A Comparison Between Statistical and Symbolic Learning Approaches for Generative Policy Models

Graham White*, Daniel Cunnington*, Mark Law†, Elisa Bertino‡, Geeth De Mel* and Alessandra Russo†
*IBM Research, Winchester, UK. Email: {gwhite,dancunnington}@uk.ibm.com
†Imperial College, London, UK. Email: {mark.law09,a.russo}@imperial.ac.uk
‡Purdue University, West Lafayette, IN, USA. Email: bertino@purdue.edu

***Abstract*—Generative Policy Models (GPMs) have been proposed as a method for future autonomous decision making in a distributed, collaborative environment. To learn a GPM, previous policy examples that contain policy features and the corresponding policy decisions are used. Recently, GPMs have been constructed using both symbolic and statistical learning algorithms. In either case, the goal of the learning process is to create a model across a wide range of contexts from which specific policies may be generated in a given context. Empirically, we expect each learning approach to provide certain advantages over the other. This paper assesses the relative performance of each learning approach in order to examine these advantages and disadvantages. Several carefully prepared data sets are used to train a variety of models across different learning algorithms, where models for each learning algorithm are trained with varying amounts of labelled examples. The performance of each model is evaluated across a variety of metrics which indicates the strength of each learning algorithm for the different scenarios presented and the amount of training data provided. Finally, future research directions are outlined to fully realise GPMs in a distributed, collaborative environment.**

***Index Terms*—Distributed Analytics, Information Science, Collaborative Environments, Symbolic Learning, Machine Learning, Generative Policy Model, Generated Policies**

I. Introduction

As technology continues to develop, the autonomous capability of intelligent systems is advancing. Managing the actions and behaviours of these systems is becoming a complex and less feasible task for humans as the number of systems and their complexity increases [1]. Traditional approaches for managing the behaviour of intelligent systems rely on a predefined set of policies to enforce the underlying goals and constraints, controlled by a central policy management system. However, existing policy-based approaches do not provide practical methods for handling the adaptability required for the challenging, distributed and dynamic environments in which intelligent systems will operate. The notion of Generative Policy Models (GPMs) has been proposed [2] as a means by which intelligent systems can autonomously manage their actions and behaviours in complex environments.

The use of GPMs involves learning the policies to be obeyed in a given scenario across a variety of different contexts. This can be achieved through continuous observation of policy examples over a period of time, so that policies can be learned and refined according to contextual changes. Policy examples can be used across systems and devices in a distributed environment to collaboratively learn a GPM. Once a suitable GPM has been learned, it can be used to generate new policies within any of the observed contexts. The GPM can be represented in a number of different forms such as a list of logical rules or a machine learning model. The utility of generative policies has recently been demonstrated in access control [3], logistical resupply [4] and collaborative environments [5] that all require learning a GPM in order to generate new policies across a variety of contexts. The use of GPMs within collaborative environments is motivated by the requirement for autonomous system behaviour such that decisions can be made seamlessly between distributed collaborative partners.

In order to facilitate the use of GPMs in such scenarios, frameworks are being developed such as the ASGrammar-based GENerative Policy (AGENP) framework [6] where state-of-the-art Answer Set Grammars (ASGs) are used to facilitate the learning of GPMs via Inductive Learning of Answer Set Programs (ILASP) [7]—a symbolic learning approach based on Inductive Logic Programming. Symbolic learning approaches such as ILASP are fully explainable due to their white-box nature where learned rules can be expressed in English via a trivial translation. This is important and required for safety-critical domains such as coalition operations [8]. Currently, our symbolic learning algorithm, ILASP, may take a long time to learn complex policies due to the need to pre-compute data (known as the hypothesis space) on which inferences can be made. This may be a limiting factor if near real-time decision making is required in a collaborative environment.

This paper performs a thorough comparison of the relative advantages and disadvantages of statistical and symbolic machine learning approaches, where the symbolic ASG-based approach is evaluated against state-of-the-art statistical machine learning and deep learning approaches across varying numbers of training examples, different data sets and different policy metrics. This demonstrates the suitability of the ASG-based approach for learning GPMs in collaborative environments as well as outlining future research directions to realise this vision. The paper is structured as follows. In section II we present relevant background material, in section III we discuss related work and in section IV we outline our experimental approach. In section V we present our results and perform an evaluation with respect to policy metrics and finally, we discuss future work and conclude in sections VI and VII.

II. Background

In this section, we introduce the GPM architecture; provide an example-driven explanation of the ASG-based symbolic learning approach; discuss statistical approaches to GPM learning; and outline a set of policy metrics that can be used to evaluate each learning technique.

A. *Policy Learning*

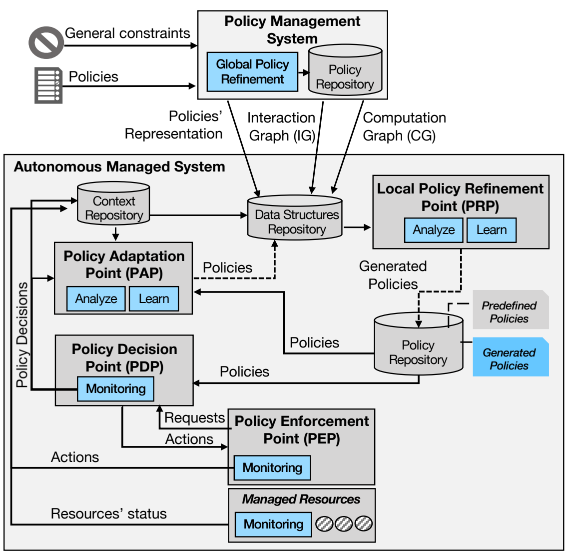


Fig. 1: A GPM Architecture [6]

Figure 1 details the GPM architecture as introduced by Calo et al. [6]. Within the *Policy Adaptation Point* and the *Local Policy Refinement Point*, a learning component is used to learn the GPM from annotated policy examples—i.e. policies that are annotated with a previous decision. The Local Policy Refinement Point enables an autonomous system to learn its own GPM based on its operating capability and constraints, whereas the Policy Adaptation Point enables new contextual experiences as well as outcomes from previous policy decisions to be analysed to update and refine the GPM. We now introduce two variants of this architecture that include two different learning components: *(1)* the ASG-based symbolic learning approach; and *(2)* a statistical machine learning approach.

1) Learning ASGs via ILASP: In order to learn a GPM using ILASP, the policy domain alongside positive and negative examples must be represented as an ASG learning task. ASGs extend Context-Free Grammars with context sensitive conditions written in the language of Answer Set Programming [9]. This enables the learning of GPMs that each take into account a variety of contexts. Once learned, a GPM can be used to generate a policy when given a specific context. An ASG learning task consists of a set of *production rules* that define the structure of the learning task, a set of *constants and variables* present in the domain, a set of positive and negative examples with contextual annotations that represent *background knowledge* and finally a set of *mode declarations* that outline the *hypothesis space* that represents the set of rules that can form possible learned hypotheses. Figure 2 lists an example ASG representing a subset of the Coalition Asset Loaning learning task that is introduced in section IV. Note that various production rules, constants, variables and examples have been omitted due to space constraints. Full listings of the data sets used in this paper and their associated ASGs are available at *https://github.com/dais-ita/asg-ml-comparison*.

The learning task represented in ASG form can be solved by ILASP—an Inductive Learner of Answer Set Programs [10], [11]. Once the learning task is solved, a learned ASG is returned with the root production rule containing learned rules according to the hypothesis space and the examples given. These rules can be used to generate a policy in the form of strings in the language of the learned grammar in a given context. For example, in the Coalition Asset Loaning learning task, the learned GPM may contain the rule $autonomy\_level \leq 6\ AND\ weather\_score \leq 5000$ to accept an asset loan request. Inputting contextual features to the GPM, such as a value for $weather\_score$, generates a policy that defines the set of non-contextual asset attributes that indicate acceptable asset loan requests in this context (e.g. when $weather\_score > 5000$, assets must have $autonomy\_level > 6$ in order to accept a loan request). For a detailed explanation on the ASG-based learning approach, we refer the reader to [7].

```
% root production rule
start -> requestor "; " asset_worth "; "
    trust "; " mission_type {}

% production rules for features
asset_risk -> "11" {asset_risk(11).}
trust -> "9" {trust(9).}

% constants and variables
#constant(asset_worth, 12).
#constant(weather_score, 2277).

% positive and negative examples
+ ["KISH", "; ", "37", "; ", "3", "; ",
    "person of interest tracking"]{
    weather_score(1700).
    physical_constraint("inactive").}
- ["UK", "; ", "9", "; ", "4", "; ",
    "logistical resupply"]{weather_score(8850).
    physical_constraint("inactive").}

% mode declarations
#modeba(1, requestor(const(requestor))):[1].
#modebb(1, var(asset_risk) < const(asset_risk),
    (positive)):[1].
```

Fig. 2: Example ASG representing a subset of the Coalition Asset Loaning learning task

2) Statistical Approaches: In order to learn a GPM via a statistical machine learning approach, state-of-the-art explainability techniques may be required as for many applications of GPMs, including safety-critical collaborative environments, explainability of the GPM is paramount. Explainability enables managing parties to visualise the set of potential behaviours and actions for a system in a range of different contexts to validate these actions are intentional and within the bounds of acceptable behaviour. Also, the ability to reduce a learned model to a list of rules expressed in English enables the sharing of learned models in distributed, low-bandwidth environments. Ultimately, explainability provides confidence to human users and administrators of any system because they can see why a decision has been taken in addition to the decision value.

Within statistical machine learning, ‘white-box’ models (models that are inherently interpretable) such as decision trees can be used, provided an explanation can be generated to describe the entire model. To account for ‘black-box’ models such as support vector machines or neural networks that do not exhibit explainable properties, various techniques can be used in order to extract learned rules. For example, as evident in [12], surrogate white-box models that exhibit explainable properties can be used to approximate existing black-box models by training the surrogate model on a smaller set of training samples labelled with predictions from the existing black-box model. Also, genetic algorithms to evolve locally interpretable explanations [13] can be used to infer rules that cover the entire model. However, care should be taken when using a black-box learner to learn a GPM as the choice of explanation method may introduce a layer of uncertainty to the resulting rule set. This choice may also effect how interpretable the explanation is [14]. For a comprehensive review of explainability techniques for statistical machine learning, we refer the reader to [15]. In this paper, we do not perform any statistical machine learning explanation, but acknowledge this is required when using a black-box statistical learner within a GPM architecture.

B. Policy Metrics

In order to evaluate the two learning approaches, a comprehensive set of policy metrics are required. A set of metrics for evaluating the quality of access control policies has been introduced [16]. We utilise these when evaluating our experiments, in particular the learning component in a GPM. Specifically, these include: *(1) consistency*—the GPM must always generate the same set of policies in a given context assuming the GPM has not been refined; *(2) relevance*—the GPM must generate policies that apply in the given context. Irrelevant policies may undermine correct operation, security or confidence; *(3) minimality*—the GPM must not generate redundant policies that permit a greater range of behaviour than is required; *(4) completeness*—the GPM must cover all possible contextual and non-contextual attributes required in order for the system to operate; *(5) correctness*—the learning component within a GPM architecture must have learned a desirable outcome in order to avoid generating incorrect policies; *(6) enforceability*—generated policies must be enforceable at the time of action. If the generated policy depends on the acquisition of contextual information and the contextual information is unavailable, the policy may not be able to be enforced autonomously and may require human intervention. In a collaborative environment, enforceability also applies to any learning time required within the GPM architecture and any bandwidth constraints that may be present when sharing generated policies with other devices. Finally, *(7) explainability*—the GPM must be explainable to allow human operators to validate the set of generated policies in varying contexts conform to desired system behaviour and actions.

III. Related Work

Within the field of artificial intelligence, statistical and symbolic learning algorithms have been pursued as competing methods for supervised learning tasks. The symbolic paradigm has largely focused on simulating human reasoning using explicit symbols whereas the statistical approach (otherwise referred to as a connectionist, non-symbolic or sub-symbolic) has largely focused on simulating the neural processes within the human brain [17]. Shavlik et al. [18] perform an early comparison between the symbolic ID3 decision tree and neural learning algorithms such as the perceptron and backpropagation, evaluating the algorithms with respect to accuracy and training time. The authors also investigate varying the number of training examples, adding noise and dropping feature values in order to understand the effect on classification performance. The authors conclude the backpropagation algorithm takes longer to train, but outperforms the ID3 and perceptron algorithms when learning over numerical features. Also, with smaller amounts of training data and with noisy examples backpropagation outperforms the other algorithms. The authors report that reducing the number of features resulted in minimal loss of classification performance across all algorithms. Foggia et al. [19] also conducted a comparative study between the two learning paradigms with a focus on learning structured data using a symbolic attributed relational graph approach and a neural network. The authors found that whilst classification performance between the two approaches was similar, each approach correctly classified examples that were mis-classified by the other approach. As a result, an integration between the two approaches to achieve greater performance was proposed. In this paper, we focus our experiments on an emerging symbolic learning technique based on ASGs with a comparison to state-of-the-art statistical learning algorithms. We then evaluate our results with respect to GPM metrics.

IV. Experimental Approach

This section explains our experimental approach which focuses on analysing different statistical and symbolic learning algorithms across different data sets and different amounts of data per data set. We also describe how the data was prepared and outline the specification of the benchmark machine used in our analysis.

A. Data Sets

When considering the selection of data sets, the presence of contextual features is important since this type of feature is present in the generative policy data we wish to model. It is also important to consider data sets with varying amounts of training examples, features and feature complexity. We

Data Set	Size	Features	Feature Complexity	Contextual Features
Coalition	Small	16	Low	Weather, Physical Constraints
Mushroom	Small	22	High	Habitat
Adult	Large	14	Medium	Age, Sex

TABLE I: Data Set Summary

selected 3 data sets as summarised in Table I and explained as follows:

1) Coalition Asset Loaning: A synthetic, multivariate binary classification data set containing 1,000 samples. It was created for a demonstration of GPMs [5] in coalition environments. There are 16 features containing a mix of categorical and numeric data types. Features include asset autonomy level, asset worth and a trust level between coalition partners. This synthetic data was generated to reflect a wide range of coalition asset loaning requests in varying contexts.

This data set was selected due to its small size, limited complexity and coalition relevance. It contains a relatively small number of features and low variance in the feature values. The data set contains 2 contextual features: 'Weather' and 'Physical Constraints' (e.g. a no-fly zone). From previous work [5], it is possible to learn a satisfiable GPM with both statistical and symbolic approaches and hence provides a good benchmark.

2) Mushroom Edibility[1]*:* A multivariate binary classification data set to determine if a mushroom is edible or poisonous, containing 8,124 examples covering 23 distinct mushroom species. There are 22 features, all of which are categorical with no numeric features. These include features related to the mushroom cap, odor and stalk. It is known that there is no simple rule for determining the edibility of a mushroom [20].

This data set was selected due to its relatively small size and presence of the contextual feature, 'Habitat'. It also contains a high level of complexity due to the number of mushroom species included and the feature variance.

3) Adult Income[2]*:* A multivariate binary classification data set to determine whether a person's income exceeds $50K USD per year, containing 48,842 samples. There are 14 features containing a mix of categorical and numeric types. These include, for example, age, education level and occupation. The data was extracted from the 1994 US census database.

This data set was selected due to its large size and presence of contextual features, 'Age' and 'Sex'. It has a large number of examples, categorical features with many values and continuous numeric features that result in a complex learning task.

B. Data Preparation

All data sets have been prepared using exactly the same methods as outlined below.

[1]Mushroom Edibility, https://archive.ics.uci.edu/ml/datasets/Mushroom
[2]Adult Income, https://archive.ics.uci.edu/ml/datasets/adult

No assumptions have been made about the order of the data within each data set. Hence, the data was first sorted in ascending order starting with the categorical features (working left-to-right across the feature columns) and then by the features considered to have the greatest indicative impact on classification. To identify the column sorting order for the non-categorical features we trained a Random Forest (RF) and used the output of the feature importances (the *feature_importances_* given by the SciKit-Learn RF Classifier) to determine the order in which feature columns should be sorted. The intention of this approach is to deliberately order the data for the next step in data preparation.

A validation set was created with an 80% (training) and 20% (validation) split. This was achieved by taking every 5th line from the sorted file and holding those samples out from the remaining data set in a separate validation file. By selecting every 5th line from a sorted data set we can ensure we are selecting at least some of each data from each feature in the data set.

Using the remaining 80% of the data for training (hereafter known as the training set), we wish to perform a number of experiments with differing amounts of training data. Hence, from the training set (that represents 100% of the training data) we created 9 additional sets of training data per data set. A 1% set was created by selecting every nth line of the training set in order to achieve a total of 1% and for the same reasons as previously, selecting periodically along a set of ordered rows gives us a spread of feature values throughout the entire data set so we are assured of a fully representative sample. Similarly, a 2%, 3%, 4%, 5%, 10%, 25%, 50% and 75% training set files were created by selecting the relevant periodic lines throughout the 100% training set file. We deliberately concentrated on a larger number of smaller percentages since we are interested in measuring how much data might be required to accurately learn a GPM in a real-world collaborative environment, such as IoT based applications.

1) Preparation for Statistical Experiments: All 3 selected data sets include categorical features that, for statistical learning only, require encoding. In our experiments we chose to use one-hot encoding. Hence, an additional step for all 10 training data files fed to the statistical learning algorithms was to one-hot encode the categorical features. The symbolic learning approach does not require such encoding due to the grammatical nature of the system we are using.

2) Preparation for Symbolic Experiments: As we are using the ASG-based symbolic learner, the symbolic experiments must be formulated as an ASG learning task. The ASG learning task representing the tabular training data is then fed into the symbolic learning process. Additionally, for the symbolic learning approach only, a hypothesis space must be calculated that outlines the set of possible policies that can be learned. This is performed as a pre-processing step on each training ASG learning task fed into the symbolic learner.

C. Algorithms and Method

Given our comparison between statistical and symbolic machine learning, we have chosen a set of algorithms that fit into these categories. For statistical learning we selected two shallow learning algorithms: a tree method using a RF; and a General Linear Model (GLM). We also selected one deep learning algorithm and chose a Fully Connected Feed-Forward Network (FCN). For symbolic learning we use the ASG method [7], [10]. The statistical algorithms were chosen to provide a range of differing techniques that are mature and widely used within the field of machine learning. The RF was trained using the SciKit-Learn libraries, the GLM was trained using H2O libraries, and the FCN was trained using Keras with a TensorFlow (CPU) back end.

In order to reflect real-world usage, we subjected all of the statistical techniques to hyper-parameter tuning in the form of a grid search for each percentage of training data. For the RF we tuned the number of estimators across $\{100, 200, 500\}$. For the GLM we tuned the regularisation strength (λ) across $\{1, 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0\}$. For the FCN we performed a hyper-parameter search across the L2 regularisation and dropout rate parameters as well as experimenting with varying network shapes. For L2 regularisation, we tuned across $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and for dropout rates we tuned across $\{0, 0.1, 0.2, 0.3, 0.4\}$. For network shape, we selected: 1X 8-node hidden layer for the Mushroom Edibility data set; 2X 4-node hidden layers for the Adult Income data set; and 3X hidden layers of 16-node, 8-node and 4-node for the Coalition Asset Loaning data set. These shapes were selected from a range of experiments and represent the best performing model. In all hyper-parameter tuning, we selected the combination of hyper-parameters that resulted in the best performing model.

Similarly, we applied tuning to the symbolic learning in the form optimising the grammar to reduce the search space calculation. The symbolic learning approach has parameters representing the depth of the search, the maximum number of literals, the maximum rule length, and the number of threads and batch size per thread. Tuning of these is time and memory consuming so the tuning process for the symbolic learner was limited as discussed further in Section V. Additionally, we pre-calculated the search space prior to attempting to find the optimal solution during the learning process.

Once a trained model was obtained for each algorithm, the standard prediction function for each model was used to obtain the classification decisions. The validation set was used to obtain the results we present in this paper where we calculate the difference between the decision reached by each classifier and the ground truth recorded in the validation set.

D. Benchmark Machine Specification

All benchmarks performed in this paper were run on the same machine, running a single learning task at any time. The specification for the machine is as follows:

- *Hardware*: IBM System x3850 X5; 4 Intel Xeon E7-8870 2.40GHz (80 cores total); 576GB DDR3 1333 RAM.
- *Operating System*: Fedora 29 (x86_64); Kernel 5.1.11.
- *Software*: python 3.7.3; keras 2.4.2; tensorflow 1.14.0 (CPU); h2o 3.24.0.5; pandas 0.23.4; scikit-learn 0.19.1; ILASP version 3.4 beta 20/04/2019.

V. Results and Discussion

In this section we present and discuss our experimental results and consider them in light of the policy metrics outlined in subsection II-B.

A. Experimental Results

Figures 3, 4 and 5 show the accuracy and training time results of running each learning algorithm for the Mushroom Edibility, Coalition Asset Loaning and Adult Income data sets. The performance of the symbolic ASG learning algorithm is shown by the solid blue line and the performance of the RF, GLM and FCN are shown by the green, red and magenta lines respectively. For data set percentages that are missing ASG results, this indicates the solution is *unsatisfiable*—i.e. the computed hypothesis space is not sufficiently detailed to learn a satisfiable solution based on the training examples given.[3]

When the ASG-based symbolic learning approach learns a solution, it demonstrates comparable performance to the best statistical learning approach. For example, in Figure 3a for the Mushroom Edibility data set, the ASG achieves 99% accuracy compared to 100% for the RF up to and including 4% of the training data. The ASG fails to learn a solution for larger percentages of training data. This is due to the hypothesis space not capturing all the possible combinations of rules that enable a valid rule set to be identified. *Unsatisfiable* means there are no rules within the hypothesis space that correctly distinguish between the positive and negative training examples. For larger problems, it is necessary to increase the values for the *Max Literals (ML)* and *Rule Length (RL)* hyper-parameters to enable the ASG to compute a sufficiently detailed hypothesis space.

If adequate computational resources are available and the correct ASG hyper-parameters are used, it is possible to compute the full hypothesis space and thus learn an optimal solution. As shown in Table II, increasing the values for the hyper-parameters results in a significant increase in the required computational resource for computing the hypothesis space. Increasing the number of literals from 3 to 4 for 2% of the Mushroom training data results in a $105X$ increase in the amount of memory required and a $1.2X$ increase in the amount of computation time required. Using a greater number of literals, 1% of the Mushroom training data resulted in a $9X$ increase in computation time and 3% exceeded the memory specification of the benchmark machine. Referring to Table I, the Mushroom data set is small when compared to the Adult Income data set. As shown in Figure 5a, the ASG

[3] In our experiments the ASG learning algorithm was configured to find a solution that covered all of the examples. It is possible to configure the ASG learning algorithm to handle noisy examples. In this case, the ASG learner would search for an ASG that optimises a scoring function defined in terms of the hypothesis length and the number of uncovered examples.

failed to learn a solution with small *ML* and *RL* values for all percentages of the Adult Income data set. Table II shows increasing the values of the hyper-parameters when computing the Adult Income hypothesis space for 1% of the training data exceeded the memory specifications of the benchmark machine.

Dataset	Data %	ML	RL	Memory (GB)	Elapsed Time (mins)
Mushroom	1	3	5	3.03	245.07
Mushroom	1	4	5	255.00	2286.52
Mushroom	2	3	5	4.22	377.22
Mushroom	2	4	5	442.68	756.67
Mushroom	3	3	5	5.22	511.12
Mushroom	3	4	5	N/A	N/A
Coalition	1	4	5	0.77	81.10
Coalition	1	4	6	110.03	678.50
Adult	1	3	5	2.11	172.57
Adult	1	4	6	N/A	N/A

TABLE II: The effect of ASG hyper-parameter selection on hypothesis space computation

Given the Coalition data set—a smaller data set with a lower number of features and a lower feature complexity, the ASG performs similarly compared to the best statistical model—the RF, achieving 100% accuracy after being trained on 25% or more of the training examples as can be seen in Figure 4a. The ASG is also able to learn a solution with 100% of the training data. Figure 6 lists the learned ASG at 100% which describes various learned rules, rejecting an asset loan request if $(physical\_constraint = active)$ OR $(weather\_score > 4879$ AND $autonomy\_level \leq 6)$. This is more explainable than state-of-the-art explainability techniques for statistical models as shown in [12]. In Figure 4b, despite the increased learning time for the ASG when compared to the RF, a learning time of one minute may be acceptable for some applications. Note the learning time is the time it takes to learn from the examples once the hypothesis space has been computed.

B. Policy Metrics

As well as evaluating the learning component of a GPM with respect to accuracy and training time, here we discuss the experimental results for each algorithm with respect to the policy metrics outlined in section II-B.

(1) Consistency: Once a GPM has been learned, both symbolic learners and statistical learners will produce consistent results for the same contextual inputs. However, during learning, statistical models often include an element of random initialisation. With neural networks, weights are subject to a random initialisation and in a GLM, coefficients are initialised randomly. This means using the same algorithm to learn over the same training data can lead to different GPMs—with no guarantee of optimality. The ASG symbolic learner will always learn a consistent rule set for the same training data. This can be verified easily due to the explainable nature of the ASG. It can be hard to asses statistical models for consistency due to their inherent black-box nature.

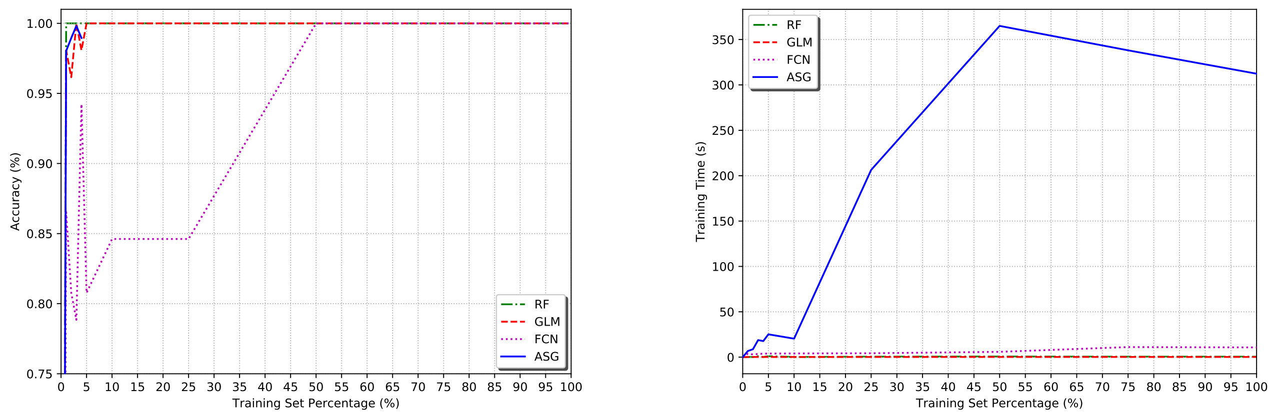
(2) Relevance, *(3) Minimality*, *(4) Completeness* and *(5) Correctness*: These metrics can be easily evaluated for symbolic learners due to their explainable nature. Generated policies can be post-processed if necessary to filter out undesirable system behaviour. For statistical learners, these metrics are hard to evaluate without applying explainability techniques to understand specifically what the model has learned. Ultimately these metrics depend on the training examples given to the learner and could potentially be undermined if the learning algorithm was subject to adversarial training examples.

(6) Enforceablity: With regards to a collaborative environment, time is an important factor as near real-time decision making may be required. Also, in order for GPMs to be deployed, they must outperform human analysts. Both statistical and symbolic learners enable fast inference once a GPM has been learned, although some complex neural networks (e.g. a deep convolutional neural network) may take longer to perform a forward pass. As demonstrated by our experiments, the time required to compute the hypothesis space for the symbolic ASG is currently too large when using complex data sets. Once the hypothesis space has been computed, the learning time for the ASG is larger than the statistical models, although may be acceptable in some applications. Bandwidth is another important factor, as collaborative partners may be required to share a GPM or its generated policies with limited communication infrastructure. With the ASG-based symbolic learner, the GPM is represented as a list of rules that can be easily compressed and shared. For statistical models, it may be necessary to compress the model or use explainability techniques to extract a list of rules so that the model can be shared.

(7) Explainability: As discussed, in terms of explaining what the GPM has learned, the ASG symbolic learner is fully explainable and statistical approaches require further work. As an extension, it may also be necessary to explain *why* certain policies have been learned based on the observed training examples.

VI. Future Work

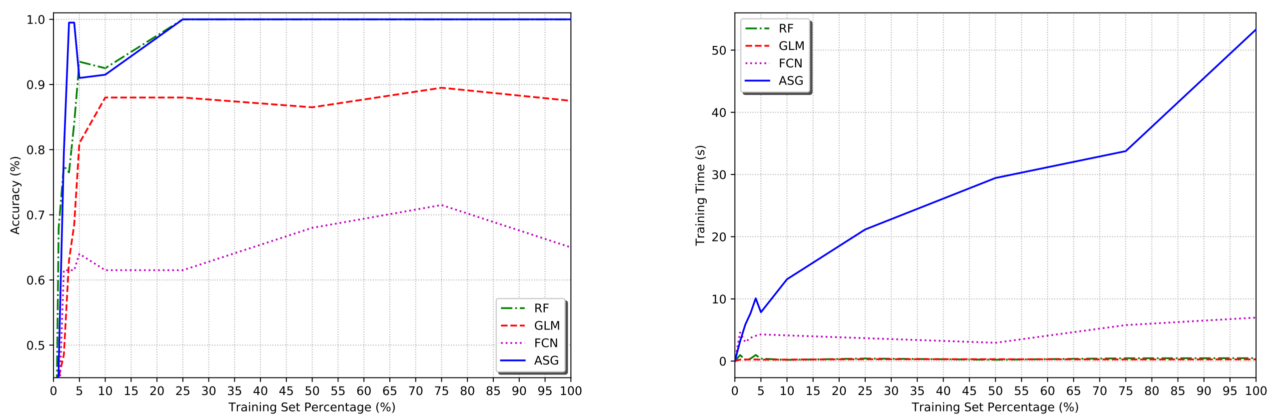
As evidenced in section V, the ASG based symbolic learning approach demonstrates a promising research direction with a view towards enabling GPMs to be deployed in real-world collaborative environments. The ASG demonstrates powerful performance comparable to state-of-the-art, matured and highly optimised statistical learning algorithms when learning over small data sets. Also, the ASG is fully explainable and statistical learning algorithms struggle to match this standard. However, as demonstrated in our experiments, the ASG requires a significant computational resource to compute a detailed hypothesis space required for learning over complex problems or large data sets. Future work should explore methods to parallelise the hypothesis space computation and optimise the ASG algorithm for large, complex problems. Also, the emerging field of integrated statistical and symbolic



(a) Accuracy

(b) Training Time

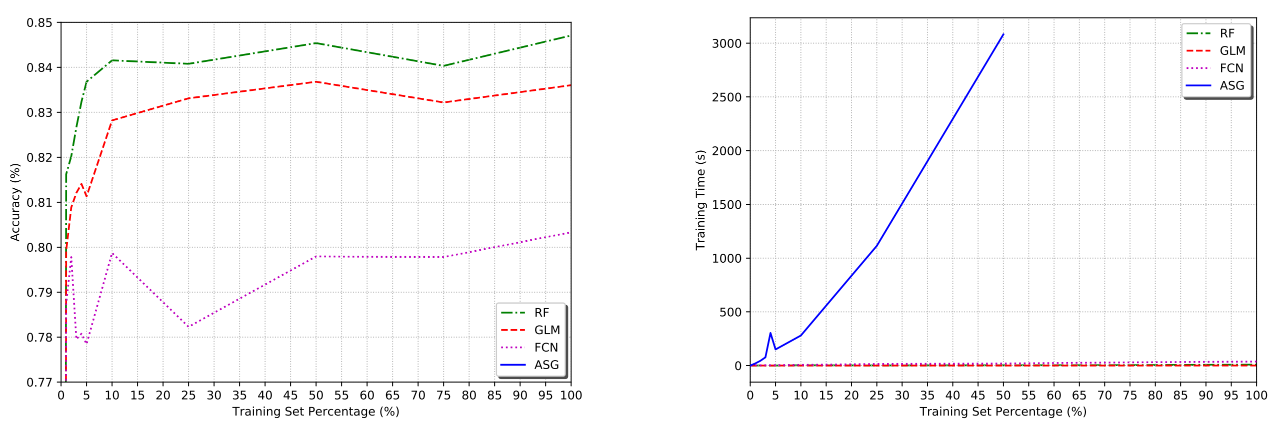
Fig. 3: Mushroom Edibility results for all learning algorithms across all data set percentages



(a) Accuracy

(b) Training Time

Fig. 4: Coalition Asset Loaning results for all learning algorithms across all data set percentages



(a) Accuracy

(b) Training Time

Fig. 5: Adult Income results for all learning algorithms across all data set percentages

learning should be investigated to leverage the learning power and relative efficiency of statistical models on complex data sets alongside the explainable properties of a symbolic learner. Similarly, efforts to make statistical learning approaches more explainable should also be considered.

```
start -> requestor "; " asset_type "; "
    asset_sub_type "; " asset_risk "; "
    asset_available "; " asset_worth "; "
    autonomy_level "; " asset_owner "; " trust
    "; " mission_type "; " mission_environment {
        :- physical_constraint("active").
        :- V1 > 4879, V2 <= 6, weather_score(V1),
            autonomy_level(V2)@13.
}
```

Fig. 6: Learned ASG using 100% of the Coalition Asset Loaning training data

VII. Conclusion

This paper has presented a thorough comparison between statistical and symbolic learning algorithms over a variety of data sets and amounts of training examples, where the performance of each approach has been evaluated with respect to a set of policy metrics. We have demonstrated the ASG-based symbolic learning algorithm is a promising research direction towards enabling deployable GPMs in real-world collaborative environments. The ASG-based approach performs similarly to state-of-the-art statistical learning algorithms on small data sets and is fully explainable, but requires large computational resources to learn a solution on more complex problems. This paper serves as a future benchmark for evaluating the performance of symbolic and statistical learning algorithms with respect to collaborative GPMs.

Acknowledgments

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] S. B. Calo, D. C. Verma, and E. Bertino, "Distributed intelligence: Trends in the management of complex systems," in *Proceedings of the 22Nd ACM on Symposium on Access Control Models and Technologies*, ser. SACMAT '17 Abstracts. New York, NY, USA: ACM, 2017, pp. 1–7. [Online]. Available: http://doi.acm.org/10.1145/3078861.3078881

[2] D. Verma, S. Calo, S. Chakraborty, E. Bertino, C. Williams, J. Tucker, and B. Rivera, "Generative policy model for autonomic management," in *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug 2017, pp. 1–6.

[3] S. Calo, D. Verma, S. Chakraborty, E. Bertino, E. Lupu, and G. Cirincione, "Self-generation of access control policies," in *Proceedings of the 23Nd ACM on Symposium on Access Control Models and Technologies*, ser. SACMAT '18. New York, NY, USA: ACM, 2018, pp. 39–47. [Online]. Available: http://doi.acm.org/10.1145/3205977.3205995

[4] G. White, J. Ingham, M. Law, and A. Russo, "Using an asg based generative policy to model human rules," *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, Jun. 2019.

[5] D. Cunnington, G. White, M. Law, and G. de Mel, "A demonstration of generative policy models in coalition environments," in *Advances in Practical Applications of Survivable Agents and Multi-Agent Systems: The PAAMS Collection*. Springer International Publishing, 2019, pp. 242–245.

[6] S. Calo, I. Manotas, G. de Mel, D. Cunnington, M. Law, D. Verma, A. Russo, and E. Bertino, "AGENP: An ASGrammar-based GENerative policy framework," in *Policy-Based Autonomic Data Governance*. Springer, Sep. 2019, pp. 3–20. [Online]. Available: https://doi.org/10.1007/978-3-030-17277-0_1

[7] M. Law, A. Russo, B. Elisa, B. Krysia, and L. Jorge, "Representing and learning grammars in answer set programming," in *AAAI*, 2019.

[8] G. White, S. Pierson, B. Rivera, M. Touma, P. Sullivan, and D. Braines, "DAIS-ITA scenario," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006. International Society for Optics and Photonics, Apr 2019, p. 110061F. [Online]. Available: https://doi.org/10.1117/12.2520150

[9] G. Brewka, T. Eiter, and M. Truszczynski, "Answer set programming: An introduction to the special issue," *AI Magazine*, vol. 37, no. 3, pp. 5–6, 2016.

[10] M. Law, A. Russo, and K. Broda, "The ILASP system for learning answer set programs," http://ilasp.com/, 2015.

[11] ——, "Inductive learning of answer set programs," in *European Workshop on Logics in Artificial Intelligence*. Springer, 2014, pp. 311–325.

[12] D. Cunnington, M. Law, G. de Mel, I. Manotas, E. Bertino, S. Calo, and D. Verma, "Towards a learning-algorithm agnostic generative policy model for coalitions," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006, 2019. [Online]. Available: https://doi.org/10.1117/12.2520243

[13] N. Puri, P. Gupta, P. Agarwal, S. Verma, and B. Krishnamurthy, "Magix: Model agnostic globally interpretable explanations," *arXiv preprint arXiv:1706.07160*, 2017.

[14] R. Tomsett, D. Braines, D. Harborne, A. Preece, and S. Chakraborty, "Interpretable to whom? a role-based model for analyzing interpretable machine learning systems," *arXiv preprint arXiv:1806.07552*, 2018.

[15] C. Molnar, *Interpretable Machine Learning*. Self Published on GitHub, 2019, https://christophm.github.io/interpretable-ml-book/.

[16] E. Bertino, A. A. Jabal, S. Calo, D. Verma, and C. Williams, "The challenge of access control policies quality," *J. Data and Information Quality*, vol. 10, no. 2, pp. 6:1–6:6, Sep. 2018. [Online]. Available: http://doi.acm.org/10.1145/3209668

[17] J. Dinsmore, *The symbolic and connectionist paradigms: closing the gap*. Psychology Press, 2014.

[18] J. W. Shavlik, R. J. Mooney, and G. G. Towell, "Symbolic and neural learning algorithms: An experimental comparison," *Machine Learning*, vol. 6, no. 2, pp. 111–143, Mar 1991. [Online]. Available: https://doi.org/10.1007/BF00114160

[19] P. Foggia, R. Genna, and M. Vento, "Symbolic vs. connectionist learning: an experimental comparison in a structured domain," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 2, pp. 176–195, March 2001.

[20] G. H. Lincoff, *The Audubon society field guide to North American mushrooms*. Knopf, 1989.