, and at a variety of scales. This convention will be used for all the types of grammars dealt with in this dissertation.

The basic idea of a shape grammar can also be extended by introducing parametric rules (Stiny and Mitchell, 1978), in which rules can encompass variables, such as overarching symmetry lines, or

numbers or sizes of particular features in a transformation. Another common extension to the fundamental idea of the shape grammar are rules which contain other rules, meta-rules (Sondheimer and Weischedel, 1980). Meta-rules enable complex and recursive behavior with grammars. These two approaches enable significantly more sophisticated emergence to be captured by grammar systems as they allow for decision making at the time of use of a grammar to have substantial and ongoing impacts on the outcomes. Because of the flexibility inherent in meta-rule systems, this work utilizes grammars with this feature, however, it does not explicitly deal with parameterizing grammar rules. The specific reason for this decision is that for most situations, parameterizations allow for fewer rules that are somewhat more easy for humans to use, however, since this work focuses on computerized implementations of grammars, this is not necessary, so more rules, and meta-rules can end up serving the same purpose. One nuanced difference is that meta rule systems do not inherently afford scale variability in grammar rule use. As a consequence, this work assumes scale variability on all grammar implementations.

An alternative formalism of grammars is the graph grammar (Rozenberg, 1997), which is conceptually similar, however applying to ontological relationships in graphs, as opposed to shapes. Graph grammars are used in engineering to deal with diagrams of functional and structural relationships (Schmidt and Cagan, 1997; Schmidt and Chase, 2000) as well as domains less similar to design, where structured information may be represented in a graph form, such as in providing legal analysis and filtering (Pinkwart *et al.*, 2006). Graph grammars, are a relatively more general form of grammar because they don't require any implicit representation, and are thereby incredibly flexible. This work operates under the assumption that shape grammars can be recapitulated with graph grammars by treating the nodes in a graph ontology as shapes, and assuming that an edge in that graph is a shared position. This assumption allows the grammar manipulation and analysis tools introduced in this work to be more broadly applied,

to the extent that several of the experiments conducted in this work utilized structured graphs as challenging datasets that can be quickly generated.

Inducing Grammars

One of the biggest remaining challenges underlying the use of shape and graph grammars is making them from existing structured data (Gips, 1999). This process is often referred to as grammar induction, because it requires using inductive reasoning to establish rules that are likely to capture aspects of the existing data. In this way the existing data provides a statistical ground truth for which rules should exist.

Traditionally shape grammar induction has been done manually and is a painstaking process because rules must be established and duplicates must be removed by hand (Speller, Whitney and Crawley, 2007). More recently a variety of approaches for making this easier and more automatic have been introduced. For example, statistical methods have been used to find likely grammar rules based on coded splines (Orsborn, Cagan and Boatwright, 2007), using visual design data with structure tags (Talton *et al.*, 2012), and using image based analysis applied in machine vision (Teboul *et al.*, 2010, 2011). This dissertation introduces a method for extracting this information from a graph representation, avoiding the need for these more data intensive statistical approaches.

Graph grammar induction has been explored more successfully, in part because many graph problems can be treated as lingual grammar problems, which are less complicated on the basis that language can be represented as a fixed degree linear graph (Ding and Palmer, 2005). For example, Sequitur (Nevill-Manning and Witten, 1997) was introduced as a grammar induction algorithm for arbitrary linear graphs. In this dissertation an extension to Sequitur is introduced in which repeated explorations of a graph can lead to an approach for inducing grammars without the linear restriction.

The domain of data mining has also led to the introduction of a variety of techniques for graph grammar induction style representations (Jiang, Coenen and Zito, 2013), in which graph based information is made simpler by applying subgraphs (Kuramochi and Karypis, 2001), and these subgraphs can be a tool for analysis (Yan and Han, 2002). For example, Subdue (Holder, 1994) was introduced as a method for learning rules and meta-rules from a graph for the purposes of compression and comparison. The induction method introduced in this dissertation provides additional flexibility compared with these approaches, in that it affords induction of graphs where structure is not explicitly defined in the representation, but instead, implicit to the nature of the data being induced, such as is the case with applications of grammars to design.

A further challenge in inducing grammars is the fact that comparing grammars' effectiveness is context dependent. For instance, shape grammars are used to encompass style (Stiny and Mitchell, 1978), though this is not an explicitly defined characteristic. Computational approaches have been introduced for evaluating groups of rules for generative design and determining rule quality (Königseder and Shea, 2014), however, this approach does not introduce a formalism by which style can be evaluated. A contribution of this work is a principled approach to the interpretation of grammar in representing in order to help refine grammar induction techniques. Although this heuristic can be evaluated on any grammar induction process, it is most relevant for use with grammar induction with a focus on preserving stylistic notions intrinsic to the structure of a structured dataset.

Anomaly detection with grammars

One of the applications of grammars explored by this work is in using them to detect anomalies. Grammar methods have been applied for classification of differences in an abstract setting (Cook and Holder, 1994; Eberle and Holder, 2007). Anomaly detection more generally, has seen broad spectrum use (Chandola, Banerjee and Kumar, 2009), such as social network analysis (Savage *et al.*, 2014) and in

other graphs (Akoglu, Tong and Koutra, 2015). In this work, an approach to using grammars for anomaly detection is introduced that leverages distance metrics and frequency based characterizations. The introduced approach is also benchmarked against Oddball (Akoglu, McGlohon and Faloutsos, 2010), an anomaly detection algorithm based on an ensemble of subgraph indicators, which will be elucidated in chapter 4.

Distance metrics use a heuristic to compute the distance in some space between cases. Early applications include detecting degree of differences between biological samples based on genetic feature maps (Hattemer, 1982); this technique has also seen use in measuring the distances between designs analogically (Christensen and Schunn, 2007). These techniques rely on vector spaces, which can be formalized as Hilbert spaces (Wootters, 1981), and afford quick computation of distance metrics such as the Euclidean distance. A significant benefit of these approaches is the relative simplicity by which they can be computationally interpreted. For this reason, this dissertation utilizes a distance metric based approach to facilitate analysis of grammar classification problems.

The frequency distributions of features found in cases is another important indicator which has been used for classification (Hodge and Austin, 2004; Akoglu, Tong and Koutra, 2015). Techniques such as frequent subgraph mining (Kuramochi and Karypis, 2001; Jiang, Coenen and Zito, 2013), often rely on this modality, for example gSpan (Yan and Han, 2002) and AutoPart (Chakrabarti, 2004) are two established approaches in this domain. These kinds of approaches offer compelling support for inducing structure as an initial or intermediary step when performing data analysis, a theme which this work leverages with grammar, as a structural intermediary representation of many kinds of structured data. This dissertation moves beyond these existing approaches by focusing on grammar, in particular shape grammar, as a representation that can encompass important structural, morphological and ontological details when performing analysis.

Bag-of-words and bag-of-features models (Csurka *et al.*, 2004) extend this concept by leveraging the assumption that relationships of elements in the data are not critical to effective detection. However, semi structured representations provide added insight that can have improved results with these tools in text mining (Wallach, 2006). A contribution of this work is to leverage the principle of feature frequency in the context of assessing differences using grammars rules as features with particular frequencies in observed data.

Medical Image analysis

Medical image analysis conducted by radiologists is a domain ripe for disruption though the introduction of automated approaches. For example, human dermatological diagnosis has recently been outperformed by a deep learning system (Esteva *et al.*, 2017).

Structured information in medical imagery has been a point of interest as a precursor to more automated methods. For example, methods have been introduced to isolate blood vessel graphs throughout the body and in the brain (Kaus *et al.*, 2001; Bullitt *et al.*, 2005). This dissertation leverages this capability to induce grammars of blood vessel networks in patients' brains from MRI scans.

Particular methods for segmenting medical imagery such as magnetic resonance imaging (MRI) scans into structured graphs of splines or surfaces are diverse depending on the desired features and modality of imagery (Clarke *et al.*, 1995). However, in practice, simple techniques involving extracting threshold boundaries and coalescing them to splines to indicate blood vessels are used with a high degree of reliability for vascular segmentation (Helmberger *et al.*, 2014). In this work, a similar approach is adopted for segmenting scans of brains because of its robustness and simplicity.

Extended related works

This chapter has introduced literature from many of the topic areas covered by this dissertation. However, more focused context, describing particular issues, methods, and literature based considerations will be provided in the next 3 chapters when describing the key methods that the dissertation introduces.

Efficient Probabilistic Grammar Induction for Design

Abstract

The use of grammars in design and analysis has been set back by the lack of automated ways to induce them from arbitrary structured datasets. Machine translation methods provide a construct for inducing grammars from coded data which have been extended to be used for design through pre-coded design data. This work introduces a 4 step process for inducing grammars from un-coded structured datasets which can constitute a wide variety of data types, including many used in design. The method includes 1) extracting objects from the data, 2) forming structures from objects, 3) expanding structures into rules based on frequency and 4) finding rule similarities that lead to consolidation or abstraction. To evaluate this method, grammars are induced from generated data, architectural layouts and 3D design models to demonstrate that this method offers usable grammars automatically which are functionally similar to grammars produced by hand.

Introduction

Grammars are a proven tool for the representation, generation, and analysis of design (Stiny, 1980), however, difficulty in producing grammars has hindered their widespread adoption outside of academia. The key challenges of using shape grammars—*generation*, *parsing*, and *inference* (Gips, 1999)—have been partially addressed by literature over the past 20 years in the form of grammar interpreters (Piazzalunga and Fitzhorn, 1998; McCormack and Cagan, 2002; Trescak, Rodriguez and Esteva, 2009), but automated grammar generation through inference programs remains largely unsolved for design grammars (Chau and Chen, 2004). A foremost challenge of automatically inducting grammars from a suitable dataset is that this process is of combinatorial complexity (Yue and Krishnamurti, 2013) and thus takes impractically long. Intuitively, forming a shape or graph grammar is a process of breaking a

target graph down into a manageable set of morphological rules. To simply reproduce a graph is easily feasible, but to do so while maintaining subtlety of style or meaning is difficult, and to do it provably optimally requires considering every plausible rule and determining which minimal set demonstrates the grammar best, which makes it computationally intensive.

Traditionally, grammar rules have been developed by hand and utilized via grammar interpreters (Gips, 1999), which offer ways to apply grammar rules in order to achieve specific design outputs. Grammar inference, or grammar induction, requires developing a plausible set of rules, which may then be applied as needed to represent candidate designs (Gips, 1999). Interpreters either automatically (Orsborn and Cagan, 2009), or through user support, execute judgments about the coherent use of rules (Trescak, Esteva and Rodriguez, 2012). Inference, on the other hand, requires a relatively reversed judgment to be made, the process of determining where one rule stops and the next rule starts.

Language translation is limited by many of the same challenges as developing grammars for design (Gips, 1999). However, due to the comparative linearity of language models, approaches for learning lingual grammars have existed for some time (Berwick and Pilato, 1987). The modern approach to machine translation involves extracting a grammar and lexicon for both the starting and final languages and constructing a transform from one to the other (Sánchez-Martínez and Pérez-Ortiz, 2010; DeNero and Uszkoreit, 2011). For example, when a sentence is translated from one language to another, grammatical models are used to improve translation quality significantly. Services that provide translation features use a number of other machine learning techniques to reduce errors and improve the speed of computation for this process (Schwenk, 2012; Mikolov, Le and Sutskever, 2013), but in essence, the practice of building a grammar and executing it is a quintessential step in effective translation. As a result, there are various mechanisms for inducting lingual grammars and ontologies that have been made efficient enough for practical use through years of refinement by industry players (Balahur and Turchi, 2014). For example, many popular translation tools use language models that are

constantly updated based on popular usage to improve accuracy and efficiency around new uses of language and jargon. Additionally, advanced methods of chunking have been developed to improve the performance of such systems (Kudo and Matsumoto, 2002; Lee and Wu, 2007).

Learning from these advances in machine translation and their applications to design (Talton *et al.*, 2012), this chapter introduces a means for efficiently inducing grammars for design by using a probabilistic chunking method (Lee and Wu, 2007) paired with a multi scale random walk based graph exploration approach (Kang, Tong and Sun, 2012) to accommodate the complexities required in inducing design representations. The probabilistic approach does not reduce computational efficiency in the worst case, a random graph, but drastically reduces it in most cases likely to be assessed with grammars because chunk recognition can be done at a rate that is significantly faster than exhaustive searches, and the random walk allows for many discrete explorations of the graph without a time intensive exhaustive search in most cases (Fouss *et al.*, 2007).

To evaluate the success of the introduced method, experiments were conducted assessing four key attributes of each graph: accuracy, variability, repeatability and conciseness. The experiments utilized several datasets including rule based unidimensional graphs, rule based multidimensional graphs, historic buildings with well-established hand-made shape grammar representations, and a greenhouse design with component wise specificity. Rule based graphs were used to demonstrate basic effectiveness, while real world examples demonstrate specific applications of the induction method as well as alignment with human based grammar induction results.

The remainder of this chapter will outline the related work in machine translation, graph isomorphism and grammar induction that lead to these results, the specific details of the proposed grammar induction method and its evaluation across 3 experiments.

Related Work

The difference between building grammars for translation and building grammars for design is that in general design problems do not use one-dimensional data structures, such as those of written languages, and don't have clear representations, such as letters and words in written languages. Instead, shape grammars (Stiny and Mitchell, 1978; Stiny, 1980) use rules that refer to shape elements such as points, lines and surfaces to visually represent designs, and Graph grammars (Rozenberg, 1997; Schmidt and Cagan, 1997), which refer to sets of rules that adjust a graph's ontology introducing nodes or edges. The sophistication of grammars used in design over those of language is important in several specific ways. Most directly, more complex representations introduce new ways to parse the data to be induced into a grammar, because where in text every word is immediately preceded by at a maximum one word, and tailed by at a maximum one word, in a shape design or graph context, any element or node could be connected to a large number of other elements, and this must be accommodated to build a complete grammar.

Graph Similarity

Another aspect that makes building grammars for graphs challenging is that checking if two rules are the same is computationally intensive, and becomes exponentially harder as the average degree of the graph increases. In other domains, this problem is known as a subgraph isomorphism problem (McKay and Piperno, 2014), in which the task is to efficiently compare two subgraphs and determine if they are the same graph. Figure 1 shows two graphs that are identical in terms of connection but completely different in terms of visual arrangement. These graphs are isomorphic, but do not appear so. Graph isomorphism is widely considered non-deterministic polynomial-time hard (NP hard) (though recent work in this field intends to show that it may be polynomial) (Babai, 2015). The current state of the art approach to

computing solutions to this type of problem are on the order of $2^{O(sqrt(n\ log(n)))}$ (Babai, Kantor and Luks, 1983).

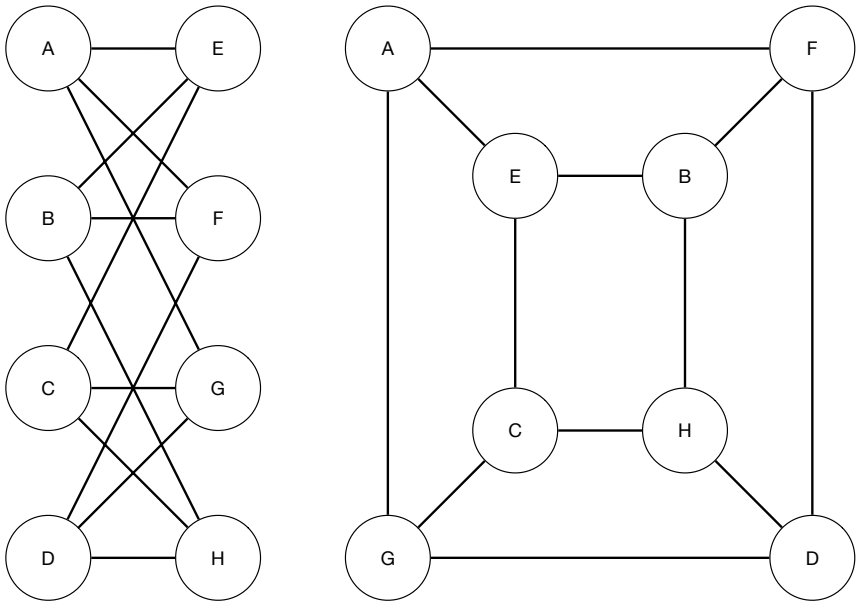


Figure 1 Graph Isomorphism. Graph G and Graph H are identical sets of nodes and edges and hence isomorphic, but visually appear very different.

Humans can process higher level similarities intuitively, so when building design grammars, they usually have the advantage of not needing to spend time to determine if two parts of an ontology are identical using brute force (Speller, Whitney and Crawley, 2007). Computational analysis does not have this intuitive advantage, so, for example, determining if two car door moldings from opposite sides of a car are the same door or a different door requires an extensive search involving finding an orientation in which similarity and symmetry are matched and then evaluating on a per node basis if the two parts are similar. With shape grammars the nodes might be sets of vertexes or B-Splines (Chau and Chen, 2004) while for graph grammar representations the nodes are some other kind of data object representation depending on the context (Rozenberg, 1997). Additionally, when considering richly connected

ontologies, humans also have difficulty making intuitive sense of the similarities of graphs, as Figure 1 demonstrates.

Inducing Grammars Without Coded Data

Formulating formal grammars is a critical step in machine translation (Ding and Palmer, 2005). However, language data is full of examples of misuse, so probabilistic grammars can be used when formal representations can't be computed. In these situations, a field of representations is used with probability weightings to generate systematic awareness of which cases are most likely to be the ones that should be formalized (Stolcke and Omohundro, 1994). Approaches like these afford inducing grammars from datasets that exhibit specific properties such as an annotated vocabulary, but cannot effectively support inducing grammars from generalized data; in other words, they do not resolve situations with no syntactical coding. As a result, they are not particularly suitable for making grammars in design more available or powerful at this time, because they would require input data with some structural coding, but may become useful as more dynamic approaches emerge. An example of this is demonstrated by Talton et al. (Talton *et al.*, 2012), showing that with coded design data, in which elements are functionally or stylistically categorized before evaluation, machine translation methods can be effectively used to generate grammars.

In addition to the reasons mentioned above, generating grammars automatically requires assessing many possible rules, and there are few metrics for determining the single best grammar out of a set of possible grammars (Königseder and Shea, 2015). As a result, high-level sketches for automated shape grammar induction have been proposed (Gero, 1994; Schnier and Gero, 1996; Gips, 1999), and some parts of the process have been automated with statistically generated rules (Orsborn and Cagan, 2009; Talton *et al.*, 2012). Additionally, context sensitive graph grammar induction has been elucidated through VEGGIE

(Ates and Zhang, 2007). However, these too do not sufficiently solve the issue for design grammar induction on data without semantic interpretability, which remains a major underlying challenge.

Evaluating Grammar Generation

In design, a grammar's effectiveness can be judged first by its ability to communicate a design's style, and then by its ability to do so in an optimal way, from a computational performance perspective. As a mechanism for conveying style, there are at least 3 critical previsions for effectiveness, adapted from Stiny & Mitchell (Stiny and Mitchell, 1978): 1) it should clearly convey the common elements of stylistically similar artifacts, 2) it should be complete enough to facilitate determining if an artifact is within a style or not, and 3) it should be usable to generate new artifacts that adhere to the style but were not part of the style corpus.

The *accuracy* of a grammar, defined as how well the grammar affords representing the input materials used to establish it, is intended to evaluate Stiny & Mitchell's first point above. However, quality of representation is essentially binary with non-probabilistic grammars because there is no margin of error – it's either correct, or not correct – so for this work, accuracy is assumed to be a requirement of any grammar. Without this feature, the grammar is not even reliable enough to reproduce its own input and it should be rejected. Accuracy reports the percent of instances that a grammar induction system can produce its input. This is different from saying that the grammar is entirely stylistically accurate, because it can only be accurate enough to represent the data it is given, and it is important to note that accuracy is not binary if a grammar incorporates probabilistic representations, which is not focused on in this work.

The *variability* of a grammar, on the other hand, interpreting Stiny & Mitchell's third point, is defined as how well a given grammar can be used to generate new examples of the style or construction rules embedded in the grammar. Again, with non-probabilistic grammars, a grammar either offers variability,

or does not, so this will be considered a necessary condition for a grammar to be accepted. Variability reports the percent of instances of a grammar induction system that can produce alternatives to their input.

Another aspect of a grammar system is the expectation that a similar grammar could be achieved from different observed artifacts. A practical example of this is if one builds a lingual grammar for English from two different instances of the New York Times Newspaper, there would be some hope that the contents of the two grammars would be similar and functionally interchangeable. We term this as the *repeatability* of a grammar, or the likelihood that grammars A and B will be identical when grammar B is learnt based on material generated from grammar A. In practice, grammar variability means that most data can be represented by many different valid grammars, so it is better to require that there is a grammar B in the set of grammars learnt from material generated by grammar A, such that grammar A and B are functionally identical. Repeatability is not exactly the same as Stiny & Mitchell's second point, because for repeatability it is assumed that grammars of two different artifacts are comparable, as opposed to determining the degree to which one grammar serves two different artifacts. However, these two approaches end up being computationally similar in the worst case, because applying a grammar to an existing artifact is approximately as hard as developing a new grammar from that artifact. This work does not prove that claim, but since our definition of repeatability is stricter than Stiny & Mitchell's second point, that is the preferred metric in this work. Specifically, repeatability reports the percentage of paired rules when computing a rule set from an induced grammar output.

Computational complexity of grammars is a well-studied challenge (Slisenko, 1982; Yue and Krishnamurti, 2013), and determining if two different grammars can have identical output, with only a difference in complexity, is nontrivial. Prioritizing *conciseness* in generated grammars can be established by adhering to the information axiom found in Axiomatic Design (Suh, 2001); if two designs are otherwise equal, choose the simpler one. When learning grammars, and after ensuring they have

accuracy, variability and repeatability, the next priority is to establish that the selected grammar is the simplest. In practice, this is convenient for computational complexity but also because it implies that more salient information is stored per grammar rule, so arguably, it can demonstrate more nuance in alternative outputs. Conciseness in this work is reported as the percentage of rules in a grammar per unique node type in the data. Because some input data may have more types of nodes and others may have more types of rules connecting fewer nodes, it is not possible to set a specific goal for conciseness between datasets. However, it is feasible to minimize this among possible grammar solutions for a given dataset or group of similar datasets. This can be achieved by adjusting parameters such as the starting point of grammar induction or rule definition metrics, and evaluating the relative conciseness of each approach and selecting the most concise grammar representations. Additionally, across a large number of datasets this value tends toward 50% because on average a rule represents a relationship between two node types.

Together, accuracy, variability, repeatability and conciseness offer a multi-factor means for establishing computational tractability as well as effective communication of style. Achieving the first two is a necessary condition for a grammar to be considered effective. The latter two offer helpful insight when deciding which grammar representation best suits a given data set. These will be used as key metrics for determining the effectiveness of grammar induction methods introduced in this chapter.

Methods

The approach introduced in this chapter induces arbitrary grammars from ontological data via the following steps (Table 1): 1. deconstruct the data into objects that constitute atomic units of meaning in the context of the data (e.g., words in text, points in a drawing, people in a social network), 2. formulate structures from known relationships between objects in the original data (e.g., word pairs, lines, and friendships, respectively), 3. compute frequency based assertions about probable inter-rule relationships,

4. establish new rules from probable inter-rule relationships that were found. The resulting rule set will be concise, and complete, however it may also introduce a number of anomalous rules that can be removed by expanding the dataset and checking for coherence between elements in the ontology. Because this approach can use abstract ontological data, it can be used for both shape and graph grammars, and consequentially, the grammar representation in this work stores only ontological structure of the rules. Figure 2 shows the abstracted form of this algorithm in pseudocode, and in the following subsections these steps are discussed in detail with contextualizing examples.

Input: Data representing a graph of connected objects

Output: A set of rules that can be used to reconstruct the input graph

1. Create a list of objects in the graph
2. While the list is not empty:
    2.1. Remove the first object from the list and use it
    2.2. Find instances of this object in the graph
    2.3. Store collections of nodes within a distance parameter as candidate rules
    2.4. Check for collections that are structurally identical; store these as candidate inter rule relationships
3. Repeat 2 with the list of collections instead of objects
4. Remove duplicates

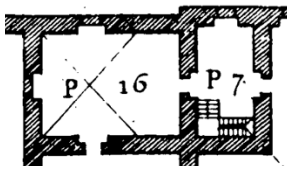
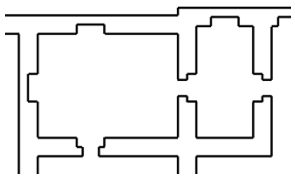
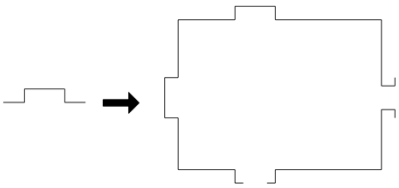
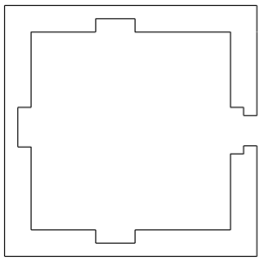
Figure 2 Pseudocode of the automated grammar induction algorithm.

Deconstructing Data for Grammar Induction

The first step is deconstruction of data into objects. This involves breaking a dataset down into the smallest units that are relevant to a grammar. This is done so that the dataset can be treated as a computationally addressable graph and so that patterns in that graph can be used to establish the grammar.

The process of data deconstruction is domain dependent, so approaches for deconstruction depend on the intrinsic hierarchy of the data representation, and the degree of connectedness within the data. For example, in text, words are easily extracted producing a linear graph as is shown in Step 1 in **Table 1**. In more complex data such as ontologies that represent buildings, data could be split into entities like a light switch or toilet, or any other element that is an isolated participant in the overall system graph. Piecewise models afford this kind of component level decomposition, so extra steps are not required for models that provide inherent hierarchy, such as the COLLADA file format (Barnes and Finch, 2008). If interpreting blueprints on the architectural feature level, points and lines can be treated as objects in this way, with positional information serving as the data structure, however detailed positional data is not needed when treating these structures as ontologies because position information can be stored alongside the intrinsic structure of the component graph. It is important to note that decomposition in more sophisticated, integrated models, such as an A-class injection moldings requires specialized methods that are not examined in this work.

Table 1 Grammar Induction Example. Demonstrating 4 steps and sample output based on two types of data, sample text, to show the simplest form of rule development and a building plan (a subset of the Villa Malcontenta (Rowe, 1977)), to show application in a shape grammar context. Raw input data is segmented in step 1. Step 2 involves forming structures in the ontology. In step 3 relationships between structure sets are used to define rules. In step 4 parameterized relationships are used to form more sophisticated rules. Finally, output examples are shown that express different features from the input but also incorporate similar stylistic aspects.

Data Type	**Text**	Building Plan
Raw Input	…This is a sample English sentence….	
Step 1. Extracting Objects	(This, is, a, sample, English, sentence, .)	
Step 2. Forming Structures	(This) → (This, is) (is) → (is, a)	
Step 3. Expanding Rules	(This) → (This, is, a)	
Step 4. Inter-Rule Relationships	(This) → (This, is, a, (insert object rule))	
Sample Output	This is a sample sentence.	

Formulating Initial Structures

In the second step, structures are formed based on only the structure of the graph and converted into initial rule candidates. Rules take the generic form of a token on the left hand side of the rule and a replacement token on the right hand side of the rule, which can replace the left hand side token in its given context. In text, this means a particular string can be replaced by an alternate string. In graphs, rules adhere to the same formalism, identifying a subgraph that can be replaced by an alternate subgraph and rejoin with the same edges in the graph. This way of dealing with rules allows for shape or graph grammar rules to be treated relatively similarly during induction. It also allows for rules to refer to large areas or chunks, and affords meta-rules, or rules that refer to other rules as elements of the graph.

For parsing text, the words before and after each word become strong candidates for rules and they can be processed without any awareness of the part of speech or meaning of individual words. Rules can later be expanded to encompass larger groups of words or parametric patterns as they are encountered by the parser. Step 2 in Table 1 demonstrates this for the initial formulation of structures.

In complex ontologies, establishing rules is less straightforward than in text because it requires finding matching subgraphs. To do this, ideally every traversal of the ontology would be evaluated, by enumerating possible paths through the graph, so all possible rule options are considered. This is not computationally practical, especially for heavily connected graphs, so constructing larger structures as chunks or sets of rules, by grouping or topological patterns, can be used to reduce the load (Ates and Zhang, 2007). A useful chunk is a larger region of the graph that is repeated, either identically, or thematically, in more than one place in a graph. Chunks are formed by finding matching regions in the ontology and exploring their local neighbors to find larger regions of similarity. In the example shown in

Step 2 of Table 1, the architectural chunks demonstrate a key characteristic of this approach, that frequently repeated elements will appear as strong chunk candidates (e.g., door and window frames), while repeated elements that have internal differences are less strong candidates (e.g., the section of wall next to one side of a window frame). Because of the recursive nature of this process, rules and chunks are ontologically similar so no cutoff is needed, and chunking can continue until no more repeated graph sections are found.

Probabilistic Enhancement

The frequency of a rule being found in the dataset is interpreted as a predictor of the likelihood of its use in the final grammar, so this is the main indicator that is used to determine how to extend initial rules of pairs of objects to encompass more objects as either one dimensional lists or graphs. For textual data this is a straightforward process of checking the data for repeated pairs or patterns through a direct string comparison on the rule level as is shown by Step 3 in Table 1. The frequency of repetition of a pattern may be used to determine its likelihood in the graph.

For graphs, if any two rules match, the surrounding areas in the graph are evaluated and local patterns are defined. The process of frequency analysis for establishing rule likelihood is done continuously so by the time a pass of the data has been completed, both simple and complex rules (chunks) have already been established. This is necessary because the data is sampled randomly from unexplored parts of the graph, to enhance computation speed and avoid getting stuck exploring repeated localities. For well-structured, intrinsically rule abiding datasets, it is generally the case that all rules will be found before a full traversal of the data is complete because chunks make up a majority of such data.

Rule Parameterization

Inter rule patterns are also established to extend the probabilistic rule set. This is functionally similar to evaluating a grammar on the rule level. The set of rules already in the grammar are treated as the input corpus for another grammar learning process. This builds parameterized rules that can refer to a contextual variable such as a rule or token that could be used to complete some part of the graph. The new abstract rule parameterization is then added back into the initial grammar as higher level rules. Step 4 of Table 1 shows this in a text example where an object rule can be used to augment a new parameterized rule that has been developed. In practice, for text, these types of rules may represent higher level features such as writing tropes or an author's particular writing style. In this way, this process can keep occurring providing more and more abstracted versions of rules governing a dataset, and these higher level rules can be optionally stored separately for use in other analysis approaches, such as trying to define meta-styles in writing.

The parameterization of rich ontologies is a time consuming process because it requires the comparison of many rules to generate higher level rules. However, it can be executed by repeatedly running the same induction approach while new rules and their parameterizations are established (Rawson and Stahovich, 2009). This also ensures that groups of rules don't overlap significantly, which helps facilitate conciseness of the resulting grammar. This is arguably the slowest stage of the grammar induction approach but doing it after structured rules have been established reduces computation time.

Efficiency in Induction

The first and second steps are fast and run in linear time with the number of rules. The third and fourth steps are nondeterministic and slower, depending on the ease of establishing equivalence between rules. In datasets where rules are unidimensional, equivalence can be done very quickly by using a hashing

function on both sides of every rule. Hashing functions allow mapping large input signals, such as many lines of text, to a unique and fixed length key (Knuth, 1998). When data has more interconnection, such as in a social network or a building ontology, the hash approach only encompasses a single representation of the graph, and can lead to significant inaccuracies. The hashing approach only works on very accurate datasets and does not show any partial equivalence for noisy matches. As a result, in cases such as dealing with typos or poorly generated drawings, preprocessing of data may be required to create accurate matches to ensure objects that are semantically the same are interpreted similarly by the induction algorithm.

Instead of hashing, when inducting graphs, isomorphism is established using the fastest current algorithm from Babai et al. (Babai, Kantor and Luks, 1983). Additionally, graph chunks can be hashed, affording speed increases as larger rules are constructed.

Choosing steps to improve the performance of a grammar induction process is heavily dependent on attributes of the representation and underlying structure of the data. For this reason, provisions for processing each dataset to be evaluated will be outlined specifically.

Evaluation

Evaluations were conducted on 3 distinct datatypes: generated patterned ontologies, a building with known grammar representations, and a structured piecewise model. Generated partnered ontologies allow analysis of the proposed methods in an abstract context and allow for ontologies exhibiting specific features to be evaluated. Many buildings have received attention from the academic community and have well established grammatical representations which can be used for comparison. Because of the large body of existing grammatical analysis and clear rule abiding layout, Andrea Palladio's Villa Foscari, La Malcontenta, a home near Venice, Italy built in 1550, was chosen for evaluation (Rowe,

1977; Stiny and Mitchell, 1978). The piecewise model used is a 3D model of an integrated, automated greenhouse in which each part is individually modeled and independently identified. Evaluating data of this kind allows multi scale graph representation of the structure and facilitates highly accurate chunking. Analysis was conducted using the accuracy, variability, repeatability and conciseness framework.

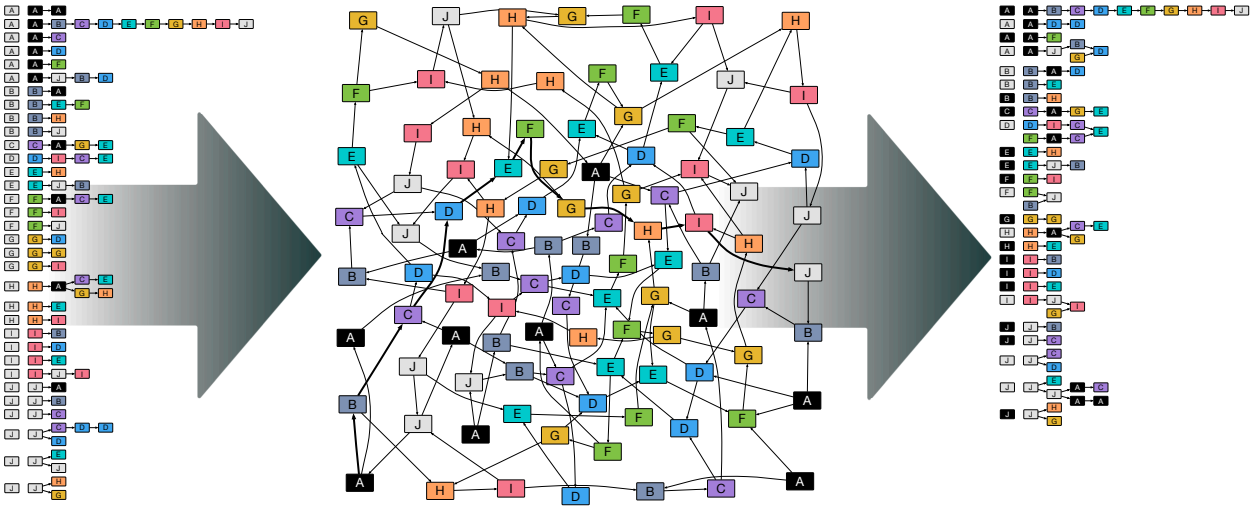


Figure 3 Example Small Generated Ontology. Rules on the left are used to generate the graph of 100 nodes and the induction method is used to establish the rules on the right. Note that in this example, the induction method produces a grammar with 25% fewer rules.

Inducing Generated Ontologies

Randomly generated grammar rules were used to build the artificial data sets, as opposed to purely random data, to ensure that there were sufficient patterns for the grammar induction algorithm to find.

A set of rules was constructed and then applied at random to produce datasets with specific size and connection properties. Graphs were generated with small (n=100, shown in Figure 3), medium (n=10,000) and large (n=1,000,000) numbers of nodes and average degree per node connection degrees

of 2, 4, and randomized degree, emulating text, visual design, and ontological design representations. These evaluation sizes were chosen to emulate problems of vastly different levels of complexity. The number of unique nodes in each case was 10% of the total number of nodes in that trial. Trial data objects consisted of strictly typed Universally Unique Identifiers (UUIDs) (Leach, Mealling and Salz, 2005) for high speed comparison.

The implementation was in Python 3 and interconnections were managed with the NetworkX library (Hagberg, Schult and Swart, 2008). All trials were run locally on commodity hardware with 8 cores and in instances in which processing lasted longer than one hour, trials were cut short.

When evaluating performance on synthesized ontologies, the nodes and connections are intrinsic to the representation of the ontology so the first step of the procedure is not explicitly performed and the second step can be conducted directly. Other steps for this evaluation were functionally similar to the text example given in Table 1, with the exception that random starting positions were used, and traversing the graph was done by selecting edges at random from the current operational node. As a consequence, the heavily connected nature of these graphs lead to larger graphs exceeding the allotted computational time and, in some cases, being removed from the evaluation.

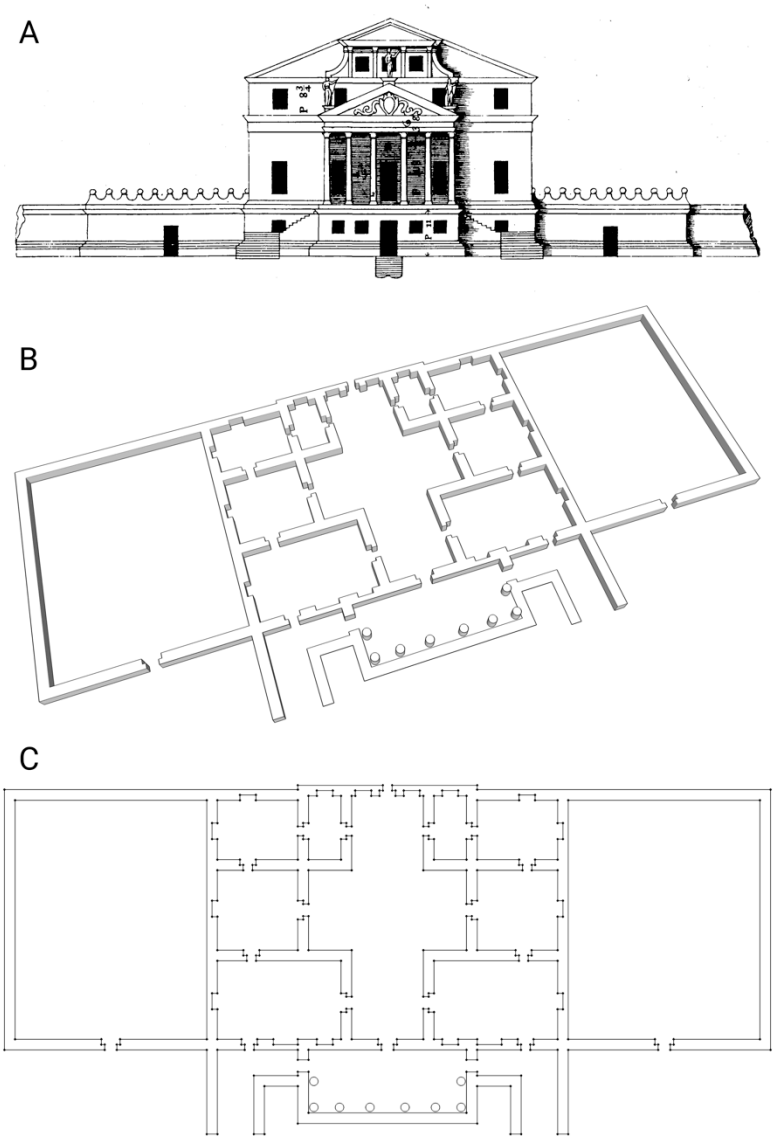


Figure 4 Villa Malcontenta. A. Illustration from plate XLIII in Quattro Libri (Rowe 1977), B. CAD rendition of features extracted from plan drawings of the villa. C. Visualization of the underlying graph with node position from coordinates in CAD information.

Inducing Palladio's Villa

A 3-dimensional model of the Villa Foscari, La Malcontenta was produced from reference drawings (Rowe, 1977) using a CAD package, and the load bearing walls and high-level plan were extracted, as

shown in Figure 4. This data was chosen, instead of, for example, the entire 3D model, because it more closely reflects the existing grammars that have been created for the dataset facilitating direct comparison to the previously conducted methods. The CAD data was converted into an ontological form in which the points making up each line segment were treated as nodes in the dataset and the lines were interpreted as their connections in the ontology. In other words, only places where lines end or join become critical. For example, a wall is interpreted as just the connection between the start and finishing points of it. Similarly, a door is just a relationship of points where the walls of two rooms meet. This way the grammar making process is not constrained by the physical structure of the building but by the relationships of each point to each other point. Spatial information was retained but not made available to the grammar induction algorithm so that the evaluation would be ontological while building shapes could be visualized in the way shown in Table 1. In this sense, the building data was converted into a graph of points found at junctions of edges from the blueprints, allowing step 1 of the induction method to be conducted in a very straightforward fashion. In steps 2 and 3, feature frequency was used as the primary indicator about rule likelihood, or chunk likelihood, leading to the most common feature patterns being used as fundamental rules. This lead to coherent rules as were shown in Table 1, and meant that expanding rules and establishing inter-rule relationships could happen without unusual wall segments or stray edges appearing out of place.

Because of the relatively small size of the grammar for the test data, the ontology could be completely parsed to find all rules existing in the data. Computations were performed on commodity computing hardware and all finished in less than 1 hour, so no measures were necessary to accommodate for computational complexity in this part of the study.

Assessment tasks were to attempt to formulate building primitives and to recreate the input data. Palladio's buildings have been the subject of a large amount of research on grammars so rule sets for comparison are readily available, such as from Stiny & Mitchell (Stiny and Mitchell, 1978), and were

used as plausible alternatives in the study of the variability of the produced grammars. Conducting this pairwise rule comparison offers insight into conciseness and variability of a grammar.

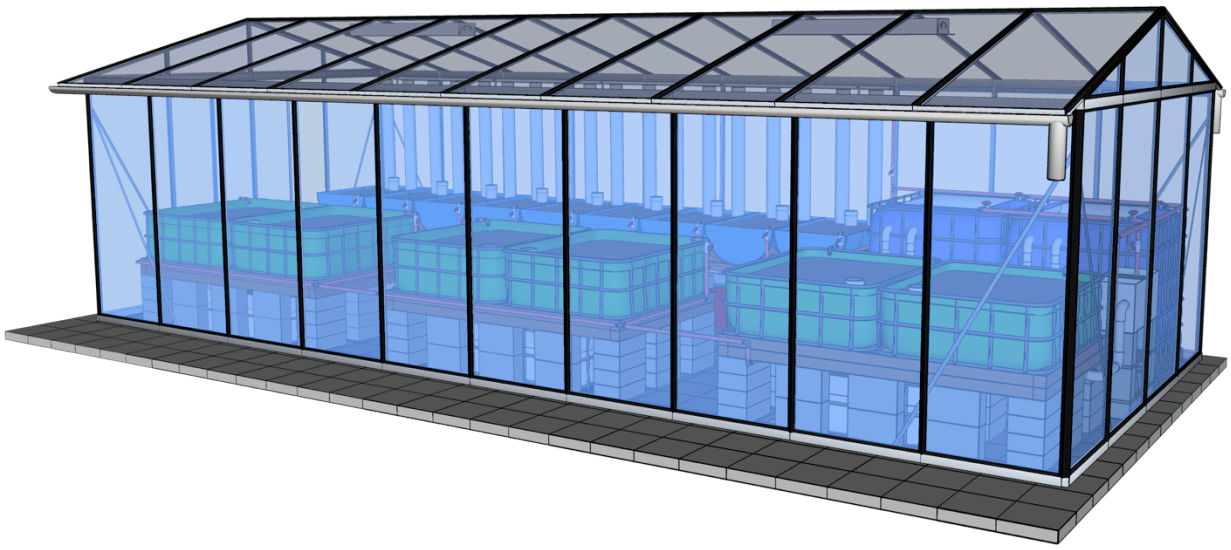


Figure 5 Greenhouse model. A CAD model of an aquaponics greenhouse with automated delivery systems (Snowgoose, (2014). Aquaponics system 8x3.4, 3D Warehouse.). Components down to the bricks and fasteners are individually modeled and are represented discretely in.

Inducing piecewise models

The greenhouse model (Figure 5) was produced using a CAD package and processed using the COLLADA 1.5.0 standard (Barnes and Finch 2008). The model contained approximately 700 components many of which were instances of other parts but configured in different relationships. All structural and machine components were included in the model but sub-assemblies were ignored when inducing the grammar because they were assumed to be readily purchased in that state, (e.g., a pump or a lightbulb would not be further decomposed).

Because of the nature of the COLLADA format, the model had an inherent structured representation so physical proximity was used to transcend structural isolation. As a result, Step 1 in the induction process involved traversing the COLLADA file and establishing a graph representation based on which components were in physical contact or orientation with one another. In this way, when processing rule chunking (i.e., Steps 2 and 3), only physical proximity was considered for building ontological relationships, hierarchical connection was not. This avoids giving weight to any assumptions made by the model's author about other hierarchical aspects of the design. Similarly, in performing Step 4 of the grammar process, establishing inter-rule relationships, this interpretation affords succinct analysis of the overarching relationships found in the model in a way that a hierarchical representation would hinder.

Results

Results are reported on trial runs of generated datasets for specific complexity levels in Table 2. Reported values are the percent of resulting grammars that express accuracy, variability and repeatability while reported values for conciseness are the percentage of the ratio of rules to total data-points, which agrees with computational bounds. All reported values are on 20 runs of the induction algorithm on the respective datasets, based on the metrics defined in Section 2 of this chapter. In data limited cases, such as the Palladian Villa and the Piecewise model, cross validation performed by inducing unique subsets of the data were conducted when needed, for each run. A necessary condition of any grammar to be selected was that it would achieve accuracy and variability; all evaluations reported achieved those goals.

Table 2 Grammar metrics. Accuracy and Variability are necessary conditions. Repeatability is desired at 100% and conciseness, reported as a percentage ratio of induced rules to unique nodes, is dependent on the type of data being induced. Lower bounds are reported, indicating the first quantile of runs. Calculations are based on the specific approaches introduced in Section 2 of this chapter. Evaluations

with runtime over one hour were omitted. In instances providing only one dataset to evaluate, such as the Palladian Villa and the Piecewise model, cross validation was used, inducing two unique subsets of the data for each comparator. For repeatability comparisons are conducted between induced grammars from either cross validation or sample diversity.

Trial	Accuracy **(% achieved)**	Variability **(% achieved)**	Repeatability **(% achieved)**	Conciseness **(% rules/n)**
Degree 2 Ontology	100% ± 0	100% ± 0	90% ± 3	52% ± 1
Degree 4 Ontology	100% ± 0	100% ± 0	80% ± 4	59% ± 4
Random Degree Ontology	100% ± 0	100% ± 0	78% ± 9	61% ± 4
Palladio's Villa	100% ± 0	100% ± 0	87% ± 2	56% ± 3
Piecewise model	100% ± 0	100% ± 0	74% ± 5	51% ± 1

Repeatability varied over the course of experimental cycles (Table 2). In the trial data cases, it varied proportionally to connectedness with higher dimensionality leading to significantly fewer repeatable grammars. This is likely because selecting for concise grammars may sometimes lead to certain grammar features being represented more than once. It can be very difficult to determine the difference

between two grammars for the same data set. For the random degree case this is not a problem for small examples, but for larger cases this might mean that it becomes almost impossible to rely on only searching for unique rules over all resulting grammars when the algorithm is run.

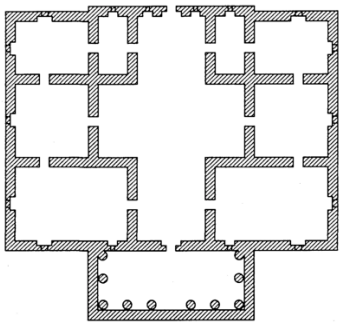
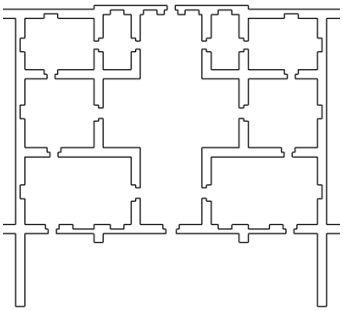
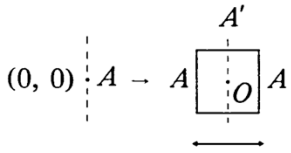
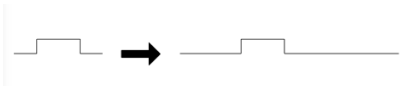
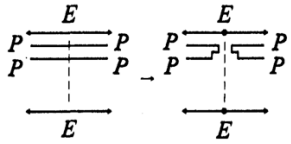
Conciseness performed best on one dimensional cases but showed significantly less favorable results in more complex evaluations. This is attributable to the number of plausible and actual node connected pairs increasing significantly with more dimensionality. As a result, however, it means that the inducted grammars are too large to be useful and are likely to include many rules that are near duplicates of each other. Removing these is nontrivial and would make the current approach orders of magnitude longer to compute. This said, efficient rule reduction occurred in many cases. This is demonstrated by the example shown earlier in Figure 2 of a random degree ontology evaluation in which the induced grammar is shorter and simpler than the input grammar, while retaining key high complexity rules, and adding multifunction rule alternatives.

The Palladian Villa grammar demonstrated similar results to the other evaluations. The induced grammar is able to almost perfectly express its source data and fail only where aspects of the source data were not fully induced. For example, the columns at the front of the house as well as the entry way stairs were not properly resolved because rules for those features had no ontological connection to the rest of the structure. This is not the case when the grammar is created using positional data but as previously mentioned, the intention was to build an ontological grammar from this data, because it is significantly more challenging to build grammars with no structural coding.

In this process key differences between our automated grammar induction technique and the traditional human grammar generation methods become apparent as shown in **Table 3**. For instance, the rule system introduced by Stiny and Mitchell (Stiny and Mitchell, 1978), is largely top down, moving from the largest layout and symmetry rules to smaller details later in the ruleset which are seemingly built to

augment the previously constructed forms. This approach is visible in others' assessments of the same buildings (Benrós, Hanna and Duarte, 2012), however it is not the only plausible approach as Stiny & Mitchell recognize. Using a bottom up approach, as is the case with our automated tool, derives high level constraints in the context of the details that relate to them. In this way, there are some differences in the strictness with which a design might be determined to be part of the same style. With the bottom up approach, grammar outputs always exhibit both plan level stylistic features, and detail design level stylistic features, for instance, large scale symmetry as relationships between doors and windows in the same room. On the other hand, the top down approach develops grammar rules that can be applied to generate buildings that are suitably symmetrical but have none of the low level features of the style.

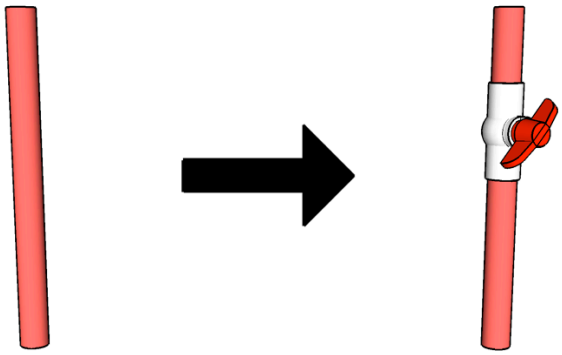
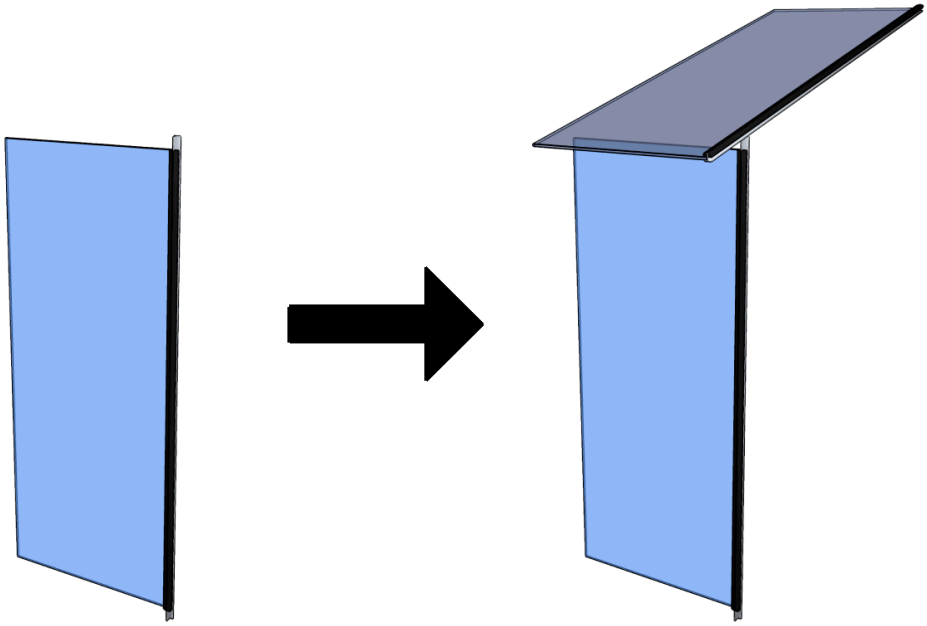
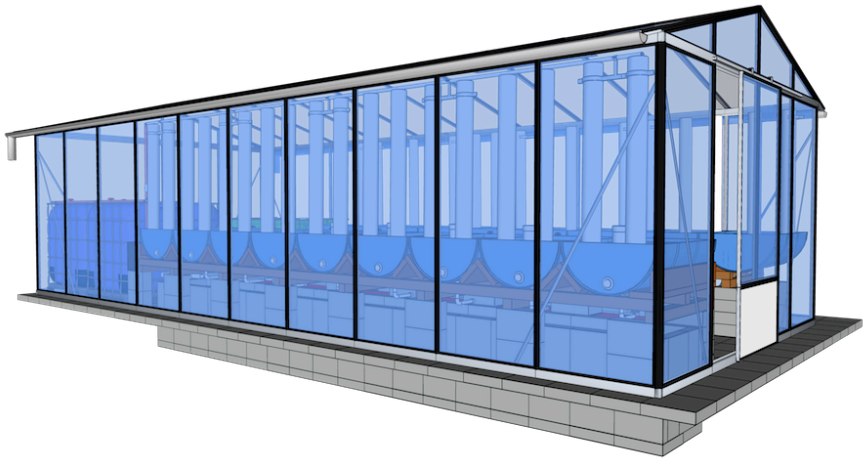
Table 3 Palladian Grammar Comparison. A comparison of grammar rules and results from foundational work on the Villa Malcontenta (Stiny and Mitchell, 1978) and an automatically induced grammar from the same source data. The outputs are largely identical but exhibit subtle differences relating to connected elements and the rendering of detail. For instance, more detail is retained in the automated output however that is due to how the manual approach was conducted, not because of fundamental differences in the approaches. The first rules are very different. Where the original approach started with grid definition, the automated approach chooses an arbitrary start point. Door rules show similarity; however, symmetry is not explicit in the automated approach. Rather, it is inferred by the ontology because components found to be part of the door making process cause the symmetry of the rule to emerge. Many rules are not directly comparable because of the difference in approaches but the results indicate that both can produce similar final results.

Data Type	**Original Palladian Villa** (Stiny and Mitchell, 1978)	Automatically Induced
Output		
First Rule		
Door Rules		

The piecewise model was also induced successfully, and this example provides a context for demonstrating how efficient a grammar can become when it is induced over a purely ontological model of a dataset, even if that data is 3D in nature. For example, rules extracted for this approach saw either very high repeated use or they implemented unique features. This shows that the induction process was able to build representative chunks that encompassed much of the inherent variability of details of the model. The rule set for such a complex model is rather large, encompassing over 800 rules governing 3900 components, so only a small number of example rules are shown in Table 4, with the addition of an example alternative output from the induced ruleset. This example offers a striking case for using the

shape agnostic, ontological approach adopted in this work. Parts in the induced model exhibited positional patterns but also functional patterns which emerged from broader symmetry and positional relationships. For example, positioning of pipes relative to valves and vats was performed accurately with the induced grammar in the example output in Table 4, even though these components had different physical relationships in the original induced model.

Table 4 **Piecewise Ontology Induction.** The highly general rule shows the combining of bricks as one of the rules in forming the base structure. In this model there are several ways bricks relate but by using chunked rules only rules about direct relationships are needed and these can be combined with higher level rules about the layout of the system to achieve the complete base. On the other hand, highly specific rules such as the one showing adding a valve to a segment of pipe, are used to position parts in their context to subassemblies, such as the valve, or such as a vat next to which this assembly would appear. A more traditional structural rule is also included showing a wall segment being used to position a roof segment. The example output shows a different configuration of vats with the associated changes in floor layout and pipe layout.

Highly General Rule	
Highly Specific Rule	
Traditional Structural Rule	
Example Output	

Conclusions

This work has introduced a method of efficiently inducing shape and graph grammars of complex design contexts using un-coded data. The method was validated with experiments establishing the accuracy, variability, repeatability and conciseness of produced grammars, with several examples including abstract data and design data. Speed and accuracy challenges are reduced but remain somewhat critical, so these will be an important part of our future work. Fast and accurate machine translation remains a focus of many researchers in computer science and linguistics; incorporating more lessons from their approaches, as well as considering other ways in which automated grammar approaches can be developed and applied are directions of deep interest to advance this work.

The difficulty of computing grammars used for design has made them relatively uncommon in industry. Facilitating their efficient use with this new automated approach may rekindle the formal assessment of shape in design outside of the academic community in computer aided design software, commercial design methods, and other fields that utilize special information. Furthermore, the academic advantages of having fast ways to generate grammars for a broad array of types of data opens the field to new grammar based endeavors.

Rule Frequency Analysis: Fast Classification for Design using Automated Grammars

Abstract

Detecting differences between designs is both challenging because of the wide range of ways design can be represented and used, and because of the need to differentiate brands. Grammar based methods have been used to represent designs and identify subtle differences but have seen limited application due to the difficulty of generating them manually. In other domains with straightforward representation schema, difference identification has been automated using machine learning, however, automating difference identification and classification using shape grammar techniques has not been previously demonstrated. This chapter introduces a multi-step pipeline for design classification, using automated shape grammar induction for case representation and the frequency of shared grammar rules as a set of features for difference detection. The method is validated through two experiments: a comparison to a common statistical learning based approach on simulated data, and with an automotive comparison task. The rule frequency approach introduced in this chapter is more sensitive and accurate than the statistical learning approach in a majority of cases and significantly faster, enabling learning with a minimal training set, due to leveraging the grammar based representation.

Introduction

Identifying subtle differences between designs can be important for clarifying brand differences and resolving intellectual property disputes, and in supporting designers as they develop novel products, however, there are few automated methods for this kind of distinction. Current approaches to

differentiating similar designs involve manual comparisons, which are time consuming, or high level comparisons, which tend to focus only on visual similarities, and fail to consider mechanistic or functional similarities. Shape grammars (Stiny, 1980) and graph grammars (Rozenberg, 1997) can be used to demonstrate systematic differences between designs (McCormack, Cagan and Vogel, 2004). For example, subtle brand differences have been assessed using shape grammar analysis (Agarwal and Cagan, 1998; McCormack, Cagan and Vogel, 2004). Graph grammars have also been used for difference detection, in contexts such as analyzing legal argumentation structure (Pinkwart *et al.*, 2006). Analysis of this type is traditionally done by hand or by using direct comparison of grammars, which is computationally expensive and has not yet been automated. As a consequence, this application of shape grammars remains under explored, so many potential applications in design and other domains are not yet available to practitioners.

Machine learning techniques provide a groundwork for classification and have been used in a wide range of domains that include structured representations and abstractions of data such as, abstractions of email text to classify mail as spam, and structured representations of photos to classify if an image includes a particular individual (Michie, Spiegelhalter and Taylor, 1994; Kotsiantis, 2007). A vast majority of classification techniques using machine learning require large amounts of data to establish probabilistic likelihood that a given classification is accurate. This finding means that machine learning approaches, while conceptually robust, are currently difficult to utilize in many of the situations where shape grammar classification would be relevant because of a lack of training data. For example, detecting brand differences in an industry with only a handful of products in the competitive market is not practical with many traditional machine learning methodologies due to the small data set.

This work introduces a classification approach by generating a vector space reflecting the frequency of grammar rules derived through an automated grammar induction process (illustrated on two node graphs in **Figure 6**). Starting with comparable structured data cases that can be characterized by graphs (Figure

1A), grammars are induced from each case, generating a series of grammar rules. The rules appear as transformations based on subgraphs from the data, and the frequency with which they occur in the data serves as a classification feature (Figure 1B). In this way, subtle differences captured by induced grammars provide a tangible differentiator, even when similarities are not visually obvious. Because this approach converts singular cases into complex sets of semantic rules, this technique can be used for near one-shot learning (Fe-Fei, Fergus and Perona, 2003), where traditional statistical learning approaches may require large amounts of exemplar data to learn a suitable representation of cases to make an accurate classification.

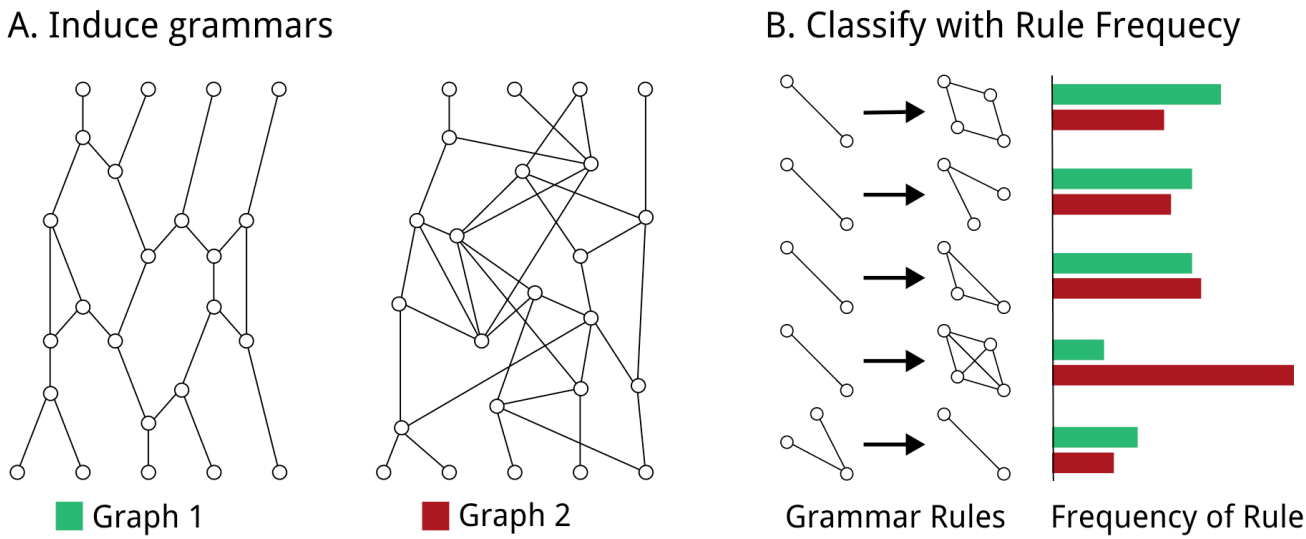


Figure 6 A pipeline for rule frequency based classification. A) The grammars of two graphs are induced into a set of common rules. Graph 1 depicts a less complex graph while graph 2 depicts a more complex graph with more connections per node on average. B) Checking how often each rule appears in each graph, a frequency representation is produced. In this example, increased average degree greatly influences the frequency of one particular rule, making classification of these graphs straightforward. The colors in the frequency chart indicate to which graph they relate.

The rule frequency method is validated in two experiments: evaluating its sensitivity to graph augmentations, and evaluating its ability to detect differences between automobile classes (Orsborn, Cagan and Boatwright, 2009). Sensitivity analysis is conducted comparatively to OddBall (Akoglu, McGlohon and Faloutsos, 2010), a common graph based statistical classification technique. The rule frequency based approach introduced in this chapter detects smaller differences in simulated graphs than OddBall given the same training sets, and is able to learn significantly faster. Additionally, the rule frequency technique was able to classify a majority of automotive categories with complete and partial representations of cars in as few as one comparison.

Related Work

Classifying designed structures as being genuine or not has been an important area of ongoing inquiry. Establishing the legitimacy of art or writing, establishing who built what (Jones, Craddock and Barker, 1990), and more recently, establishing if a designed product is brand name or counterfeit (Bamossy and Scammon, 1985) are all examples of this overarching need. Traditional techniques for detecting these differences include decomposing complex systems into their fundamental elements, and classifying instances by identifying features unique to them (Hodge and Austin, 2004). The frequency based approach introduced in this chapter mirrors these traditional techniques in an entirely automated pipeline, with a particular focus on systems and artifacts and datasets representable with graph and shape grammars.

General anomaly detection

Anomaly detection is conducted with a wide range of approaches depending on the particular needs of the application (Chandola, Banerjee and Kumar, 2009). Distance metrics and frequency based representations are two overarching categories of approaches.

Distance metrics use a heuristic to compute the distance in some space between cases. Initially theorized to establish the degree of difference between biological samples based on genetic feature maps (Hattemer, 1982), this technique has also seen use in measuring the distances between designs, both analogically (Christensen and Schunn, 2007), and based on structural features. These vector space techniques are also formalized under Hilbert spaces (Wootters, 1981). For these methods to work, a threshold distance is established that indicates a distance to be considered problematic. Sensing thresholds in high dimensions is also a field of rich discussion, however in this work only elementary methods are sufficient (e.g., nearest neighbors (KNN) (Cover and Hart, 1967)), so a more in-depth exploration has not been included.

Frequency based approaches rely on detecting differences in frequency distributions of particular features in a sample (Akoglu, Tong and Koutra, 2015). Methods utilizing this type of detection have been a center point in outlier detection in the data mining and machine learning communities (Hodge and Austin, 2004). In particular, techniques such as frequent subgraph mining (Jiang, Coenen and Zito, 2013), typified by gSpan (Yan and Han, 2002) and AutoPart (Chakrabarti, 2004), have been used with great success to find graph anomalies and outliers. OddBall (Akoglu, McGlohon and Faloutsos, 2010) is a technique for identifying anomalous nodes in a graph based on topology characteristics that has many similar properties to grammar based techniques, so it will be used as a basis for comparison in this study.

OddBall identifies anomalous nodes in a graph by using a mixture of graph properties and eigenvalues in complex graphs (Akoglu, McGlohon and Faloutsos, 2010). The approach focuses on the local subgraph (Akoglu, Tong and Koutra, 2015), relying on the local structure of a graph in a similar way that grammar rules may be defined. In this way, it shares many properties with a grammar based approach, and critically, it works unsupervised, being able to establish difference metrics without knowledge of states, so it serves as a reasonable method for comparison, with which to establish a

performance and functional baseline. Unlike the proposed grammar method, OddBall requires learning data in order to provide accurate detection.

Shape grammar as a basis for detection

Shape grammars have been used to provide classification for product designs in a predominantly manual pipeline (Gips, 1999). This generally involves first building an overarching grammar, then establishing if the grammar can be manipulated to represent a challenging case. Due to the manual nature of this process, human perception of rules is a potential source of inaccuracy, but additionally, the large amount of time it takes to conduct such a procedure makes comprehensive analysis impossible. As a consequence, statistical shape base analysis of designs (Orsborn, Boatwright and Cagan, 2008) have been leveraged as an aid in generating concepts, but this approach does not have the semantic coherence of a full grammar representation.

Grammar induction has been automated for a range of types of data in computational design and machine translation literature. A distinguishing factor of these approaches is how much information about the data being induced is required. For example (Sánchez-Martínez and Pérez-Ortiz, 2010) provides a method leveraging coded insight about words to produce lingual grammars. On the other hand, Sequitur (Nevill-manning, 1996) can interpret lingual data with no added information to formulate character level grammars. In design, coded visual design data, such as semantically coded parts of a website, have been used in automatically inducing Bayesian grammars (Talton *et al.*, 2012), while (Wu and Song, 2015) explored an approach to statistically deconstructing visual information without semantic coding. Statistical shape grammar techniques have also been applied in automotive design (Orsborn, Cagan and Boatwright, 2007). An automated, non-statistical, shape grammar induction technique for un-coded design and graph data has also been introduced (Whiting, Cagan and Leduc, 2017), allowing inducing grammars for almost any kind of structured data with a graph representation.

This final technique serves as a starting point for the rule frequency based analysis proposed in this chapter, and will be described herein.

Rule equitability

Frequency has served as a foundational indicator in information processing techniques (e.g., using a Fourier transform for spectroscopic classification of biological or chemical elements (Helm *et al.*, 1991)). However, to facilitate measures of frequency, equitability must be assessable over the elements for comparison. In other words, if rules can not be differentiated, and equated, then frequency for rules between cases (e.g. designs) being compared can not be derived.

Equating rules is nuanced because in many situations rules can be used to make other rules that may be within a grammar. To face this challenge, isomorphism techniques are required for identifying and reducing rules that are hard to otherwise compare. Markov equivalence classes (Andersson, Madigan and Perlman, 1997) provide a mechanism for formal identification of unique subgraphs by establishing an essential graph that embodies the core ontological relationship of a particular subgraph component. This approach, though not traditionally used in this way, is useful in identifying rule similarities because rules can be treated as subgraphs. Similarly sets of rules can be identified as a combination of subgraphs. When a rule and a set of rules have the same essential graph, they conduct the same ontological function as rules in the grammar.

Methods

Our work introduces an automated approach to classification leveraging shape grammar induction (Whiting, Cagan and Leduc, 2017), rule de-duplication using Markov equivalence classes (Andersson, Madigan and Perlman, 1997), multi-scale rule frequency checking, and case representation based on Hilbert spaces (Smola *et al.*, 2007), facilitating many classification techniques. Each step in this pipeline

(Figure 7) is described in detail in this section.

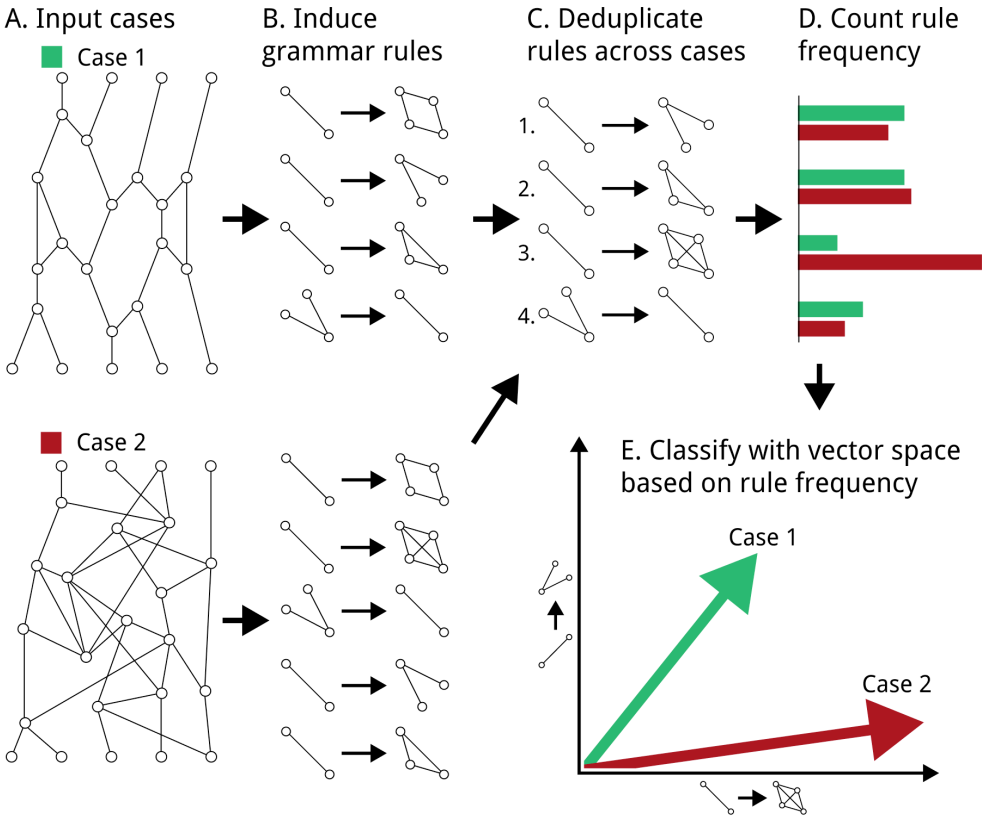


Figure 7 An automated pipeline for detection of grammar differences. A) Input 2 cases (case 1-green and case 2-red), B) Automatically induce grammars for each case, C) search for and remove duplicate rules; rules are given unique identifiers for future reference, D) process the frequency of each rule in the original cases and, E) treat the cases as vectors in a vector space of rules. For this example, only 2 rules are visualized, rule 3 on the x axis and rule 1 on the y axis.

Automated Grammar Induction

To use shape grammars for classification, the first step is to establish a reliable mechanism for acquiring consistent grammars from data. Depending on the source and features of the data, a range of approaches can be used. Essentially, any method is acceptable as long as it can produce induced grammar rules with consistency between the test cases being compared. In other words, a method is suitable, as long as there

is consistency in which properties of the data are interpreted as nodes and edges. For this reason, all data discussed in this work will explicitly describe the high-level segmentation approach of the data into a graph. It is important to note that many traditional uses of shape grammars are implicitly similar to graphs in the sense that line drawings are a form of graph and their rules are made up of subgraphs of that graph, so the distinction between shape grammars and graph grammars is blurred in the in this chapter.

In this work a general mechanism for inducing grammars from un-coded data is used as the underlying approach to establish grammars for processing (Whiting, Cagan and Leduc, 2017), shown in Figure 2B. It has been used because it offers flexible and generic grammar induction, not requiring precoding of induced data, and being agnostic to both data complexity and structure. The approach recursively examines tokens' relationships to find the most commonly repeated patterns in the graph and then defines rules based on those patterns. As more tokens are processed the number of rules iteratively grows, and the related elements of the graph are replaced with the new rules. Because this happens recursively, earlier rules are often referenced by later rules, and as a consequence, a network of rules emerges that can generalize the structure of the induced data.

The implementation in this chapter uses a random starting point and a random walk, moving through the graph choosing the next node to evaluate at random from the connected nodes, to explore the unevaluated parts of the graph. Additionally, forming groups of parsed tokens and rules based on proximity within the graph facilitates faster rule generation by providing a kind of rule chunking (Whiting, Cagan and Leduc, 2017). Together these techniques constitute the first stage of rule frequency based classification, establishing the set of rules across all cases, which will then be deduplicated between cases, and have their frequency assessed in the data.

Removing duplicate rules

After using a representative sample of test cases to induce a grammar with the previously described automated method, it is necessary to remove duplicate rules and to remove small groups of rules that are functionally equivalent. This is done so that a functionally similar part of two cases will be identified as similar when comparing with the grammar rules.

Many repeated rules are easy to identify; if the left hand side and right hand side in each case match, then it's reasonable to consider the rule identical. However small groups of rules that have similar collective function but are made up of unique rules are not so straightforward to identify.

Markov equivalence classes identify groups of elements with shared members through an adjacency matrix representation (Andersson, Madigan and Perlman, 1997). Groups are formed for chains of rules that share input and outputs. In this way, chains of rules found in one case, which compare identically to chains of rules found in another case (Figure 3A), may be treated as similar meta rules and removed (Figure 3B), even when the individual rules making up these chains don't compare exactly.

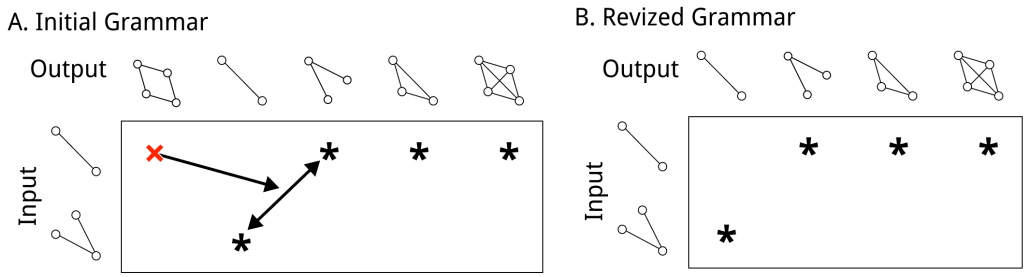


Figure 8 Removing chains or rules that have the same function. A) A matrix showing all rules for given input and output pairs. The rule marked with a red x can be removed because the rules it points to can be used in succession to provide the same function. B) The simplified rule set, which now has one fewer output tokens.

This process involves checking each possible subgroup of rules against its counterparts, essentially creating a higher level rule for each subgroup of rules. These higher level rules can be compared rapidly

in a pairwise fashion, but the process of checking each subgroup is computationally intensive. However, in practice, and in the examples conducted for this work, grammars are generally much smaller than 10,000 rules, and at this scale the delay is unsubstantial for a standard modern computer.

Ranking rules by frequency

Having established a set of shared rules, the instances of each rule are counted in each case to be compared. This is straightforward with simple rules in which both the left and right hand sides are particular elements or configurations. However, meta rules (Sondheimer and Weischedel, 1980), rules containing other rules and thereby encompassing high level, abstract relationships in the data, are only applicable when all the standard rules have already been applied. For this reason, all the standard rules are counted by applying them to the data. Then the meta rules can be counted by applying them to the combination of data and rules that remain.

Rule frequency for each case is used as the core representation for learning and classification. Because groups of comparable cases are likely to share a majority of rules, after the initial induction process, further induction is not necessary except when there is a situation in which a rule is missing. If, in the process of checking rule frequency on a particular case, there's a part of the data which no rule can be paired with, this is an indication that the ruleset being used does not completely define the relevant design space. In this situation, the new case should be induced and frequency counting should be repeated for any previously evaluated cases, to avoid biases due to the earlier, partial rule set. In practice this is an uncommon situation because needing to reevaluate rules tends to indicate that the differences between the cases of data are very significant, and may be obvious without utilizing a grammar based approach.

Classification approaches

Given the convenient rule frequency abstraction, many classification approaches are directly applicable. In this work, a simple vector space based mechanism is used as an exemplar, but others may be more suitable depending on the particular classification task. The vector space approach was deemed suitable in this chapter because it demonstrates a familiar representation which is compatible with many types of machine learning methods.

Treating each rule as a dimension in a vector representing a particular case, and the corresponding frequency of that rule in that case as its value, a Hilbert space (Smola *et al.*, 2007) of designs is derived, extending traditional vector space computation into high dimensions. Treating each rule as a dimension may mean that the space is thousands of dimensions, but the Hilbert space representation affords use of standard distance metrics such as Euclidean distance with many dimensions, providing a straightforward means to detect differences between induced cases.

Figure 4 demonstrates a simplified example with 2 dimensions and 2 cases. The *x* axis indicates the normalized frequency of rule 3, while the *y* axis indicates the normalized frequency of rule 2 in each case, based on the rules defined in Figure 2. In this way, the colored vectors show where each case would be positioned in this space, due to their differing compositions of rules. The distance between these positions in space can be interpreted as the difference between the cases in this representation.

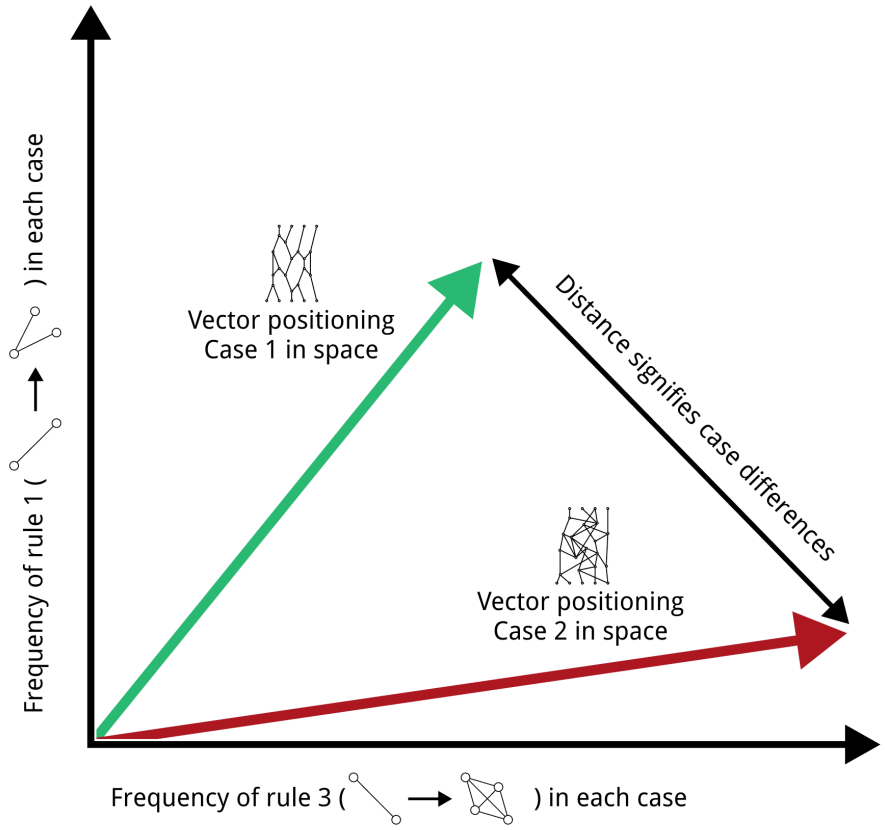


Figure 9 Treating cases as vectors in a high dimensional space. The depiction demonstrates 2 dimensions associating with 2 rules in a simplified example. The length of the difference between vectors indicates how differently each case utilizes the grammar. This can serve as a proxy for a more general difference metric, and is akin to the Euclidean distance within the space of cases. Only 2 dimensions are demonstrated for clarity, actual comparisons tend to integrate hundreds if not thousands of dimensions, one per rule in the deduplicated rule set.

The vector space representation also lends its self to many more sophisticated statistical comparison techniques. For example, nearest neighbors (KNN)(Cover and Hart, 1967) could be used to establish a nuanced classification boundary if there were many cases to train with. KNN establishes a classification boundary based on the a priori classification of the *K* nearest training cases to the test case. Nearness is defined contextually, but in this case the Euclidian distance would suffice. Other statistical learning and

classification techniques are also facilitated by the discreet vector representation of designs proposed, however in this work only the KNN approach is applied for classifying over the vector space representations because it is a method which is widely used in the broader community of people utilizing machine learning techniques.

Although many classification techniques require parameter tuning, for example, determining the smallest distance considered significant, aspects of this system require minimal intervention because there are no integrated parameters for adjusting the grammar induction and vector space representation approaches. Additionally, once a case domain has been established, further classification comes at a very low computational cost, requiring only deriving a rule histogram and then performing the preferred distance classification technique with the resulting case vector.

Evaluation

To evaluate the efficacy of the presented approach, two experiments were conducted: simulated test graphs with adjustable differences of a variety of types were classified against an existing classification method, and, systems of splines specifying particular automobiles were placed in automotive categories from the literature (Orsborn, Cagan and Boatwright, 2007).

Graph Classification

Classification accuracy was measured for comparison by determining the smallest measurable difference detected by the introduced algorithm and by the Oddball algorithm (Akoglu, McGlohon and Faloutsos, 2010), over a battery of standard graph properties.

Test graphs were generated to simulate graph abstractions of designs. Test graphs, like the ones depicted in Figure 7, were used instead of real designs' representative graphs because establishing small

measurable differences would afford improved gaging of accuracy and granular comparison with traditional methods. Previous work with automated grammar induction demonstrates that test graphs of this type have similar properties as induced grammars to their design counterparts (Whiting, Cagan and Leduc, 2017).

The Test graphs were simulated with a random graph function in the NetworkX python package with a target graph size of 1000 nodes and 50 unique node types. They were then measured for graph properties, and small modifications were made to achieve slightly different graph properties with a similar graph. The graph properties studied are all standard measure of graphs that can be readily evaluated: 1) *Circuit rank*, the smallest number of edges that can be removed to achieve an acyclic graph, 2) *Average diameter*, the average distance between vertices, 3) *Average girth*, the average minimal unique cycle size, 4) *Clique number* (Alba, 1973), the number of nodes in a clique of maximal size, 5) *Book thickness* (Bernhart and Kainen, 1979), the smallest number of independent planes with a common edge, on which a graph can be represented, and 6) *Boxisity* (Fishburn, 1983; Esperet, 2016), the number of dimensions required to represent a graph's edges through a Venn diagram style formalism. These are selected because they are fast to calculate for a given graph and change relatively independently of one another when modifications are made to graphs. Values of these properties were each normalized over a unit interval to allow for a straightforward comparison protocol.

Generated graphs were augmented by adding edges between a random pair of nodes in an existing graph. During this process, graph statistics were calculated at each change. In this way, new graphs with different levels in each graph statistic could be found and added into the dataset. The frequency of augmented graphs was logarithmically dependent on their distance from the original generated graph. In this way, more test cases were similar to the original graph and fewer were different.

In addition to using the automated grammar based approach for classification, OddBall (Akoglu, McGlohon and Faloutsos, 2010), was used to set a baseline for functionally similar techniques. OddBall was set up to use the same data for classification, but due to its inability to conduct one-shot learning, 10 trials were conducted as a singular training set, with cross validation performed against $1/10^{th}$ of the data.

Automotive Category Classification

Automotive categories utilized a dataset of spline measurements from 49 automobile bodies, encompassing 23 independent splines common among automobile designs (Orsborn *et al.*, 2006; Orsborn, Cagan and Boatwright, 2009). These data encompassed 3 distinct categories of automobile, coupe, sports utility vehicle (SUV) and pickup truck. This data enables grammatical analysis using the automated induction techniques by evaluating splines as elements in the structured graph to be induced. Additionally, by modulating spline parameters, a continuous space of automobile designs can be established allowing for a larger dataset to be evaluated with minimal cost to the data quality.

In this experiment, known automotive models and generated automotive models spanning the parameter ranges of the known models (Orsborn *et al.*, 2006) are classified into automotive categories. Generated models are used to standardize the number of cases to 100 in each category so that fair comparison can be conducted between more and less populated categories. Categorization is conducted using nearest neighbors in the grammar rule Hilbert space, and differences between designs are established with Euclidean distance. OddBall is not used here for comparison because of the nature of the representation it uses, relying on anomalous features on the graph level.

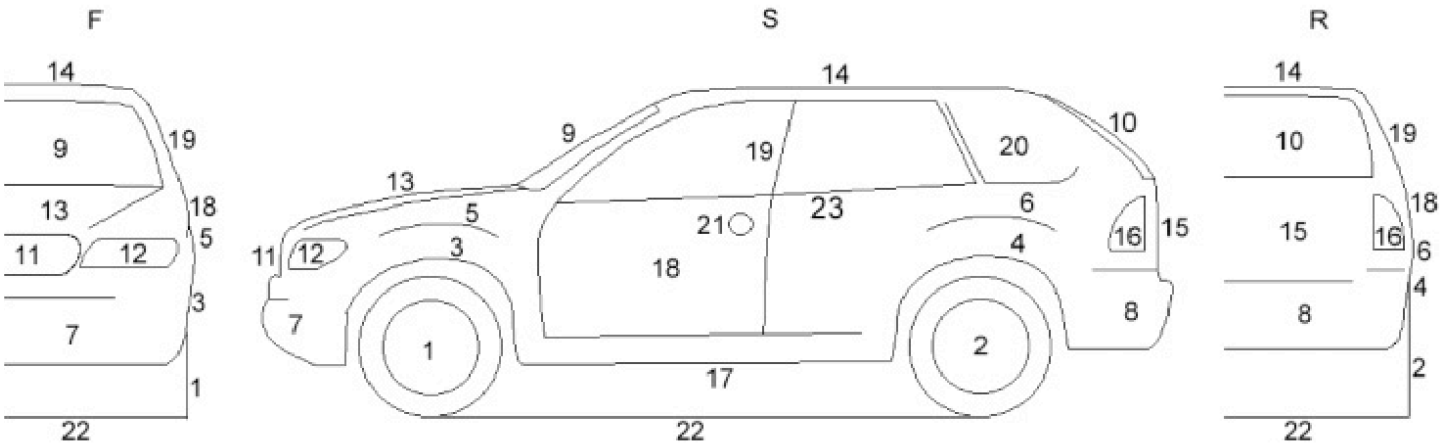


Figure 10 Vehicle splines for automotive category classification. The vehicle splines utilized: (1) front wheels, (2) rear wheels, (3) front wheel well, (4) rear wheel well, (5) front fender, (6) rear fender, (7) front bumper, (8) rear bumper, (9) front windshield, (10) rear windshield, (11) grill, (12) headlight, (13) hood, (14) roof, (15) trunk, (16) taillight, (17) rocker, (18) door, (19) front side window, (20) rear side window, (21) door handle, (22) ground, and (23) belt line (Orsborn *et al.*, 2006).

Results

This research aims to evaluate how well automated grammar based classification compares with existing methods, both generally and in applications to design. In this way, the results for this analysis are grouped by research question, instead of by experiment.

Is the grammar based approach accurate?

The Rule Frequency grammar based approach detected smaller property differences on average than OddBall in simulated graphs, with the notable outlier of circuit rank (Table 1). Overall, the grammar method had an accuracy 8.6% better than OddBall in aggregate, based on the average difference of detection accuracies of all the normalized properties across all rounds. The potential reason for the outlier is that the underlying statistic did not influence graph structure meaningfully, thereby being less detectable with the structure based grammar method than the mixed analysis of OddBall. However, the

aim of this comparison is to show parity with Oddball in this section. In the next segment of analysis, the rate at which each algorithm learnt is compared.

Table 5 Graph property detection with rule frequency. Within simulated graphs, smallest change detectible with significance ($p \leq 0.05$), normalized to a 0,1 interval, with significance established against cross validated cases over identical trials between evaluation methods.

Graph property	OddBall	Rule Frequency
Circuit rank	.25 ± .01	.28 ± .00
Girth	.23 ± .02	.05 ± .04
Clique number	.23 ± .08	.16 ± .05
Book thickness	.37 ± .09	.17 ± .01
Boxisity	.08 ± .06	.07 ± .06

In the automotive categorization problem, the grammar based method was used to successfully classify categories (Table 2) based on statistical category ranges identified in the literature (Orsborn *et al.*, 2006), with simulated and actual automotive data. The SUV category had most notable performance, which was attributed to its structural overlap with the other two categories, making it most likely to have comparable rule frequencies with all other automobiles. In other words, this result demonstrates how rule frequency functions well for differences in relatively similar data cases, a key tenant of the approach's design.

Furthermore the key splines (see Figure 5) were identified as indicative of particular category rule targets. The identified rule targets (Table 2) are those that were found as indicative of the appropriate category in the top 25% of rules. Functionally these align directly with features related to each category.

Table 6 Automotive categorization accuracy and key rule targets used for differentiation. Classification accuracy indicates the percentage of models in that category that were accurately classified out of 100 trials including both existing models and simulated models adhering to the established categories. Rule targets' semantic interpretation is based on rule descriptions from Orsborn (Orsborn *et al.*, 2006).

Category	Classification accuracy	Rule targets
Coup	76%	Tail light, headlight, grill, bumper orientation
SUV	93%	Truck closure, rear window
Pickup	82%	Truck bed, rear window

Is the grammar based approach fast?

As shown in Table 7, the amount of training needed to perform classification to a given percent accuracy is much smaller for the grammar based, rule frequency approach. In other words, the grammar based approach achieves similar accuracy with much less training data, making it useful for situations in which there is limited available data to train on.

Table 7 **Learning rate for generated graphs**, shown as the number of cases required in training, before a particular percentage of the achieved accuracy could be established. Trials are reported on cross validated data, identical between detection techniques.

Accuracy threshold	OddBall	Rule Frequency
25%	14	2
50%	23	2
90%	29	3

In the automotive problem, learning rate is not directly measured because there is no baseline. However, the classification conducted showed the same degree of consistency as the graph comparison. In other words, learning was effective after 3 instances.

Discussion

The introduced method has implications in utilizing rule frequency as a general abstraction of a shape or graph grammar and is also useful for further consideration of alternate learning and analysis methods. In addition to helping designers deal with shape grammars, this may serve as a useful tool for improving the productivity of designers by supporting them in visualizing functional and ontological differences in their concepts by leveraging rule frequency based classification on differences in representations such as graph grammars, functional decompositions or bond graphs. These types of abstractions provide valuable insight into designs, but can be difficult to differentiate when complex systems are represented. Furthermore, the introduced approach could aid in patent or tradedress infringement litigation situations

where designs' differences are subject to visual and functional scrutiny by formalizing demonstrative rules and facilitating the discussion of those rules as claims in intellectual property filings.

Similarity detection is a useful prototype application covered in this chapter, but in the long run, many other types of insight, for example, learning semantic tags of design features, and developing design strategy awareness, are both facilitated by the graph based representation approach introduced in this work. In particular, the prospect of achieving numerically robust representations (Mehta and David J. Schwab, 2014) could be used to usher in a new variety of deep learning in which representations are generalizable and precomputed to drastically improve learning rates in many cases. The longer term implication in design is that more fully automated tools for design decision making could emerge, because the current roadblocks due to the intrinsic difficulty in representing design knowledge could be mitigated, if more design information can be treated as comparable graphs and automatically processed in this way.

Conclusion

This work contributes a new approach to classification of differences in designs using automated grammar induction and a representation relying on the frequency of grammar rules in test cases. Experimentation showed it performed as accurately and significantly more sensitively than existing approaches for graph based classification and design category distinction. Additionally, this technique achieves learning rates significantly faster than a traditional statistical learning method providing a turnkey means for quickly classifying designs' differences that may have lasting impacts in the field of design, and may additionally have long-term implications for machine learning.

Acknowledgments

The authors thank Seth Orsborn for providing the automotive spline data found in (Orsborn *et al.*, 2006; Orsborn, Boatwright and Cagan, 2008). This work was funded by the Pennsylvania Infrastructure Technology Alliance.

Automated shape grammar as a tool for vascular analysis

Abstract

*S*hape grammars provide an alternative representation which can enrich structured data, facilitating learning with smaller datasets. A majority of machine learning methods learn representations apposite for further computational analysis; paradoxically, in contexts where data is limited, this leads to limited applicability of machine learning techniques. A pressing example of this is in medical scenarios where traditional diagnoses rely heavily on biopsy, so related imagery is scarce. For example, refined imagery of vessels is rarely clinically produced, unless needed for specific surgery, despite the fact that angiogenesis (Bergers and Benjamin, 2003) and tortuosity are powerful indicators of medical conditions (Arvanitakis *et al.*, 2017). This work introduces a method to leverage inherent structures, such as those found in blood vessel networks and nervous systems, to reduce the necessity for large quantities of data in producing assessments of medical conditions. The approach relies on a notion of shape grammars (Stiny, 1980), systems of structural rules, pioneered in the field of design and automated with techniques from machine translation and network analysis (Whiting, Cagan and Leduc, 2017). The method is validated with time-of-flight (TOF) MRI by generating grammars, and treating the set of grammar rules as a bag-of-features (Csurka *et al.*, 2004) over which statistical methods can be readily used to identify problematic cases in a small number of patients (n=17).

Introduction

Machine learning today requires a large amount of data, for example more than 1 million data points were used to outperform humans in dermatology to diagnose skin legions as cancerous or not, when

leveraging deep learning techniques (Esteva *et al.*, 2017). However, many kinds of medical data are in short supply. This is especially the case when indicative features are not identifiable by experts, such as radiologists. Vascular imagery is not traditionally used in many diagnosis contexts, and therefore the volume of suitable vascular imagery is relatively low. Yet, vascular tortuosity and angiogenesis have been associated with malignancy (Bullitt *et al.*, 2005), and metastasis (Bergers and Benjamin, 2003) in sarcomas and other types of tumors. Additionally, vascular anomalies in the brain are a telling indicator of a variety of health outcomes in patients (Arvanitakis *et al.*, 2017).

The reason these traditional approaches require so much data is that the representation of the problem space is determined through an approach based on incremental learning from trial and error (LeCun, Bengio and Hinton, 2015). Important, however, is how this appears unintuitive, compared with human approaches to developing similar representations: we break things down in to visual chunks and make meaning of them at multiple scales (Gobet *et al.*, 2001).

In design literature, shape grammar (Stiny, 1980) and graph grammar (Rozenberg, 1997) techniques afford representing complex structural relationships at multiple scales by breaking the information into rules with the form of left hand side (lhs) and right hand side (rhs) token pairs, and the tenant that any time the lhs is observed it could be replaced by the rhs (Figure 11). Grammar based methods provide a tool for generating output within a stylistic language (Trescak, Rodriguez and Esteva, 2009), or analyzing existing data through comparison of rules (Stiny and Mitchell, 1978). Such methods have seen a diversity of applications including analyzing architecture (Stiny and Mitchell, 1978), comprehending the underlying brand principles of modern vehicles (Figure 11A)(Pugliese and Cagan, 2002; McCormack, Cagan and Vogel, 2004), and, in an abstract form, aiding in the design of complex mechanical systems (Rozenberg, 1997). Traditionally this technique has only been automated by computer systems in particular contexts (Gips, 1999; Talton *et al.*, 2012), however, as discussed in Chapter 3, an automated technique has been introduced for establishing grammar rules from almost any

type of data that can be represented with a graph (Whiting, Cagan and Leduc, 2017). This method uses a recursive algorithm involving establishing rules based on similar components in a graph, replacing the components with new grammar rules, and then repeating the process, and establishing meta-rules (Sondheimer and Weischedel, 1980), rules composed of other rules, which are formulated with the same process. Over time the entire graph is induced providing a robust representation of the structural information encompassed in the original data.

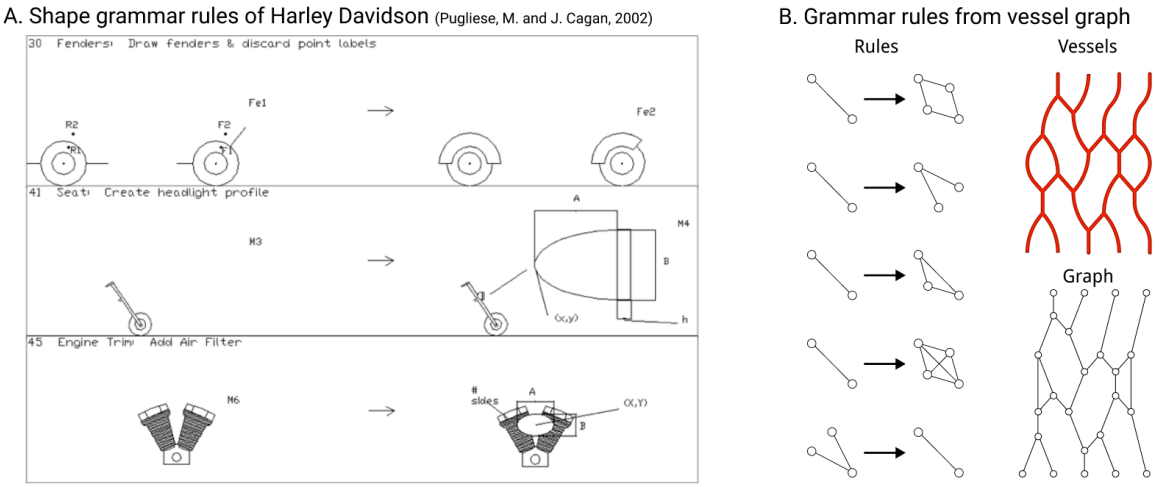


Figure 11 Shape grammar in design and biological graphs. A) Selected shape grammar rules for a motorcycle, which can be parameterized to articulate the Harley Davidson brand (Pugliese and Cagan, 2002). B) A collection of grammar rules used to generate vessel structures in the form of node graphs. In both A and B, the rules are depicted with a left hand side (lhs), which can be detected in context, and a right hand side (rhs) which replaces the lhs providing a specific augmentation. In other words, where one of the sub-graphs on the left hand side of a rule in B is found in a graph, it can be replaced by any of the corresponding right hand side sub-graphs. To afford variability in the output, some rules have the same left hand side, while having different right hand sides, such as the first 4 rules depicted here, which use the same sub-graph as the left hand side. In A, the augmentations influence overall design shapes,

whereas with B, the augmentations adjust graph connections, such as the example shown depicting an abstracted vessel graph.

Classification and analysis based on the frequency of features, such as support vector machines (Cortes and Vapnik, 1995) (SVM) and bag-of-features (Csurka *et al.*, 2004) models, provide a context for converting a distribution of features into a meaningful signal about underlying data. A key principle of these methods is that particular relationships between features may not need to be considered to uncover a reliable signal (Wallach, 2006). For example, bag-of-words models of text use only the presence and frequency of word use to classify its content, while ignoring word order. A strength of this approach is that it affords analysis without true comprehension of complex data. On the other hand, a weakness is that small datasets can be confounding because word order plays a significant role on the sentence level. Treating grammar representable data as bags-of-rules enables this kind of analysis to be conducted with contextually derived features, grammar rules, as was shown in Chapter 4. A grammar based representation also has the strength that it is not simply a bag-of-words, but a bag of induced rules, which encompass structural information, affording analysis with even relatively small datasets. In this sense, the intuition of a rule in a grammar tends to be more like a single word in a multi-page document, as opposed to a word in just one sentence.

Using automated grammar induction from Chapter 3 (Whiting, Cagan and Leduc, 2017) (Figure 12C), and grammar-based classification from Chapter 4 to determine differences, and key factors with medical outcome associations, this work introduces a technique for detecting important and specific differences between patients by leveraging a simple protocol for extracting spline networks of vasculature (Helmberger *et al.*, 2014) (Figure 12B) from time-of-flight MRI data (Figure 12A). The spline networks are converted to a graph which can then be used to build a grammar with the automated induction method. The resulting vessel grammar undergoes deduplication and simplifications before being used to evaluate how often each patient sample uses each rule. The rule counts serves as a feature for

differentiating indicative factors using a traditional statistical learning techniques such as those leveraging bag-of-features models (Csurka *et al.*, 2004).

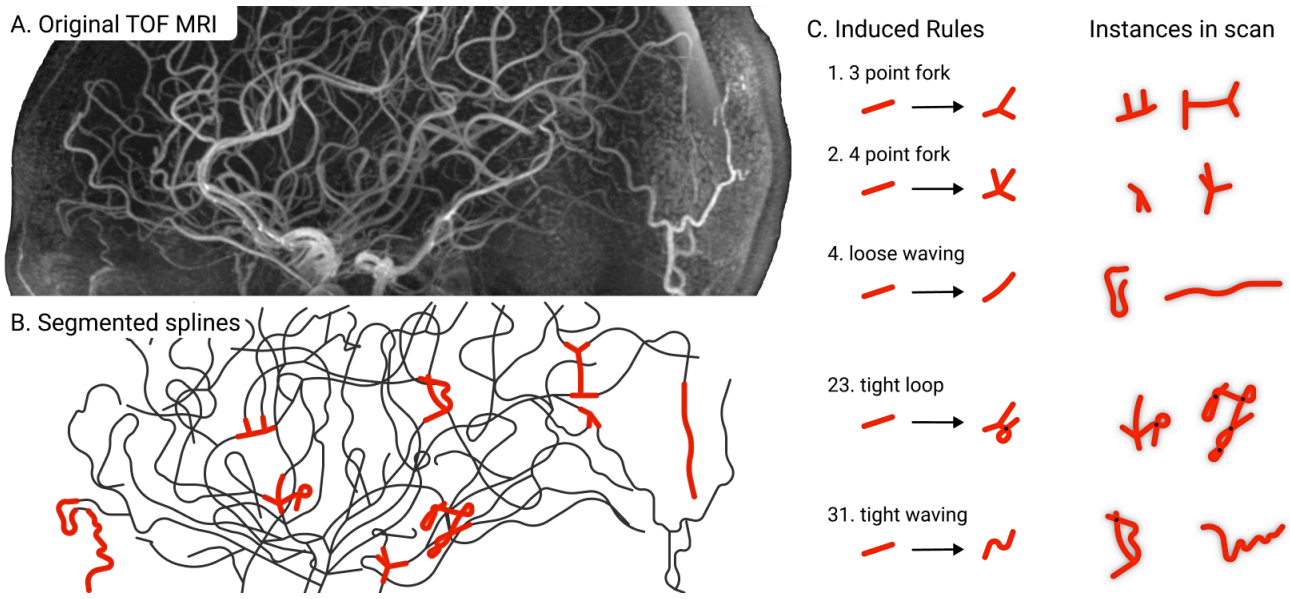


Figure 12 A depiction of the pipeline from MRI to grammar rules. A) A time-of-flight MRI shown from the mid-sagittal plane and rendered with a maximum intensity projection of the full brain. B) segmentation from the shown MRI aligned at the same orientation with parts of the vessel structure marked indicating distinct areas found in highlighted rules. C) A subset of the induced rules shown with their index and description as well as instances found in the scan. This pipeline indicates the major processing steps in preparing rules for classification and then finding them in the data from a given scan to establish the bag-of-rules representation. Note that through the deduplication and matching process rules often do not appear identical to the actual cases in the scan but are abstracted through the grammar induction process. Additionally, note that some rules are relatively similar to the human eye, but structurally significant for the grammar analysis. These are 2D projections of 3D vessel networks, so some rules include vessels that overlap in this visualization but do not intersect, these situations are marked with a small black dot in the rules. The full set of rules is available in Appendix 1.

This work leverages T3 TOF MRI scans from 17 sickle cell anemia patients. Known condition indicators for each patient were annotated, and a regression of patients and diagnoses was conducted based on the rule frequency representation of patient scans from the automated grammar classification process. Repeated grammar induction and deduplication resulted in a stable set of 47 rules with diverse counts across the data (Figure 13). By checking how well each rule, and groups of rules predicted each condition, singular rules and groups of rules that are significantly indicative of the conditions were identified. In testing this model through cross-validation, setting aside all but one sample of the data for testing, and training on the rest of the available data, conditions were identified commensurate to the significance of the model in the first round of analysis. In addition, the rules whose presence was significantly indicative of conditions tended to be similar in form to indicators radiologists' use when identifying these conditions. For instance, drastic reductions in venous radius indicated ischemia, a shortage of blood flow to a specific part of the body, causing infarction, the death of nearby tissue due to lack of blood flow. Rules indicating this reduction in venous radius were identified as prevalent indicators from the data as shown in Figure 17.

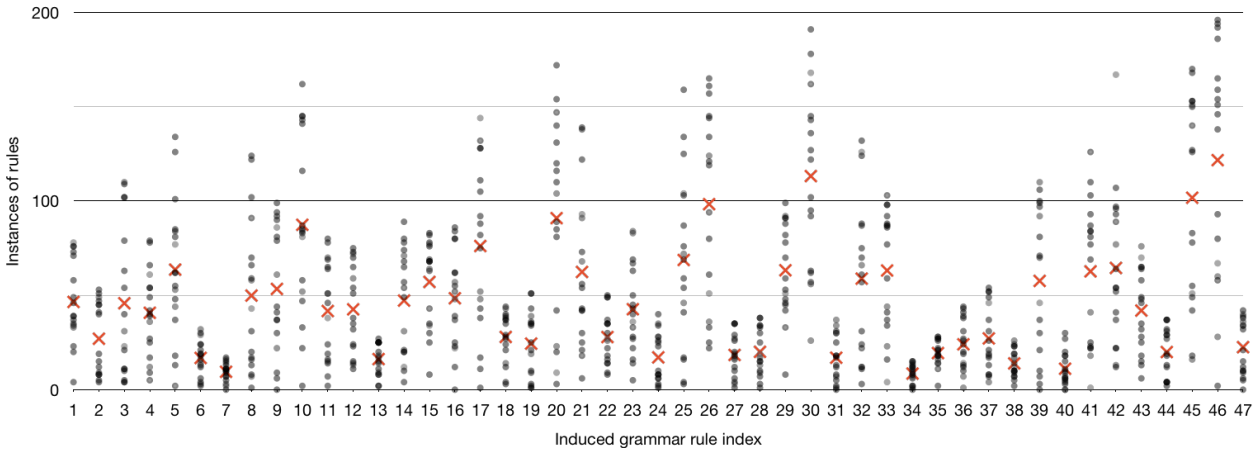


Figure 13 Grammar rule counts shown across all patients. The red markers indicate the mean of patients' rule frequencies for each rule. The black markers indicate that one of the 17 patients had that many instances of a particular rule in the evaluated data. Rule indices are assigned arbitrarily based on the

order the rules are finalized by the algorithm. This demonstrates that rule frequencies vary significantly across the body or rules.

Methods

Dataset

Our dataset consists of 17 sickle cell anemia patients' multi modal MRI scans of the brain taken as part of a larger study conducted under the guidance of Dr. Enrico Novelli and Dr. Joseph Mettenburg at the University of Pittsburgh Medical Center[1]. Patients ranged in age from 23 to 66 (average age 37.4) and in prognosis, from fully functional, to bedridden, and cognitively impaired. 7T time-of-flight imagery was used as it is a modality of MRI that provides coherent signals of arterial flow in the brain (as is visible in Figure 12A). The scans were taken with 320μm isotropic resolution over the full brain. An advantage of time-of-flight imagery is that at this resolution, the arterial flow is very clear and assumed to be physically accurate in terms of position and scale, due to minimal blooming in the time-of-flight modality.

6 radiologically identifiable conditions were incorporated in this analysis based on key vascular indicators which can be used by radiologists when identifying them.

1. General infarction and ischemia is correlated with areas of reduced flow or uneven flow, and can have impacts such as migraines. Ischemia is also age dependent in that a natural progression of brain aging is for some tissue to die irreparably.

[1] The data used in this study were used upon request from UPMC and are therefore not available to be shared publicly at the time of publishing.

2. Arteriolosclerosis, a variety of infarction caused by buildups of plaque, can be identified by signals such as uneven changes in venous radius.
3. Hypertension, a condition brought on by extended high blood pressure implicates tortuosity in arterial vasculature.
4. Aneurism is identified with areas of distended vasculature and has serious health impacts including sudden death.
5. Signals like abruptly terminating vessels in the brain are indicators of other problems, such as potential stroke.
6. Lastly, smooth reductions in venous radius can indicate local pressure, or flow problems from infection.

With these indicators and associated conditions 6 categories of vessel patterns are identified and are used in assessing patients in the analysis in this chapter. They will be referred to by number for the remainder of the chapter.

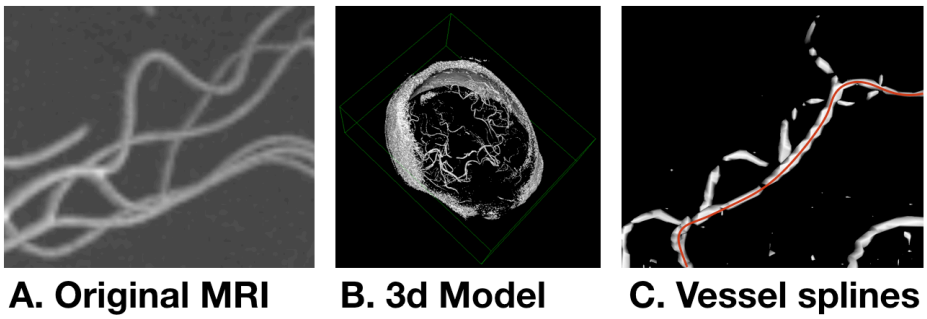


Figure 14 Data preparation process. A) A sampling of the original MRI image, B) a 3d extraction of the brain and skull, C) A spline running through a singular vessel after point averaging.

Data preparation

Vascular surfaces were extracted from the time-of-flight scans and exported to the STL file format using Horos's default threshold standard (Figure 14A). Extraneous data such as isolated venous artifacts and skull fragments were manually removed en masse in Blender (Figure 14B). The center splines of the vascular surfaces were identified by averaging 3D points' locations until a singular point remained at every cross-section using the Blender API (Figure 14C). This process was conducted semi manually to provide added flexibility when dealing with this small dataset; however, analogous approaches have been fully automated, achieving a similar resulting vessel graph (Zhang *et al.*, 2006; Helmberger *et al.*, 2014). The distance threshold used to average vessels to splines was recorded at each spline point as an accurate approximation for vessel width. A graph was formed, identifying position and diameter at every point along all the resulting splines. Splines that, at some point, pass closer than the width of a vessel were considered intersecting for the sake of the graph abstraction, so a node would be added at that point in the graph encapsulating an intersection.

Grammar induction and checking

Grammar rules were inferred from the vascular graph data of each patient by identifying repeated subgraph elements as rules and recursively expanding the bounds of the isolated subgraph to get the largest reasonable rule representation, leveraging the method introduced in Chapter 3 (Whiting, Cagan and Leduc, 2017). Meta-rules (Sondheimer and Weischedel, 1980) were inferred through the same recursive method after the initial data was entirely induced (Figure 12C demonstrates this pipeline and shows a subset of the induced rules). The set of rules was deduplicated between patients to achieve a minimal coherent set by using a Markov equivalence class (Andersson, Madigan and Perlman, 1997) approach that found groups of rules from different cases with overlapping functions using the methods

introduced in Chapter 4. The rules were counted in the data of each patient (Figure 13), producing a rule count representation amenable to use as a Hilbert space and for bag-of-feature style processing.

This process was conducted repeatedly, with the entire set of 17 patients, and with subgroups of the population to evaluate how grammar induction results were impacted by the number of patients induced. 47 rules were identified when the entire dataset was processed and deduplicated (the entire body of rules are visualized in Appendix 1). In alignment with the results in chapter 4, even inducing just 3 patient cases established a robust library of rules, enabling rule counting from other patients' data. Adding more patients' data to the induction process had a minimal impact on the quantity of resulting rules. As a consequence, and to leverage a standardized representation throughout the classification analysis, the set of 47 rules is used for the rest of this chapter.

Identifying condition indicators from rules

Condition indicators were treated as binary categorical features. Rules were treated as features of varying occurrences, depending on how many times the rule was found in a patient's data. By associating patients with conditions they're known to possess, rule count distributions for each condition were established. These were processed as deviations from the average number of times a particular rule was used, because rule count is not regularized (Figure 15 shows conditions deviance from the norm for each rule). By ranking the rules from most to least significant for each condition, rules with outstanding significance for unique conditions were identified (Figure 17). These rules are key indicators and for the conditions, in some cases, even just one rule was enough to identify a condition within the patients with better than 1:100 odds of error, based on the likelihood that a given rule is a statistically accurate representation of that condition indicator.

The rules better than 1:20 odds of error were combined as generalized linear models predicting each condition for evaluation leading to 6 models with that set of rules. Additionally, models encompassing

the set of the 6 key rules and models with individual key rules for individual conditions were also assessed. These three groups of regressions were used to construct 18 models in total, because one was trained to predict each of the 6 condition indicators (all 18 regression tables are included in Appendix 2).

The models were evaluated with leave-one-out cross validation, such that one patient's data was not considered in generating the model for evaluation, and that patient was used as a test and the error of the model prediction versus the patient's actual condition was recorded. After every patient had been evaluated in this way, the error was averaged. Error values for each condition, in each of the three model types are reported.

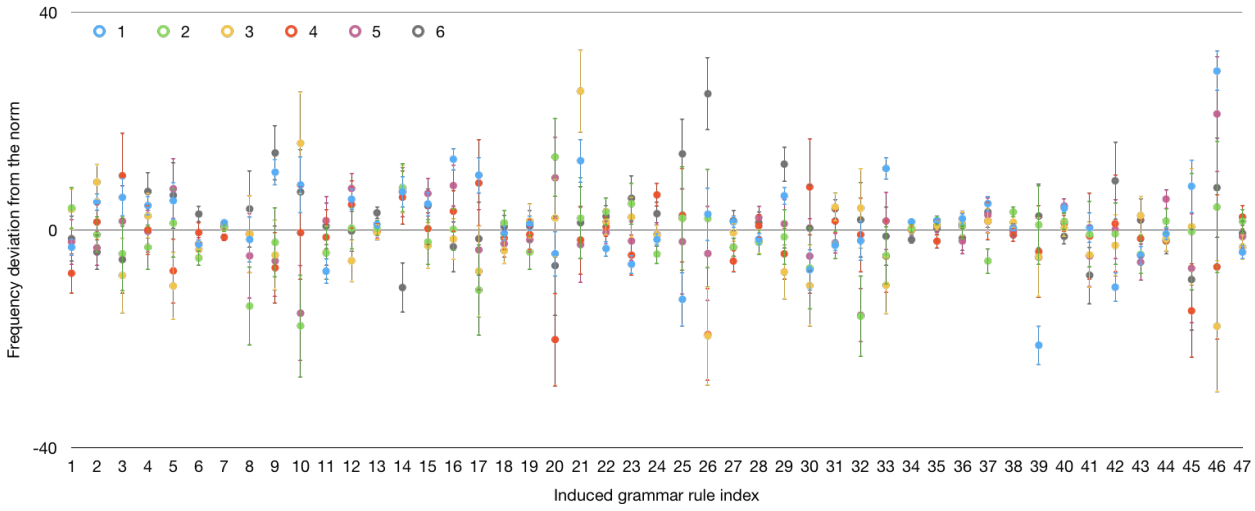


Figure 15 Conditions' deviance from rule count norms. Each condition (numbered) is rendered to show how it deviates from the average of patients for that rule. Bars indicate 95% confidence intervals of a particular condition for a particular rule. Some rules are clearly distinct indicators of particular conditions, for example rule 26 for condition indicator 6, and rule 39 for condition indicator 1.

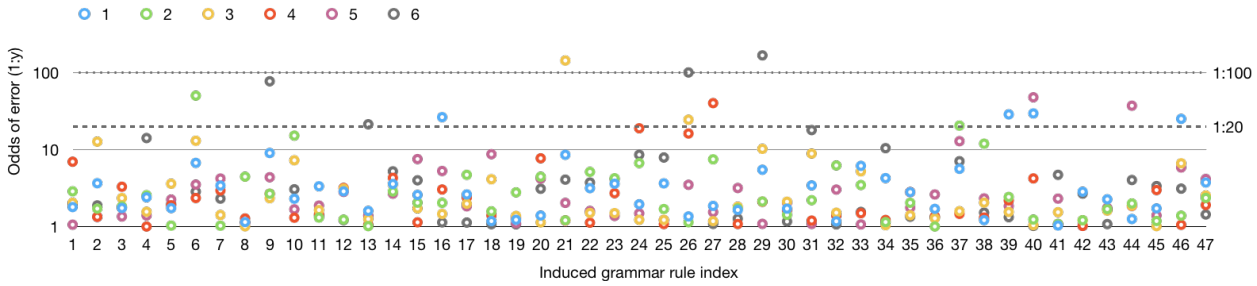


Figure 16 Indicative error odds associated with each rule for each condition, rendered on a logarithmic scale. The inverse p-values were used to indicate the error rate of an individual rule at detecting a particular feature. Dotted and dashed lines indicate the 1:100 and 1:20 error odds thresholds respectively. These are analogous to p-values ≤0.01and ≤0.05. Several rules indicate conditions with better than 1:100 odds of error, while many more indicate at 1:20 odds of error.

Results

Identifying indicative rules

Using a regression model, the significance of each rule in predicting a given condition indicator was computed (see Figure 5). With this approach, 6 key rules were identified which individually detect particular conditions with significance; rule 39 indicates condition 1 ($p \leq 0.035$), rule 6 indicates condition 2 ($p \leq 0.020$), rule 21 indicates condition 3 ($p \leq 0.007$), rule 27 indicates condition 4 ($p \leq 0.025$), rule 44 indicates condition 5 stroke ($p \leq 0.027$), and rule 29 indicates condition 6 ($p \leq 0.006$) (These rules are visualized in Figure 17). Each condition has at least one rule above a 1:20 error odds threshold, and several perform with error odds of less than 1:100, in accordance with some clinical standards of accuracy, for example, rules 29 and 21 in identifying condition 5 and condition 3 respectively.

This analysis provides the insight that individual rules can sometimes be very robust indicators of conditions. However, this analysis alone can't distinguish the causal nature of the relationship between a rule and condition indicator because this data only encompasses one moment in the life of the patient.

Evaluating training accuracy

The leave-one-out cross validation experiment agreed with the model evaluation above by averaging error from each round of the cross validation when detecting each condition indicator. However, overfitting is expected, as rules judged important to the set, failed to indicate a condition in the experiment as accurately as the simpler models (Table 8). This is judged to be the case because of the large number of rules (or features) relative to the amount of training data.

Some conditions, like condition 4, have much higher error in the larger models but relatively low error in the key rule alone case. This is related to the fact that condition 4 has only rule 27 as a significantly associated rule, as as shown in Figure 16. On the other hand, condition 5 retained relatively better predictive accuracy compared with other conditions, which is attributed to rules 40 and 44. This suggests that establishing indicative rules is a more effective diagnostic technique when broader signals are not available.

Table 8 Leave-one-out cross validation of evaluation models for each condition indicator. The cross-validation error values are reported (smaller is better). The *key rule alone* column shows the error when predicting conditions using only the key rule. The *key rule model* column shows the error when predicting conditions using a model that encompasses all the key rules. The *1:20 rule model* column shows the error when all of the rules with better than 1:20 odds (as shown in Figure 16) are used to generate the model.

Condition	*Key rule alone*	*Key rules model*	*1:20 rule model*
1	0.2050	0.2695	7.2847

2	0.1884	0.2892	7.6045
3	0.1647	0.2141	8.3606
4	0.2196	0.4108	25.0388
5	0.2280	0.2797	1.2642
6	0.1539	0.3427	5.8774

Key rules in-situ

Qualitatively comparing key rules to vessel patterns expected by radiologists for each condition indicator, there are some visually coherent rule associations, and some that do not align intuitively. Figure 17 shows each rule in-situ with the related diagnostic indicator. The key rules for condition indicators 1 and 3 show a pattern that can be visually interpreted similarly to the condition indicator. However, many of the other rules found to predict and indicator well don't actually appear visually similar to the signals radiologists use. This does not degrade the accuracy of prediction, but speaks to a challenge that radiologists face when assessing conditions: traditional indicators remain relatively nonspecific, and, due to a lack of structured computational analysis, it is hard for them to formally refine the indicators for which they look. On the other hand, radiologists look for much more than a few graph rules when diagnosing a patient, and the wealth of other data they incorporate plays an instrumental role in achieving high quality diagnoses.

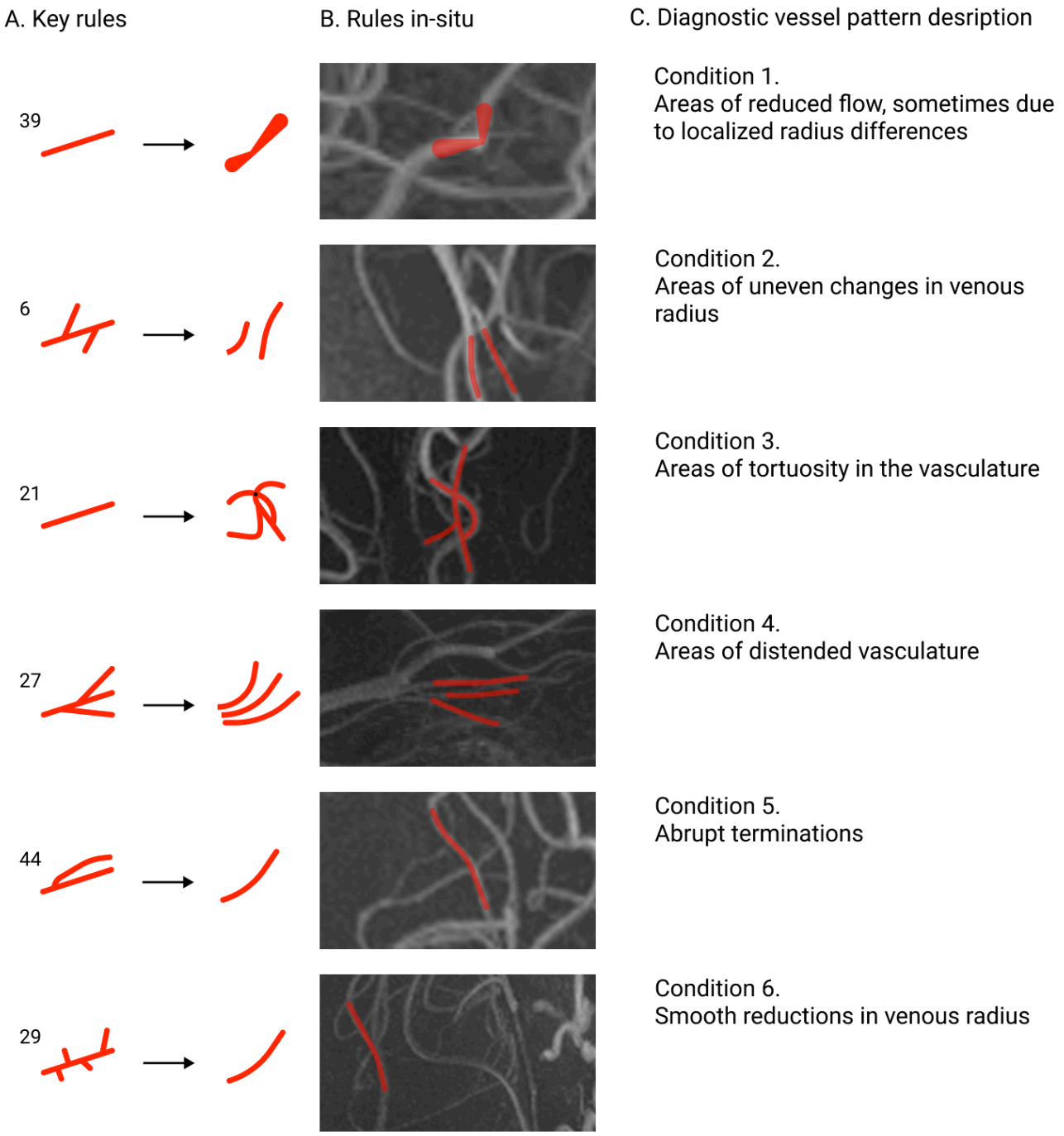


Figure 17 Rules in-situ vs rules in theory. A) key rules from the grammar with their rule index, B) key rules in-situ based on the detection conducted in the classification process, C) descriptions of each condition indicator based on the radiological assessment of a vessel pattern. Condition indicators 1 and 3 are visually recapitulated by the grammar rules, however the others do not offer an obvious

interpretation. As a side note, rules in-situ show the outcome of the rule being applied, and in some cases, like condition 6, it is not obvious that a rule should be applied at this time, without insight from the specific graph context of the rule application process, which is not readily visualized. Also note that the grammar rules are abstracted during the induction process to make them applicable between cases, so examples from the data are not visually identical. Lastly, these are 2D projections of 3D vessel networks, so some rules include vessels that overlap in this visualization but do not intersect, these situations are marked with a small black dot in the rules.

Discussion

Data limitations and opportunities

This work introduced a mechanism for identifying a variety of conditions with a small number of patients as training data. Though some condition indicators were evaluated with a clinical level of accuracy, others remain too low for clinical application. This is attributed to a small dataset in which overfitting occurred, however, demonstrating robust accuracy for some conditions asserts that with more data and statistical relevance for each condition, reliable signals are likely able to be found with a dataset in which there are a similar number or more patients compared with rules. For example, a complete conditional model, describing how the ensemble of rules can indicate conditions together could be evaluated if more test cases were available.

It is hard to draw conclusions about how conditions that were not explicitly identified in the data may or may not be revealed by the rule frequency technique. Assessment of larger medical diagnosis datasets and other nonmedical attributes, such as lifestyle data, are necessary for that analysis, and could lead to an exciting line of exploration.

The time-of-flight mode of imagery used in this study is effective for identifying large scale vessel structures. However, small vessels are not identified coherently by the method used for these scans due to a rate of flow threshold. The method introduced in this chapter is amenable to rich vessel graphs that include these smaller vessels, however traditional analysis techniques are not, so such data is not readily produced clinically. Incorporating this kind of data could provide a trove of previously unexplored signals.

Another challenge faced in this work is that brain morphology and local vascular patterns were not considered. Vessels grow differently in each part of the brain, however the root cause for this difference is not yet readily understood, so it is challenging to quantify these differences without a targeted study. Additionally, other aspects of brain morphology, such as the cortical thickness were outside of the scope of the vascular graph and were not incorporated.

Grammar method developments

The number of grammar rules, or number of detection features, has direct implication on the kinds of analysis that can be conducted. By finding ways to better pair down the rule set before analysis is conducted, for example by defining a more aggressive approach to targeting rule deduplication, or by leveraging a probabilistic similarity check when conducting initial grammar induction, an even more concise grammar may be found to be sufficiently expressive.

Alternative approaches to each step in the pipeline were not explicitly evaluated. For instance, automated techniques for segmenting and extracting a spline network, generating subgraph frequency based representations (Kuramochi and Karypis, 2001), and more advanced statistical anomaly detection methods could all have been used in comparison. In this way, this work stands as a proof of concept in introducing a new technique, which has many opportunities for growth and future research development.

Conclusion

This work demonstrates two important perspectives on the state of automated data analyses, particularly relevant to computational radiological assessment: 1) data requirements are drastically decreased by utilizing inherent structural information and learning over structural features, making the bag-of-features approach assessable on a diversity of alternative types of data, 2) specific alternative representations, such as design based grammar models, afford structured symbolic abstraction of graph structured data. The combined implication of these is that many traditional applications of machine learning in medicine may be revisited, and many new avenues of analysis may be unlocked by both considering structural information as a key indicator, and identifying derivable high-level representations which can be learnt as preprocesses for further classification or analysis using statistical methods. The net effect of these methods is that clinical analysis can leverage greater speed in processing and greater reliability in isolating indicators for particular conditions. In the context of this work, the introduced approach is also highly suitable for many other types of analysis in the medical domain such as, oncology, neurology, hematology and epidemiology, in addition to fields outside of medicine that implicitly or explicitly utilize graph structured data.

Discussion

This work has focused on presenting a way of inducing structural information that facilitates accurate training with fewer data samples than traditional, statistical learning techniques. In this case, automatic induction of shape and graph grammar has been used to form an intermediary representation that allows a small number of cases to be used with machine learning frameworks such as support vector machines, bag-of-features representations, and Hilbert spaces. However, a key implication of this work is that other similar alternative and intermediary representations may exist which could prove very useful for a wide range of applications. One example of this is medical analysis situations where governing models of a particular condition are known, but not being used when applying machine learning for data rich analysis.

In this way, an important underlying concept of this work was not to suggest that grammars are the only reasonable structured representation, but to suggest that grammars may be one of many that can be used in a wide range of contexts.

Limitations

This work has taken a very practical and applied approach to evaluating the proposed techniques. There is the possibility to produce more generalizable claims and proofs about these types of methods, in particular, proving that structural preprocessing could improve information in such a way that a learning algorithm would perform provably better given such a representation, than using a traditional learnt representation through deep learning. This kind of analysis has not been included in this work as it intends to serve as a proof of concept, opening the door for more explorations of this type in the future, by communities of researchers who have particular focus in determining computational bounds.

Another core limitation of the approach taken is that a majority of the cases considered are from either engineering or design, and thereby have not explored the full spectrum of potential application areas. The medical application presented in Chapter 5 is a notable outlier in this way, however many other contexts would be exciting to explore in the future. For example, how do grammar induction and classification techniques work in identifying specific members in social networks, such as fake accounts on Facebook. Another group of exciting directions is in the analysis of information and logistic networks such as the internet, local intranets, or even physical networks like plumbing systems. During the preparation of this dissertation, many of these directions were considered, however the design and medical applications were settled upon because the design examples are most salient to the experts in the associated methods, shape and graph grammar, and because the medical application offers an application with a high potential to improve people's lives. Additional medical applications were also explored in the preparation of this dissertation, in particular applying this method in assessing soft tissue tumor risk, however availability test data made it impossible to pursue that exciting direction.

Constrained by the number of application areas studied in this dissertation, only a small number of statistical learning mechanisms were tested with the grammar based classification method. Initial tests indicate that other techniques would work well, but they were not explored as part of this work, which serves as an opportunity for further research. In particular, integrating structured representations into deep neural networks seems to be an exciting future pursuit which could lead to useful improvements in the learning speed of such systems.

Larger implications

Learning more with less data is an oft cited challenge in trying to build artificially intelligent systems. Statistical machine learning can lead to high accuracy predictions, but the training data involved makes many types of prediction impossible. This work has tried to introduce an approach to avoiding this

problem, using a specific intermediary representation, grammar. However, other formalisms could be developed that led to even more robust efficiency gains. For instance, it may be possible to design formalisms specifically for the purpose of learning certain kinds of information quickly and robustly.

In recent years, the tendency for deep analysis of the underlying intuitions of machine learning techniques has led us to realize that much of what deep learning does differently is evoke a learnt representation. Perhaps, with techniques like the one this work introduces, the processes for learning such representations can be accelerated, and made more strategically focused on the particular goals of the learning at hand. Though we are relatively unaware of the underlying mechanisms of how humans think so dynamically, the prospect of having higher order awareness of structural sensibilities about material to be learnt is intuitive, and yet, this multi-scale metacognition is still largely unaddressed by computational approaches to the same problem.

An additional avenue of implications lies in the communities who already use grammars and who may be able to now leverage them more easily, and rapidly. For example, in design, shape grammars have been used to distinguish brand characteristics and help in related litigation cases. With an automated grammar based classification technique, this kind of analysis may be possible pre hoc, supporting designers in considering a wider spectrum of design solutions, and mitigate infringement situations. Although in this work, classification has been the main outcome of automated grammar induction, the prospect of pairing particular grammar rules with user intent, or designer intent, is also an exciting model which could lead to drastically new tools for automated design.

Conclusion

This dissertation introduces an approach for detecting differences in structured datasets using a small amount of data based on a combination of automated grammar induction, and a rule frequency based representation, making statistical methods effective, despite limited training data. The method was validated with design cases, with comparator cases from machine learning literature, and in a medical application, identifying brain vasculature structure indicating health conditions in sickle cell anemia patients.

This chapter will review the contributions and discuss avenues for future work.

Summary of contributions

This work has presented 4 main contributions:

Automated grammar induction

A generalizable automated grammar induction algorithm was introduced and found to be widely applicable, scalable, and efficient for moderately sized graphs. The literature is rife with solutions to parts of this problem, this work contributes a first complete and generalizable solution that can be applied in any context where a dataset can be represented as a graph of some kind.

Grammar induction evaluation heuristics

A set of 4 simple metrics that can be used to evaluate a grammar induction process were introduced. Accuracy, variability, repeatability and conciseness leverage the existing literature on representing style with grammar and creating concise representations, to contribute a quick way for grammar induction efforts to be evaluated and compared. Although some methods exist for particular characteristics of

grammar representation effectiveness, these heuristics offer a new, simple and generalizable measure which can be applied in many different settings without adjustment.

Grammar based classification

A new pipeline and combined approach for using grammars to represent data in a rule frequency based Hilbert space or similar representation was introduced. In particular, providing a way to convert grammar data into a form that is easily processed by existing statistical learning methods, so it can then be quickly computationally processed, is of critical convenience for the use of grammars. In this case, doing so in an end-to-end, automated pipeline facilitates a trove of further applications of grammar formalisms being used in design and non-design classification situations.

Application of grammar based classification methods in medical image analysis

Lastly, this work is the first to demonstrate using grammar based classification to predict health conditions based on rule distributions from grammars constructed with brain vasculature. The medical implication of this contribution is two-fold, 1) grammar methods may help in situations where a limited amount of diagnosis data is available, 2) existing radiological techniques can be extended through careful application of structured information.

Impacts

The most immediate impact of this work is to provide new ways for grammar based analysis to be applied. However, the long-term implications and impacts lie in the potential for these methods to be used in broader fields. Additionally, the key insight of this approach—using intermediary structuring representations to improve learning—can be applied with other structuring techniques and in other

domains. Together these factors mean that this work could have lasting impacts in a variety of research, business and clinical applications in which structured data is readily assessed.

An additional impact of this dissertation could be to provide new ways to augment radiologists' work, providing more affordable and more reliable service that can help let the radiologists focus on special cases that are not easily computationally addressed.

Future work

The most pressing directions for future work are in exploring alternative structuring techniques, and establishing robust bounds for the general approach of utilizing intermediary representations to provide improved performance in statistical learning settings. Additionally, working to improve the efficiency and coverage of grammar induction methods, working to integrate a wider variety of statistical techniques for distinguishing grammars, and exploring the range of representation output styles, are all directions of research which would lead to long-term improvements in applying grammars in classification and machine learning problems.

From a research perspective, each area this dissertation has focused provides opportunities for exciting future assessment. The grammar induction algorithm introduced in chapter 3 is robust and generalizable, but requires a graph based representation to start with. An important area of inquiry is in using this kind of method with a graph abstraction pre-process, which in many cases, may leverage transfer learning (Pan and Yang, 2010) to facilitate extracting graphs from unlabeled visual data, to be used with grammar methods. Also from chapter 3, the grammar induction heuristics have potential to become a useful assessment aid in the application of grammars in design, and through human-in-the-loop analysis, these heuristics could be validated stylistically, as well as computationally.

The methods described in chapters 4 and 5 also offer a wide range of opportunities for future research. This work served as the tip of the iceberg in applying grammar in assessment of medical conditions and in more general classification settings. Exploring how other machine learning techniques can be supported with grammars is a next step. Additionally, exploring how these techniques work with partially structured and unstructured data types may offer interesting meta insights into how some deep learning techniques operate, especially those in which fundamental representations are learnt.

A future for grammar

This work has presented a range of opportunities for using grammars in data analysis. Automated induction makes this representation accessible, and adding classification techniques provides a context for using grammars with other existing computational tools and processes. Additionally, providing example applications opens the door for other applications and directions of inquiry. In addition to provoking consideration of alternative ways to structure machine learning processes, this work aims to make it possible for grammar techniques to be revisited in many domains where they have seen restricted use due to traditional difficulty.

References

Agarwal, M. and Cagan, J. (1998) 'A blend of different tastes: The language of coffeemakers', *Environment and Planning B: Planning and Design*, 25(2), pp. 205–226. doi: 10.1068/b250205.

Akoglu, L., McGlohon, M. and Faloutsos, C. (2010) 'OddBall: Spotting anomalies in weighted graphs', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 410–421. doi: 10.1007/978-3-642-13672-6_40.

Akoglu, L., Tong, H. and Koutra, D. (2015) *Graph based anomaly detection and description: A survey*, *Data Mining and Knowledge Discovery*. doi: 10.1007/s10618-014-0365-y.

Alba, R. D. (1973) 'A graph-theoretic definition of a sociometric clique', *The Journal of Mathematical Sociology*, 3(1), pp. 113–126. doi: 10.1080/0022250X.1973.9989826.

Alexander, C., Ishikawa, S. and Silverstein, M. (1977) *A Pattern Language*, *Ch. Alexander*. doi: 10.2307/1574526.

Andersson, S. A., Madigan, D. and Perlman, M. D. (1997) 'A characterization of Markov equivalence classes for acyclic diagraphs', *Annals of Statistics*, 25(2), pp. 505–541. doi: 10.1214/aos/1031833662.

Arvanitakis, Z., Capuano, A. W., Leurgans, S. E., Buchman, A. S., Bennett, D. A. and Schneider, J. A. (2017) 'The Relationship of Cerebral Vessel Pathology to Brain Microinfarcts', *Brain Pathology*, 27(1), pp. 77–85. doi: 10.1111/bpa.12365.

Ates, K. and Zhang, K. (2007) 'Constructing VEGGIE: Machine learning for context-sensitive graph grammars', in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, pp. 456–463. doi: 10.1109/ICTAI.2007.59.

Babai, L. (2015) 'Graph Isomorphism in Quasipolynomial Time', *arxiv*, 7443327, p. 84. doi: 10.1016/j.disc.2006.09.012.

Babai, L., Kantor, W. M. and Luks, E. M. (1983) 'Computational complexity and the classification of finite simple groups', in *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pp. 162–171. doi: 10.1109/SFCS.1983.10.

Balahur, A. and Turchi, M. (2014) 'Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis', *Computer Speech & Language*, 28(1), pp. 56–75. doi: 10.1016/j.csl.2013.03.004.

Bamossy, G. and Scammon, D. L. (1985) 'Product Counterfeiting - Consumers and Manufacturers Beware', *Advances in Consumer Research*, 12(Eds), pp. 334–339.

Barnes, M. and Finch, E. L. (2008) 'Collada-digital asset schema release 1.5.0', *Specification, Khronos Group*. doi: 10.4161/auto.7.11.17661.

Beck, K. and Cunningham, W. (1987) 'Using Pattern Languages for Object-Oriented Programs',

Proceedings of the OOPSLA workshop on the Specification and Design for ObjectOriented Programming, (No. CR-87-43), pp. 9–16.

Benrós, D., Hanna, S. and Duarte, J. P. (2012) 'A Generic Shape Grammar for the Palladian Villa, Malagueira House, and Prairie House', *Design Computing and Cognition '12*, '12(18), pp. 321–340. doi: 10.1007/978-94-017-9112-0_18.

Bergers, G. and Benjamin, L. E. (2003) 'Tumorigenesis and the angiogenic switch.', *Nature reviews. Cancer*, 3(6), pp. 401–410. doi: 10.1038/nrc1093.

Bernhart, F. and Kainen, P. C. (1979) 'The book thickness of a graph', *Journal of Combinatorial Theory, Series B*, 27(3), pp. 320–331. doi: 10.1016/0095-8956(79)90021-2.

Berwick, R. C. and Pilato, S. (1987) 'Learning Syntax by Automata Induction', *Machine Learning*, 2(1), pp. 9–38. doi: 10.1023/A:1022860810097.

Bullitt, E., Zeng, D., Gerig, G., Aylward, S., Joshi, S., Smith, J. K., Lin, W. and Ewend, M. G. (2005) 'Vessel tortuosity and brain tumor malignancy: A blinded study', in *Academic Radiology*, pp. 1232–1240. doi: 10.1016/j.acra.2005.05.027.

Chakrabarti, D. (2004) 'AutoPart: Parameter-Free Graph Partitioning and Outlier Detection', in Boulicaut, J.-F., Esposito, F., Giannotti, F., and Pedreschi, D. (eds) *Knowledge Discovery in Databases: PKDD 2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 112–124. doi: 10.1007/978-3-540-30116-5_13.

Chandola, V., Banerjee, A. and Kumar, V. (2009) 'Anomaly detection: A survey', *ACM Computing Surveys (CSUR)*, 41(September), pp. 1–58. doi: 10.1145/1541880.1541882.

Chau, H. and Chen, X. (2004) 'Evaluation of a 3D shape grammar implementation', *Design Computing and ...*, (1972), pp. 357–376. doi: 10.1007/978-1-4020-2393-4_19.

Christensen, B. T. and Schunn, C. D. (2007) 'The relationship of analogical distance to analogical function and preinventive structure: the case of engineering design.', *Memory & cognition*, 35(1), pp. 29–38. doi: 10.3758/BF03195939.

Clarke, L. P., Velthuizen, R. P., Camacho, M. A., Heine, J. J., Vaidyanathan, M., Hall, L. O., Thatcher, R. W. and Silbiger, M. L. (1995) 'MRI segmentation: Methods and applications', *Magnetic Resonance Imaging*, 13(3), pp. 343–368. doi: 10.1016/0730-725X(94)00124-L.

Cook, D. J. and Holder, L. B. (1994) 'Substructure Discovery Using Minimum Description Length and Background Knowledge', *Journal of Artiicial Intelligence Research Submitted*, 1, pp. 231–255. doi: 10.1613/jair.43.

Cortes, C. and Vapnik, V. (1995) 'Support vector machine', *Machine learning*, pp. 1303–1308. doi: 10.1007/978-0-387-73003-5_299.

Cover, T. and Hart, P. (1967) 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory*, 13(1), pp. 21–27. doi: 10.1109/TIT.1967.1053964.

Csurka, G., Dance, C., Fan, L., Willamowski, J. and Cedric Bray (2004) 'Visual categorization with bag

of keypoints', *International Workshop on Statistical Learning in Computer Vision*, pp. 1–22. doi: 10.1234/12345678.

DeNero, J. and Uszkoreit, J. (2011) 'Inducing sentence structure from parallel corpora for reordering', *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 193–203.

Ding, Y. and Palmer, M. (2005) 'Machine Translation Using Probabilistic Synchronous Dependency Insertion Grammars', *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 38(June), pp. 541–548. doi: 10.3115/1219840.1219907.

Eberle, W. and Holder, L. (2007) 'Anomaly detection in data represented as graphs', *Intelligent Data Analysis*, 11, pp. 663–689.

Esperet, L. (2016) 'Boxicity and topological invariants', *European Journal of Combinatorics*, 51, pp. 495–499. doi: 10.1016/j.ejc.2015.07.020.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. and Thrun, S. (2017) 'Dermatologist-level classification of skin cancer with deep neural networks', *Nature*. Nature Publishing Group, 542(7639), pp. 115–118. doi: 10.1038/nature21056.

Fe-Fei, L. F.-F. L., Fergus, R. and Perona, P. (2003) 'A Bayesian approach to unsupervised one-shot learning of object categories', *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 0–7. doi: 10.1109/ICCV.2003.1238476.

Fishburn, P. C. (1983) 'On the sphericity and cubicity of graphs', *Journal of Combinatorial Theory, Series B*, 35(3), pp. 309–318. doi: 10.1016/0095-8956(83)90057-6.

Fouss, F., Pirotte, A., Renders, J. M. and Saerens, M. (2007) 'Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation', *IEEE Transactions on Knowledge and Data Engineering*, 19(3), pp. 355–369. doi: 10.1109/TKDE.2007.46.

Gero, J. (1994) 'Towards a model of exploration in computer-aided design', *Formal Design Methods for CAD*.

Gips, J. (1975) *Shapes Grammars and their Uses Artificial Perception, Shape Generation and Computer Aesthetics*, *Artificial Intelligence*. doi: 10.1007/978-3-0348-5753-6.

Gips, J. (1999) 'Computer Implementation of Shape Grammars', *NSF Workshop on Shape Computation*, pp. 1–11.

Gobet, F., Lane, P., Croker, S., Cheng, P., Jones, G., Oliver, I. and Pine, J. (2001) 'Chunking mechanisms in human learning', *Trends in Cognitive Sciences*, 5(6), pp. 236–243. doi: 10.1016/S1364-6613(00)01662-4.

Hagberg, A. a., Schult, D. a. and Swart, P. J. (2008) 'Exploring network structure, dynamics, and function using NetworkX', in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pp. 11--15.

Hattemer, H. H. (1982) 'Genetic distance between populations', *TAG Theoretical and Applied Genetics*, 62(3), pp. 219–223. doi: 10.1007/BF00276242.

Helm, D., Labischinski, H., Schallehn, G. and Naumann, D. (1991) 'Classification and identification of bacteria by Fourier-transform infrared spectroscopy.', *Journal of general microbiology*, 137(1), pp. 69–79. doi: 10.1099/00221287-137-1-69.

Helmberger, M., Pienn, M., Urschler, M., Kullnig, P., Stollberger, R., Kovacs, G., Olschewski, A., Olschewski, H. and Bálint, Z. (2014) 'Quantification of tortuosity and fractal dimension of the lung vessels in pulmonary hypertension patients', *PLoS ONE*, 9(1). doi: 10.1371/journal.pone.0087515.

Hodge, V. J. and Austin, J. (2004) 'A survey of outlier detection methodologies', *Artificial Intelligence Review*, pp. 85–126. doi: 10.1023/B:AIRE.0000045502.10941.a9.

Holder, L. (1994) 'Substucture Discovery in the SUBDUE System .', (October 2014).

Jiang, C., Coenen, F. and Zito, M. (2013) 'A Survey of Frequent Subgraph Mining Algorithms', *The Knowledge Engineering Review*, 28(2), pp. 75–105. doi: 10.1017/S000000000000000.

Jones, M., Craddock, P. T. and Barker, N. (1990) 'Fake? : the art of deception', in *Book*, p. 312.

Kang, U., Tong, H. and Sun, J. (2012) 'Fast Random Walk Graph Kernel', in *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 828–838. doi: 10.1137/1.9781611972825.71.

Kaus, M. R., Warfield, S. K., Nabavi, A., Black, P. M., Jolesz, F. A. and Kikinis, R. (2001) 'Automated segmentation of MR images of brain tumors.', *Radiology*, 218(2), pp. 586–591. doi: 10.1148/radiology.218.2.r01fe44586.

Knuth, D. E. (1998) 'The Art of Computer Programming Volume 3. Sorting and Searching', *Addison Wesley*, 3, p. 829. doi: 10.2307/2005383.

Königseder, C. and Shea, K. (2014) 'Systematic rule analysis of generative design grammars', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 28(3), pp. 227–238. doi: 10.1017/S0890060414000195.

Königseder, C. and Shea, K. (2015) 'Analyzing Generative Design Grammars', in *Design Computing and Cognition '14*, pp. 363–381. doi: 10.1007/978-3-319-14956-1.

Kotsiantis, S. B. (2007) 'Supervised machine learning: A review of classification techniques', *Informatica*, 31, pp. 249–268. doi: 10.1115/1.1559160.

Kudo, T. and Matsumoto, Y. (2002) 'Japanese dependency analysis using cascaded chunking', *proceeding of the 6th conference on Natural language learning - COLING-02*, 20, pp. 1–7. doi: 10.3115/1118853.1118869.

Kuramochi, M. and Karypis, G. (2001) 'Frequent subgraph discovery', *Proceedings 2001 IEEE International Conference on Data Mining*, (Icdm), pp. 313–320. doi: 10.1109/ICDM.2001.989534.

Leach, P., Mealling, M. and Salz, R. (2005) 'A Universally Unique IDentifier (UUID) URN Namespace', *The Internet Society*, pp. 1–32.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, 521(7553), pp. 436–444. doi: 10.1038/nature14539.

Lee, Y. S. and Wu, Y. C. (2007) 'A robust multilingual portable phrase chunking system', *Expert*

Systems with Applications, 33(3), pp. 590–599. doi: 10.1016/j.eswa.2006.06.022.

McCormack, J. P. and Cagan, J. (2002) 'Designing inner hood panels through a shape grammar based framework', *AI EDAM*. Cambridge Univ Press, 16(4), pp. 273–290.

McCormack, J. P., Cagan, J. and Vogel, C. M. (2004) 'Speaking the Buick language: capturing, understanding, and exploring brand identity with shape grammars', *Design Studies*. Elsevier, 25(1), pp. 1–29.

McKay, B. D. and Piperno, A. (2014) 'Practical graph isomorphism, II', *Journal of Symbolic Computation*, 60, pp. 94–112. doi: 10.1016/j.jsc.2013.09.003.

Mehta, P. and David J. Schwab (2014) 'An exact mapping between the Variational Renormalization Group and Deep Learning', *arXiv*, pp. 1–7.

Michie, E. D., Spiegelhalter, D. J. and Taylor, C. C. (1994) 'Machine Learning , Neural and Statistical Classification', *Technometrics*, 37(4), p. 459. doi: 10.2307/1269742.

Mikolov, T., Le, Q. V and Sutskever, I. (2013) 'Exploiting Similarities among Languages for Machine Translation', *arXiv preprint arXiv:1309.4168v1*, pp. 1–10.

Minsky, M. L. (1991) 'Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy', *AI Magazine*, p. 34. doi: 10.1609/aimag.v12i2.894.

Nevill-manning, C. G. (1996) *Inferring Sequential Structure*.

Nevill-Manning, C. G. and Witten, I. H. (1997) 'Identifying Hierarchical Structure in Sequences: A linear-time algorithm', *arXiv preprint cs/9709102*, 7, pp. 67–82. doi: arXiv:cs/9709102.

Orsborn, S., Boatwright, P. and Cagan, J. (2008) 'Identifying product shape relationships using principal component analysis', *Research in Engineering Design*, 18(4), pp. 163–180. doi: 10.1007/s00163-007-0036-8.

Orsborn, S. and Cagan, J. (2009) 'Multiagent Shape Grammar Implementation: Automatically Generating Form Concepts According to a Preference Function', *Journal of Mechanical Design*, 131(12), p. 121007.

Orsborn, S., Cagan, J. and Boatwright, P. (2007) 'A methodology for creating a statistically derived shape grammar composed of non-obvious shape chunks', *Research in Engineering Design*. Design knowing and learning: Cognition in design …, 18(4), pp. 181–196.

Orsborn, S., Cagan, J. and Boatwright, P. (2009) 'Quantifying Aesthetic Form Preference in a Utility Function', *Journal of Mechanical Design*, 131(JUNE), p. 61001. doi: 10.1115/1.3116260.

Orsborn, S., Cagan, J., Pawlicki, R. and Smith, R. C. (2006) 'Creating cross-over vehicles: Defining and combining vehicle classes using shape grammars', *AIEDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 20(3), pp. 217–246. doi: 10.1017/S0890060406060185.

Pan, S. J. and Yang, Q. (2010) 'A survey on transfer learning', *IEEE Transactions on Knowledge and Data Engineering*, pp. 1345–1359. doi: 10.1109/TKDE.2009.191.

Piazzalunga, U. and Fitzhorn, P. (1998) 'Note on a three-dimensional shape grammar interpreter',

Environment and Planning B: Planning and Design, 25(1), pp. 11–30. doi: 10.1068/b250011.

Pinkwart, N., Aleven, V., Ashley, K. and Lynch, C. (2006) 'Toward legal argument instruction with graph grammars and collaborative filtering techniques', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 227–236. doi: 10.1007/11774303_23.

Pugliese, M. J. and Cagan, J. (2002) 'Capturing a rebel: modeling the Harley-Davidson brand through a motorcycle shape grammar', *Research in Engineering Design*, pp. 139–156. doi: 10.1007/s00163-002-0013-1.

Rawson, K. and Stahovich, T. F. (2009) 'Learning design rules with explicit termination conditions to enable efficient automated design', *Journal of Mechanical Design, Transactions of the ASME*, 131(3), pp. 310111–3101111. doi: 10.1115/1.3066681.

Rowe, C. (1977) 'Mathematics of the Ideal Villa and Other Essays', *Jae*, p. 48. doi: 10.2307/1424540.

Rozenberg, G. (1997) 'Handbook of graph grammars and computing by graph transformation', *Handbook of Graph Grammars*, 1, pp. 1–8. doi: 10.1142/9789812384720.

Sánchez-Martínez, F. and Pérez-Ortiz, J. A. (2010) 'Philipp Koehn, Statistical machine translation', *Machine Translation*, 24, pp. 273–278. doi: 10.1007/s10590-010-9083-4.

Savage, D., Zhang, X., Yu, X., Chou, P. and Wang, Q. (2014) 'Anomaly detection in online social networks', *Social Networks*, 39(1), pp. 62–70. doi: 10.1016/j.socnet.2014.05.002.

Schmidt, L. C. and Cagan, J. (1997) 'GGREADA: A graph grammar-based machine design algorithm', *Research in Engineering Design*, pp. 195–213. doi: 10.1007/BF01589682.

Schmidt, L. C. and Chase, S. C. (2000) 'A Graph Grammar Approach for Structure Synthesis', *Journal of Mechanical Design*, 122(DECEMBER), pp. 371–376. doi: 10.1115/1.1315299.

Schnier, T. and Gero, J. S. (1996) 'Learning Genetic Representations as Alternative to Hand-coded Shape Grammars', *Artificial Intelligence in Design*, pp. 39–57.

Schwenk, H. (2012) 'Continuous Space Translation Models for Phrase-Based Statistical Machine Translation', *COLING (Posters)*, (December), pp. 1071–1080.

Slisenko, A. O. (1982) 'Context-free grammars as a tool for describing polynomial-time subclasses of hard problems', *Information Processing Letters*, 14(2), pp. 52–56. doi: 10.1016/0020-0190(82)90086-2.

Smola, A., Gretton, A., Song, L. and Schölkopf, B. (2007) 'A Hilbert space embedding for distributions', *Proceedings of the 18th International Conference on Algorithmic Learning Theory*, pp. 13–31. doi: 10.1007/978-3-540-75225-7_5.

Sondheimer, N. K. and Weischedel, R. M. (1980) 'A rule-based approach to ill-formed input', in *COLING 1980*, pp. 46–53.

Speller, T. H., Whitney, D. and Crawley, E. (2007) 'Using shape grammar to derive cellular automata rule patterns', *Complex Systems- ...*, 17(1/2), pp. 79–102.

Stiny, G. (1980) 'Introduction to shape and shape grammars', *Environment and planning B*, 7(3), pp.

343–351. doi: 10.1068/b070343.

Stiny, G. and Gips, J. (1972) 'Shape grammars and the generative specification of painting and sculpture', *Information Processing 71 Proceedings of the IFIP Congress 1971. Volume 2*, 71, pp. 1460–1465. doi: citeulike-article-id:1526281.

Stiny, G. and Mitchell, W. J. (1978) 'The Palladian grammar', *Environment and Planning B: Planning and Design*, pp. 5–18. doi: 10.1068/b050005.

Stolcke, A. and Omohundro, S. (1994) 'Inducing probabilistic grammars by Bayesian model merging', *Grammatical Inference and Applications*, pp. 106–118.

Suh, N. (2001) *Axiomatic Design: Advances and Applications (The Oxford Series on Advanced Manufacturing)*, *Clinical neurology and neurosurgery*. doi: 10.1016/j.clineuro.2008.04.006.

Talton, J., Yang, L., Kumar, R., Lim, M., Goodman, N. and Měch, R. (2012) 'Learning design patterns with bayesian grammar induction', *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, p. 63. doi: 10.1145/2380116.2380127.

Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P. and Paragios, N. (2011) 'Shape grammar parsing via reinforcement learning', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2273–2280. doi: 10.1109/CVPR.2011.5995319.

Teboul, O., Simon, L., Koutsourakis, P. and Paragios, N. (2010) 'Segmentation of building facades using procedural shape priors', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3105–3112. doi: 10.1109/CVPR.2010.5540068.

Trescak, T., Esteva, M. and Rodriguez, I. (2012) 'A shape grammar interpreter for rectilinear forms', *CAD Computer Aided Design*, 44(7), pp. 657–670. doi: 10.1016/j.cad.2012.02.009.

Trescak, T., Rodriguez, I. and Esteva, M. (2009) 'General shape grammar interpreter for intelligent designs generations', in *Proceedings of the 2009 6th International Conference on Computer Graphics, Imaging and Visualization: New Advances and Trends, CGIV2009*, pp. 235–240. doi: 10.1109/CGIV.2009.74.

Wallach, H. M. (2006) 'Topic Modeling: Beyond Bag-of-Words', *ICML*, (1), pp. 977–984. doi: 10.1145/1143844.1143967.

Wand, Y. and Weber, R. (1993) 'On the ontological expressiveness of information systems analysis and design grammars', *Information Systems Journal*, 3(4), pp. 217–237. doi: 10.1111/j.1365-2575.1993.tb00127.x.

Whiting, M., Cagan, J. and Leduc, P. (2017) 'Efficient Probabilistic Grammar Induction for Design Efficient Probabilistic Grammar Induction for Design', *AIEDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, pp. 1–31.

Wootters, W. K. (1981) 'Statistical distance and Hilbert space', *Physical Review D*, 23(2), pp. 357–362. doi: 10.1103/PhysRevD.23.357.

Wu, Z. and Song, S. (2015) '3D ShapeNets : A Deep Representation for Volumetric Shapes', pp. 1912–1920. doi: 10.1109/CVPR.2015.7298801.

Yan, X. and Han, J. (2002) ‘gSpan: Graph-Based Substructure Pattern Mining’, in *Proceedings of IEEE International Conference on Data Mining*, pp. 721–724. doi: 10.1109/ICDM.2002.1184038.

Yue, K. and Krishnamurti, R. (2013) ‘Tractable shape grammars’, *Environment and Planning B: Planning and Design*, 40(4), pp. 576–594. doi: 10.1068/b38227.

Zhang, Y., Bazilevs, Y., Goswami, S., Bajaj, C. L. and Hughes, T. J. R. (2006) ‘Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow’, in *Proceedings of the 15th International Meshing Roundtable, IMR 2006*, pp. 73–92. doi: 10.1007/978-3-540-34958-7_5.

Appendices

Apendix 1

The full set of 47 induced and simplified grammar rules from the patient data that was processed. Note, the grammar rules have been simplified for visualization purposes. For instance, many left hand side tokens of rules here are depicted as a linear section of vessel which indicates a relatively arbitrary segment of vessel. Additionally, feature scale is not explicitly defined in this grammar approach so scale and orientation may change more than is visually obvious in a singular representation of each rule. Lastly, these are 2D projections of 3D vessel networks, so some rules include vessels that overlap in this visualization but do not intersect, these situations are marked with a small black dot in the rules.


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

Appendix 2

Regression tables for each of the 18 models are provided.

Condition 1 as predicted by Rule 39

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.952474   0.187807   5.072 0.000138 ***
R39         -0.007339   0.002696  -2.722 0.015744 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4347 on 15 degrees of freedom
Multiple R-squared:  0.3307,    Adjusted R-squared:  0.286
F-statistic:  7.41 on 1 and 15 DF,  p-value: 0.01574
```

Condition 2 as predicted by Rule 6

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.07832    0.21625   4.986 0.000163 ***
R6          -0.03263    0.01129  -2.889 0.011236 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4259 on 15 degrees of freedom
Multiple R-squared:  0.3575,    Adjusted R-squared:  0.3147
F-statistic: 8.348 on 1 and 15 DF,  p-value: 0.01124
```

Condition 3 as predicted by Rule 21

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.012186   0.176276   0.069  0.94580
R21         0.008287   0.002372   3.494  0.00326 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3946 on 15 degrees of freedom
Multiple R-squared:  0.4487,    Adjusted R-squared:  0.4119
F-statistic: 12.21 on 1 and 15 DF,  p-value: 0.003265
```

Condition 4 as predicted by Rule 27

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  0.984017   0.208642   4.716 0.000276 ***
R27         -0.024691   0.009705  -2.544 0.022458 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4441 on 15 degrees of freedom
Multiple R-squared:  0.3014,    Adjusted R-squared:  0.2549
F-statistic: 6.473 on 1 and 15 DF,  p-value: 0.02246
```

Condition 5 as predicted by Rule 44

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.186505   0.199447   0.935    0.365
R44         0.020146   0.008407   2.396    0.030 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4455 on 15 degrees of freedom
Multiple R-squared:  0.2769,    Adjusted R-squared:  0.2287
F-statistic: 5.743 on 1 and 15 DF,  p-value: 0.03003
```

Condition 6 as predicted by Rule 29

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.158810   0.254437  -0.624  0.54190
R29          0.012732   0.003744   3.401  0.00395 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3823 on 15 degrees of freedom
Multiple R-squared:  0.4354,    Adjusted R-squared:  0.3978
F-statistic: 11.57 on 1 and 15 DF,  p-value: 0.003949
Key Rules ules together models’ regression tables
```

Condition 1 as predicted by Rules 39, 6, 21, 27, 44, 29

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.6480736  0.5140497   1.261   0.2360
R39         -0.0069824  0.0028399  -2.459   0.0338 *
R6          -0.0183819  0.0156622  -1.174   0.2677
R21          0.0000135  0.0034917   0.004   0.9970
R27          0.0060207  0.0098786   0.609   0.5558
R44         -0.0013645  0.0097857  -0.139   0.8919
R29          0.0080358  0.0049812   1.613   0.1378
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4392 on 10 degrees of freedom
Multiple R-squared:  0.5445,    Adjusted R-squared:  0.2712
F-statistic: 1.992 on 6 and 10 DF,  p-value: 0.1602
```

Condition 2 as predicted by Rules 39, 6, 21, 27, 44, 29

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.404695   0.522715   2.687   0.0228 *
R39          0.001563   0.002888   0.541   0.6002
R6          -0.045881   0.015926  -2.881   0.0164 *
R21         -0.004131   0.003551  -1.163   0.2717
R27         -0.010914   0.010045  -1.087   0.3027
R44         -0.004757   0.009951  -0.478   0.6429
R29          0.005689   0.005065   1.123   0.2876
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4466 on 10 degrees of freedom
Multiple R-squared:  0.529, Adjusted R-squared:  0.2464
F-statistic: 1.872 on 6 and 10 DF,  p-value: 0.1818
```

Condition 3 as predicted by Rules 39, 6, 21, 27, 44, 29

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.4332498  0.4139706   1.047  0.31993
R39          0.0005442  0.0022870   0.238  0.81673
R6           0.0045094  0.0126129   0.358  0.72813
R21          0.0105579  0.0028119   3.755  0.00375 **
R27         -0.0012490  0.0079554  -0.157  0.87837
R44          0.0024628  0.0078806   0.313  0.76107
R29         -0.0109983  0.0040114  -2.742  0.02077 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3537 on 10 degrees of freedom
Multiple R-squared:  0.7046,    Adjusted R-squared:  0.5273
F-statistic: 3.975 on 6 and 10 DF,  p-value: 0.02693
```

Condition 4 as predicted by Rules 39, 6, 21, 27, 44, 29

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.2934544  0.6217659   2.080   0.0642 .
```

```
R39         -0.0009515  0.0034350  -0.277   0.7874
R6          -0.0007726  0.0189441  -0.041   0.9683
R21         -0.0006754  0.0042234  -0.160   0.8761
R27         -0.0231766  0.0119486  -1.940   0.0811 .
R44         -0.0046641  0.0118363  -0.394   0.7018
R29         -0.0021220  0.0060250  -0.352   0.7320
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5313 on 10 degrees of freedom
Multiple R-squared:  0.3336,    Adjusted R-squared:  -0.06629
F-statistic: 0.8342 on 6 and 10 DF,  p-value: 0.5701
```

Condition 5 as predicted by Rules 39, 6, 21, 27, 44, 29

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.644858   0.489014   1.319   0.2167
R39         -0.002759   0.002702  -1.021   0.3313
R6          -0.014607   0.014899  -0.980   0.3500
R21         -0.001466   0.003322  -0.441   0.6683
R27         -0.018252   0.009397  -1.942   0.0808 .
R44          0.020484   0.009309   2.200   0.0524 .
R29          0.005802   0.004739   1.224   0.2488
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4178 on 10 degrees of freedom
Multiple R-squared:  0.576, Adjusted R-squared:  0.3216
F-statistic: 2.264 on 6 and 10 DF,  p-value: 0.1214
```

Condition 6 as predicted by Rules 39, 6, 21, 27, 44, 29

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.734e-01  5.045e-01  -0.344   0.7382
R39          4.668e-05  2.787e-03   0.017   0.9870
R6           9.408e-03  1.537e-02   0.612   0.5541
R21         -8.152e-04  3.427e-03  -0.238   0.8168
R27          5.465e-03  9.694e-03   0.564   0.5854
R44         -5.523e-03  9.603e-03  -0.575   0.5779
R29          1.137e-02  4.888e-03   2.327   0.0423 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.431 on 10 degrees of freedom
Multiple R-squared:  0.5214,    Adjusted R-squared:  0.2343
F-statistic: 1.816 on 6 and 10 DF,  p-value: 0.1929
1:20 rules models’ regression tables
```

Condition 1 as predicted by Rules 6, 9, 13, 16, 21, 26, 27, 29, 37, 39, 40, 44, 46

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.0161236	1.0510162	0.967	0.405
R6	-0.0476424	0.0280170	-1.700	0.188
R9	0.0057517	0.0088970	0.646	0.564
R13	-0.0004716	0.0247245	-0.019	0.986
R16	0.0054677	0.0078664	0.695	0.537
R21	-0.0052966	0.0068166	-0.777	0.494
R26	-0.0079283	0.0050983	-1.555	0.218
R27	0.0162128	0.0147424	1.100	0.352
R29	0.0118516	0.0099565	1.190	0.320
R37	0.0008158	0.0102244	0.080	0.941
R39	-0.0036944	0.0049171	-0.751	0.507
R40	-0.0162696	0.0266353	-0.611	0.584
R44	-0.0143228	0.0150392	-0.952	0.411
R46	0.0038554	0.0039192	0.984	0.398

Residual standard error: 0.4485 on 3 degrees of freedom
Multiple R-squared: 0.8575, Adjusted R-squared: 0.2403
F-statistic: 1.389 on 13 and 3 DF, p-value: 0.4423

Condition 2 as predicted by Rules 6, 9, 13, 16, 21, 26, 27, 29, 37, 39, 40, 44, 46

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.377e+00	9.532e-01	1.445	0.244
R6	-5.552e-02	2.541e-02	-2.185	0.117
R9	1.245e-02	8.069e-03	1.543	0.221
R13	1.003e-02	2.242e-02	0.447	0.685
R16	-7.909e-03	7.134e-03	-1.109	0.348
R21	-3.847e-03	6.182e-03	-0.622	0.578
R26	-2.479e-03	4.624e-03	-0.536	0.629
R27	-1.228e-02	1.337e-02	-0.918	0.426
R29	3.931e-03	9.030e-03	0.435	0.693
R37	-1.499e-02	9.273e-03	-1.617	0.204
R39	1.634e-03	4.460e-03	0.366	0.738
R40	2.004e-02	2.416e-02	0.829	0.468
R44	1.019e-02	1.364e-02	0.747	0.509
R46	-7.632e-05	3.555e-03	-0.021	0.984

Residual standard error: 0.4067 on 3 degrees of freedom
Multiple R-squared: 0.8828, Adjusted R-squared: 0.3751
F-statistic: 1.739 on 13 and 3 DF, p-value: 0.3586

Condition 3 as predicted by Rules 6, 9, 13, 16, 21, 26, 27, 29, 37, 39, 40, 44, 46

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	7.046e-01	1.223e+00	0.576	0.605
R6	4.468e-03	3.261e-02	0.137	0.900
R9	-1.882e-03	1.036e-02	-0.182	0.867
R13	7.483e-04	2.878e-02	0.026	0.981
R16	-9.415e-04	9.157e-03	-0.103	0.925
R21	1.345e-02	7.935e-03	1.695	0.189
R26	-6.522e-05	5.935e-03	-0.011	0.992
R27	-3.803e-03	1.716e-02	-0.222	0.839
R29	-8.987e-03	1.159e-02	-0.775	0.495
R37	-4.952e-03	1.190e-02	-0.416	0.705
R39	-2.197e-03	5.724e-03	-0.384	0.727
R40	-2.275e-02	3.100e-02	-0.734	0.516
R44	3.742e-03	1.751e-02	0.214	0.844
R46	1.060e-03	4.562e-03	0.232	0.831

Residual standard error: 0.522 on 3 degrees of freedom
Multiple R-squared: 0.807, Adjusted R-squared: -0.02943
F-statistic: 0.9648 on 13 and 3 DF, p-value: 0.5911

Condition 4 as predicted by Rules 6, 9, 13, 16, 21, 26, 27, 29, 37, 39, 40, 44, 46

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.547263	1.662674	1.532	0.223
R6	-0.025234	0.044322	-0.569	0.609
R9	-0.003108	0.014075	-0.221	0.839
R13	-0.007918	0.039113	-0.202	0.853
R16	0.007910	0.012444	0.636	0.570
R21	-0.009467	0.010784	-0.878	0.445
R26	-0.004801	0.008065	-0.595	0.594
R27	-0.017969	0.023322	-0.770	0.497
R29	0.010448	0.015751	0.663	0.555
R37	0.012672	0.016175	0.783	0.491
R39	-0.004422	0.007779	-0.568	0.610
R40	-0.021986	0.042136	-0.522	0.638
R44	-0.019628	0.023792	-0.825	0.470
R46	-0.003313	0.006200	-0.534	0.630

Residual standard error: 0.7094 on 3 degrees of freedom
Multiple R-squared: 0.6435, Adjusted R-squared: -0.9014
F-statistic: 0.4165 on 13 and 3 DF, p-value: 0.8853

Condition 5 as predicted by Rules 6, 9, 13, 16, 21, 26, 27, 29, 37, 39, 40, 44, 46

Coefficients:

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.2425816  0.7473009  -0.325    0.767
R6          -0.0166533  0.0199209  -0.836    0.465
R9           0.0044095  0.0063260   0.697    0.536
R13          0.0213104  0.0175798   1.212    0.312
R16         -0.0056028  0.0055932  -1.002    0.390
R21         -0.0059824  0.0048468  -1.234    0.305
R26         -0.0023486  0.0036251  -0.648    0.563
R27         -0.0054686  0.0104823  -0.522    0.638
R29          0.0001924  0.0070794   0.027    0.980
R37          0.0040234  0.0072698   0.553    0.619
R39          0.0043714  0.0034962   1.250    0.300
R40          0.0562807  0.0189384   2.972    0.059 .
R44          0.0206813  0.0106933   1.934    0.149
R46          0.0007762  0.0027867   0.279    0.799
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3189 on 3 degrees of freedom
Multiple R-squared:  0.9259,    Adjusted R-squared:  0.6049
F-statistic: 2.885 on 13 and 3 DF,  p-value: 0.2078
```

Condition 6 as predicted by Rules 6, 9, 13, 16, 21, 26, 27, 29, 37, 39, 40, 44, 46

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.0687371  0.8613625  -1.241    0.303
R6           0.0332268  0.0229614   1.447    0.244
R9           0.0009424  0.0072915   0.129    0.905
R13          0.0105955  0.0202630   0.523    0.637
R16         -0.0040567  0.0064469  -0.629    0.574
R21          0.0061441  0.0055865   1.100    0.352
R26          0.0085117  0.0041784   2.037    0.134
R27         -0.0124972  0.0120822  -1.034    0.377
R29          0.0021892  0.0081599   0.268    0.806
R37         -0.0008290  0.0083794  -0.099    0.927
R39         -0.0018143  0.0040298  -0.450    0.683
R40          0.0043360  0.0218290   0.199    0.855
R44          0.0177030  0.0123254   1.436    0.246
R46         -0.0022358  0.0032120  -0.696    0.536

Residual standard error: 0.3675 on 3 degrees of freedom
Multiple R-squared:  0.8956,    Adjusted R-squared:  0.4433
F-statistic:  1.98 on 13 and 3 DF,  p-value: 0.3145
```