Published in the Proceedings of AAAI-15, AAAI Press, Austin TX, January 2015

Measuring Plan Diversity: Pathologies in Existing Approaches and A New Plan Distance Metric

Robert P. Goldman and **Ugur Kuter**
SIFT, LLC
Minneapolis, MN USA
{rpgoldman,ukuter}@sift.net

Abstract

In this paper we present a plan-plan distance metric based on Kolmogorov (Algorithmic) complexity. Generating diverse sets of plans is useful for tasks such as probing user preferences and reasoning about vulnerability to cyber attacks. Generating diverse plans, and comparing different diverse planning approaches requires a domain-independent, theoretically motivated definition of the diversity distance between plans. Previously proposed diversity measures are not theoretically motivated, and can provide inconsistent results on the same plans.

We define the diversity of plans in terms of how surprising one plan is given another or, its inverse, the conditional information in one plan given another. Kolmogorov complexity provides a domain independent theory of conditional information. While Kolmogorov complexity is not computable, a related metric, Normalized Compression Distance (NCD), provides a well-behaved approximation. In this paper we introduce NCD as an alternative diversity metric, and analyze its performance empirically, in comparison with previous diversity measures, showing strengths and weaknesses of each. We also examine the use of different compressors in NCD. We show how NCD can be used to select a training set for HTN learning, giving an example of the utility of diversity metrics. We conclude with suggestions for future work on improving, extending, and applying it to serve new applications.

Introduction

In many applications the ability to generate multiple plans that are *interestingly* different would be useful. Boddy *et al.* (2005) propose to use diverse plan generation to give defenders insight into attackers' plans in cyber-security domains. Nguyen, *et al.* (2012a) and Myers & Lee (1999) propose to use diverse plans to probe users' preferences. Goldman *et al.* (2012) have used AI planners to identify ways in which a user's plan is vulnerable to uncontrolled actions and events. Identifying diverse ways that a plan can go wrong would help users make their plans robust to disturbances. Other applications include program analysis (Kuter et al. 2014), and attack surface generation in computer networks (Thayer et al. 2013). These applications would benefit from a domain-independent, theoretically-justified, and informative definition of plan-plan diversity.

There are a number of approaches to generating diverse sets of plans for a problem (Myers and Lee 1999; Coman and Muñoz-Avila 2011; Bryce 2014; Nguyen et al. 2012a; Roberts, Howe, and Ray 2014). Unfortunately, the only commonly-used way of measuring plan diversity, action distance (Srivastava et al. 2007; Nguyen et al. 2012a), lacks a strong theoretical basis and can give poor results in many cases. Other proposed domain-independent measures have similar problems, and while Coman & Munõz-Avila (2011) suggest using domain-specific measures, there is no commonly accepted technique for identifying appropriate domain-specific measures.

Our objective is to provide a theoretically grounded, domain-independent, and quantitative way to assess plan-plan distances and identify interesting diversity. Our contributions are as follows:

- We analyze previous diversity measures, and based on this analysis, we present an intuitive account of plan diversity in terms of conditional surprise (conditional information), and coding length.
- The challenge of conditional information as a measure of plan diversity, is that obvious approaches require domain dependent information. We meet this challenge by employing Kolmogorov (Algorithmic) complexity (Li and Vitányi 2008). Unfortunately, algorithmic complexity is not computable, so we adopt *compression distance* (Li et al. 2004) as an approximation.
- In our experiments, compression distance was successful in providing diversity measures in planning domains with correlated fluents, unordered subplans, lifted information, and could quickly identify conditional information in plans.
- We used NCD to select training sets for a Hierarchical Task Network (HTN) learning algorithm. In these experiments, plans sets found diverse by compression distance enabled more effective learning than AD.
- We also present experimental results showing potential weaknesses of compression distance as a plan-plan distance metric. In particular, our results show that compression distance can over-focus on adjacent relations in the

plans (e.g., subsequences of actions), and degrade when plans have independent causal chains or partial orders. Similarly, our results show that disrupting adjacency relations can hide similarities from compression distance metrics.

- We discuss the challenges of aggregating pairwise diversity measures to measure the diversity of sets of plans. We illustrate the difficulties of existing measures and present a simple alternative.

We conclude with a discussion of further research that is enabled by this new plan distance metric.

Background

The most influential work on plan diversity measures is that of Srivastava, *et al.* (2007), refined in Nguyen, *et al.* (2012b). They propose three different distance measures for comparing plans, and for measuring the diversity of a set of plans: Action distance (AD), Causal link distance (CLD), and state distance (SD). AD and CLD both project the plan, an *ordered* set of actions, down to an unordered set, and then compute a set-difference based distance between these sets. So, if $A(p)$ is the set of actions in plan p, and $C(p)$ the set of causal links, the action distance, AD, and causal link distance, δ_C are defined as follows:

$$\mathrm{AD}(p, p') = 1 - \frac{|A(p) \cap A(p')|}{|A(p) \cup A(p')|} \quad (1)$$

$$\mathrm{CLD}(p, p') = 1 - \frac{|C(p) \cap C(p')|}{|C(p) \cup C(p')|} \quad (2)$$

Causal links are triples, $\langle a, p, a' \rangle$, where a is an action producing p as an effect, and a' is an action consuming p as a precondition. SD differs from the action and causal link distances in that it is a measure over state *sequences*, rather than state sets. For plans p, p' with lengths $l(p') \leq l(p)$, The definition is as follows:

$$\mathrm{SD}(p, p') = \frac{1}{l(p)} \left[\sum_{i=1}^{l(p')} \Delta(s_i, s'_i) + l(p) - l(p') \right] \quad (3)$$

where $\Delta(s, s') = 1 - \frac{|s \cap s'|}{|s \cup s'|}$, a measure of the difference in the fluents holding in the two states. Nguyen, *et al.* actually provide *two* alternative definitions for state distance, which differ slightly in how they handle a difference between $l(p)$ and $l(p')$; for details see their paper. They provide two planning algorithms: GP-CSP, which can generate plans that attempt to maximize either AD, CLD, or SD; and a more efficient method, LPG-d, which only uses action distance. Using GP-CSP, they provide experimental results on several domains to argue that action distance is the hardest to maximize. In general, later work in diverse planning confines itself to using action distance; we don't know of other work that uses CLD or SD.

The measures defined by Nguyen *et al.* all have some problems. While they have the advantage of being domain-independent, no strong motivation is given, aside from their ready computability. Another problem is that they are not plan distance metrics, in the mathematical sense. A distance function, D, must satisfy three properties to be a metric (Searcóid 2007):

$$D(x, y) = 0 \text{ iff } x = y \text{ (identity)} \quad (4)$$

$$D(x, y) = D(y, x) \text{ (symmetry)} \quad (5)$$

$$D(x, y) \leq D(x, z) + D(z, y) \text{ (triangle inequality)} \quad (6)$$

Neither AD nor CLD satisfy the identity property, since different plans can give rise to the same action and causal link sets through reordering (note that AD and CLD *are* metrics over action and link sets, just not over plans). Similarly, in some problems it is possible for two different action sequences to give rise to the same state sequences. Action distance does not take into account information in the lifted representation of a domain. To AD the plans p_1 = *drive(t1, a, b), drive(t1, b, c)* and p_2 = *drive(t2, a, b), drive(t2, b, c)* look every bit as different as p_1 and p_3 = *fly(a1, a, g), drive(t3, g, c).* Causal link and state distances also fail to take such generalizations into account. Further, in the case of state distance, correlated state fluents – fluents that change in lockstep – can artificially drive up the state difference between plans.

In the following section, we describe a new plan-plan distance measure based on Kolmogorov complexity theory. Then, we provide empirical demonstrations of the pathologies in the previous measures and provide evidence that our proposed metric provides a more consistent and coherent view of plan-plan diversity distance.

Normalized Compression Distance for Plan Diversity

We measure plan diversity distance using *normalized compression distance (NCD),* which approximates *normalized information distance*, a measure of conditional information based on Kolmogorov complexity. Kolmogorov complexity provides a *domain-independent* measure of complexity (description length) which does not rely on having a probability model for the objects of the domain. This is a critical advantage for plan diversity, since we have no means of attributing a probability distribution to sets of plans.

At the time of this work, our primary interest in plan diversity was to use a set of diverse plans to seed learning of a plan library. This led us to view plan diversity in terms of how surprising a plan, p_{n+1} is, given that one has seen plans $p_1 \ldots p_n$, $S(p_{n+1}|p_1 \ldots p_n)$. We considered two ways of measuring this notion of surprise: one was to use *edit distance* (Levenshtein distance), defined as the number of edit operations required to transform one string into another. This had the advantage of being able to incorporate edit operations based on learning (e.g., swapping sub-sequences, swapping parameter values on operators), but there was no obvious way to assign a cost to the different operations. A closely related idea was to use description length: p_{n+1} would be considered as more diverse if its coding length, given $p_1 \ldots p_n$ was longer, and one could take the results of learning into account by adding to a codebook. Unfortunately, to compute an optimal coding, one needs a probabil-

ity distribution over the messages, and we had no principled means of defining a probability distribution over the plans.

Kolmogorov Complexity

Kolmogorov complexity aims to address problems like those outlined above: to provide a domain-independent measure of information content/complexity. Our discussion of Kolmogorov complexity, unless otherwise specified, follows the text by Li and Vitányi (2008). The key insight is that the information content, $K(x)$ of a string, x can be defined in terms of the shortest Turing machine program that can produce that string.[1] Strings containing significant regularities can be described by shorter programs than those without such regularities. *Conditional* Kolmogorov complexity, $K(x|y)$ may also be defined, in this case in terms of the size of the shortest program that will compute x given y as input. A key theoretical result in Kolmogorov complexity establishes that these measures of complexity are unique up to an additive constant, not arbitrarily variable according to encoding schemes.

Based on conditional Kolmogorov complexity, we may define *information distance* between two strings. An information distance measure must satisfy the three requirements for a metric (4,5,6), as well as the density conditions $\sum_{y \neq x} 2^{-D(x,y)} \leq 1; \sum_{x \neq y} 2^{-D(x,y)} \leq 1$ (the density conditions reject degenerate measures). Finally, an information distance metric is required to be upper semicomputable (approximated from above by a computable function). It is a theorem that the measure $E_1(x,y) = \max K(x|y), K(y|x)$ is a minimal information distance in the sense that for all information distances, $D(x,y), E_1(x,y) \leq D(x,y) + O(1)$.

A final wrinkle is that we are not concerned with an *absolute* measure of information distance, but a relative one. That is, we want to know the distance between a pair of plans *relative to the size of the plans*, so that we don't get anomalous effects such as pairs of longer plans always being treated as more diverse than pairs of shorter plans. The *normalized information distance* is defined as follows: $e(x,y) = \frac{\max K(x|y), K(y|x)}{\max K(x), K(y)}$. $e(x,y)$ ranges from 0 to 1, and it is approximately a metric (there are small errors in the identity and triangle inequalities).

We argue that the best *domain-independent* definition of the distance between plans p_1 and p_2 is the normalized information distance, $e(p_1, p_2)$. This definition brings together our intuitions about coding theory and edit distance, since E_1 is equal (up to a logarithmic additive term) to $E_0(x,y)$, the length of the shortest program that transforms x into y and y into x. Unfortunately, $e(x,y)$, like most interesting questions about Turing machines, cannot be computed. However, *normalized compression distance (NCD)* provides a practical means of approximating $e(x,y)$.

Normalized Compression Distance

The minimality and semicomputability of $e(x,y)$ suggests the practical approach of approximating information distance using an information measure that is computable. In several applications, *normalized compression distance (NCD)* has been used, giving good results particularly in clustering and classification (Li et al. 2004; Li and Vitányi 2008). NCD is an approximation – an upper bound – of $e(x,y)$, where a compressor is used to approximate the program length for a string. NCD is computed as follows:

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min C(x), C(y)}{\max C(x), C(y)} \tag{7}$$

where $C(x)$ is the length of a file containing the string x, after compression, and xy is the concatenation of x and y. Li, *et al.* (2004) report successful experiments in clustering of DNA sequences and language families. Li & Vitányi (2008) additionally report on clustering file types, and cite a number of successes in the KDD community.

The compressors use domain-independent, adaptive models to extract information from their input, and exploit that information in the compression process (Sayood 2012). The models differ in how they take context into account: using adaptive codebooks, predictive matching, etc. We have experimented with four compressors: gzip, bzip2, paq8, and ppmz. gzip and bzip2 are both popular compressors that provide less compression, but compress and decompress very quickly. gzip uses a combination of Huffman coding and an adaptive codebook based on the LZ77 algorithm (Gailly and Adler 2014; Sayood 2012). It is interesting to note that there is a much simpler complexity notion, Lempel-Ziv complexity, based on the adaptive codebook (Lempel and Ziv 1976, Cited in (Sayood 2012)). bzip2 uses a combination of Burrows-Wheeler block sorting, and Huffman coding (Seward 2014). paq8 uses neural networks and context-mixing to achieve high compression at the cost of long runtime (Mahoney 2014). ppmz uses Prediction with Partial Map, the ppm algorithm, with many enhancements (Bloom 2014). Since we are attempting to bound the optimal encoding, one should use the best available compressor: we generally use paq8, but we explore the use of different compressors in our experiments.

Empirical Evaluation

In this section we use test cases to explore the strengths and weaknesses of NCD as a measure of plan diversity. First we have shown that AD, CLD, and SD all get artificial diversity measures on domains with correlated fluents, while NCD can better identify shared structure. We have shown that how different compressors compare with each other in this case, exhibiting similar qualitative plan-plan distance metric behavior. Then, we have provided evidence that although variation in ordering with correlated fluents is more visible and more appropriately recognized by the existing AD, CLD, and SD measures, it is a challenge for NCD. In contrast, our third test demonstrated that information about the first order structure in planning domains is visible to NCD, but not to AD, CLD, and SD.

We compare the behavior of a number of alternative compressors when used inside NCD. We find that they provide qualitatively similar behavior in many cases, but our experience (as illustrated here), shows paq8 achieving best compression.

[1] For technical reasons, prefix-free Turing machines are used.

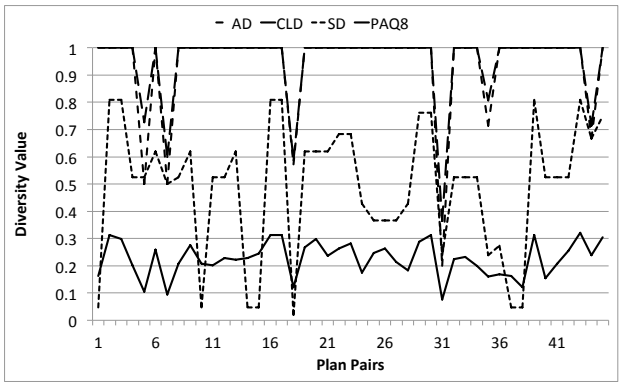

Figure 1: Comparisons of AD, CLD, SD, and NCD with pac8 compressor in problems with correlated fluents.

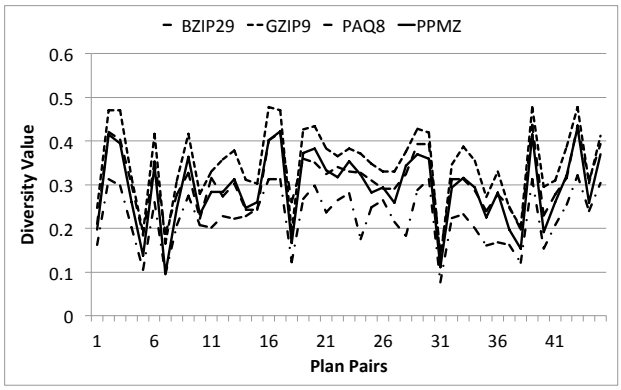

Figure 2: Comparisons of different compressors used by NCD in problems with correlated fluents. Above, NCD was run with gzip, bzip2, ppmz, and pac8.

We have also conducted an experiment in which we show that variations in naming and plan representations could pose challenges to NCD. These can cause NCD to behave poorly, but can be addressed by pre-processing. More problematic are cases involving plans with causally independent actions, and parameter shuffling. We illustrate the challenges, and suggest directions for future work.

Finally, we conducted an experiment to illustrate the utility of our diversity measure. We used diversity measures to choose training sets for the HTN-Maker learning system (Hogg, Muñoz-Avila, and Kuter 2008). We found that HTN Maker learns better with a training set that NCD labels as diverse versus one that it labels less diverse, while the value of AD cannot similarly predict learning performance.

In our experiments, we used the typed STRIPS dialect of PDDL (Planning Domain Description Language) (McDermott 1998) to represent planning domains, planning problems and plans. We have implemented the AD, CLD, and SD diversity measures, as described in (Nguyen et al. 2012a), and NCD as described in (Li et al. 2004) in Common Lisp. For NCD, we have used four different publicly-available compressors: gzip (Gailly and Adler 2014), bzip2 (Seward 2014), ppmz (Bloom 2014), and paq8 (Mahoney 2014). Since paq8 gave us the best compression, whenever we report only a single result for NCD, we report for paq8. We use all of the compressors at the highest compression effort setting, with the exception of paq8, for which we use -8 instead of -9 because -9 often crashed for us. The following sections describe our experimental scenarios and results. Experimental data (including plans and domain and problem definitions) will be available at `www.sift.net/aaai14-diversity/`.

Correlated Fluents. We have written an abstract planning domain for this experiment, where there are two agents one of which mirrors the other's actions. The problems involve an agent reaching a particular goal location in a grid, and we compare plans for problems with different initial positions of the two ninjas with different goals.

Figure 1 shows experimental results where we evaluated AD, CLD, SD, and NCD on a suite of randomly-generated problems in this planning domain. Figure 1 shows the results. As expected, AD, CLD, and SD all give inflated estimates of diversity. For some pairs (see the downward spiking entries), these measures can detect some commonality, but often they cannot make finer distinctions below the level of "these plans are completely different." NCD, on the other hand, is sensitive to the qualitative features of the plans and therefore, its distance measures are much lower.

Comparing Compressors. The different compressors use different kinds of internal models while compressing, and make different tradeoffs between run-time and compression. Standard compressors such as gzip and bzip are more concerned about compression and decompression times, than compressors such as ppmz and pac8, that try for maximal compression. In order to investigate how NCD's behavior changes with different compressors, we compare NCD on the "Correlated fluents" problem. We compared NCD's performance with compressors gzip, bzip2, ppmz, and pac8. Figure 2 shows the results. As one would expect, ppmz and pac8 are better at extracting commonalities, since they spend more effort in compression. An interesting feature, though, is that the shapes of the different curves are broadly similar, despite the difference in techniques.

We conjectured that the more in-depth compressors would do better when plans were jumbled. To test this, we took a single, 20-step plan for the driverlog domain, and split it into subsequences of length 2, 4, 5, and 10. Then we permuted these subsequences (we did a maximum of 100 permutations, choosing randomly when we could not exhaustively explore the permutations), and averaged the pairwise comparisons. As one would expect, ppmz and paq8 are better at identifying the underlying similarities. Note that this is an artificial test, since the "plans" here are not well-formed. But this simulates plans with causally-independent subplans.

Action Orderings. We hypothesized that because most file compressors exploit adjacency relations in their input, NCD would find it difficult to recognize certain similarities

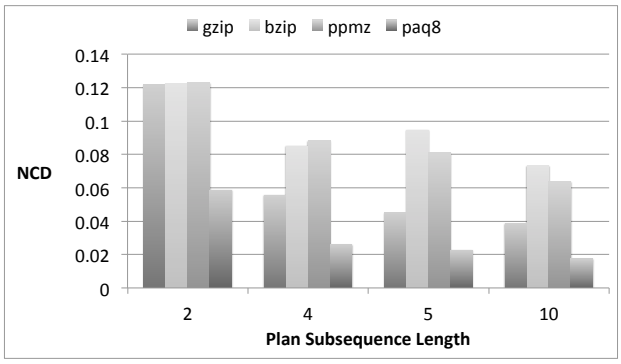

Figure 3: Comparing different compressors in NCD with shuffled subsequences of a source plan.

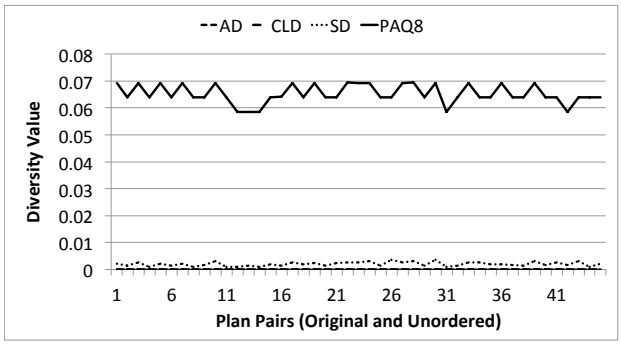

Figure 4: Comparisons of AD, CLD, SD, and NCD with the pac8 compressor in problems with correlated fluents and unordered subplans.

in plans where action-action relations operate over long distances. This can happen when there are plans that involve independent causal chains that can be interleaved.

To investigate this hypothesis, we studied how sensitive the metrics are to action orderings in plans in the same abstract planning domain as above. We generated random planning problems with correlated fluent, whose solutions consists of unordered subplans. Figure 4 shows the results: AD and CD correctly consider all the plan pairs to be identical. SD also considers the plans to be very similar but not identical; this is due to the small variations in the states that appear in different trajectories induced by different action orderings.

As also shown in Figure 4, NCD, using pac8 as its compressor, could not recognize properly the action-action similarities operate over long distances. Our metric still considers the plan-plan relations are similar, but it over-focuses on adjacent action-action similarities and therefore, identifies the plans as more diverse than did AD, CLD, and SD.

On one hand, arguably, this experiment shows a limitation on the information-theoretic way to view plan diversity. Consider the most extreme case of independence, a plan of n steps, in which all $n!$ orderings are permissible. In this case, the conditional information of one plan given another, even of one plan given a causal model, is substantially, because to

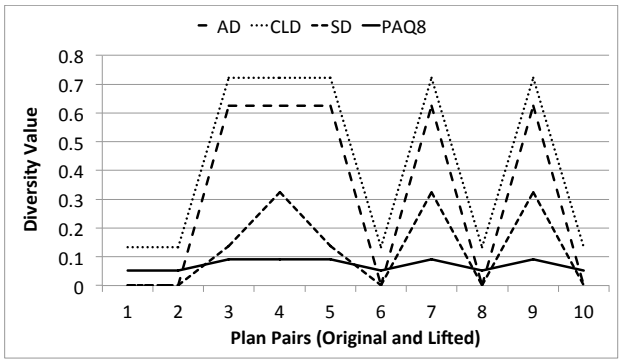

Figure 5: Comparisons of AD, CLD, SD, and NCD with pac8 in problems with lifted plan sets.

specify one of the plans given another, we must provide the full ordering.

On the other hand, since the plans differed only in subplan orderings, AD cannot distinguish between plan pairs at all. Since the subplans are causally independent, even the additional information in the causal links does not suffice to distinguish them, either.

The low values for SD are somewhat surprising, since fluents change their values at different points in the plan. This is a spurious effect caused by the fact that there are a large number of static predicates in the domain. A better version of SD would remove them from the computation, and give higher normalized results.

Lifted Information. Existing plan-plan distance metrics are very sensitive to parameter values because they work on grounded representations. For example, in the Logistics planning domain (Veloso 1992), if we compare two delivery plans, each following the same path, but using different trucks, AD and CLD will give very high diversity values, even if qualitatively the two plans are very similar.

To illustrate this sensitivity in the existing metrics, we have conducted an experiment in Logistics where we have randomly generated a suite of problems using the planning domain and problem descriptions from the 2000 International Planning Competition. First, we have run AD, CLD, SD, and NCD (with pac8) in pairs of plans for these problems. Then, we have lifted some of the objects in those plans by using existential quantification to remove some of the arguments from the operators.

Figure 5 compares AD, CLD, SD, and NCD measures of diversity over the same pairs in their original and lifted forms. The results show how sensitive AD and CLD is to grounding. In the original problems, AD and CLD returned varying high and low distance measures, classifying the plans as diverse; however, Logistics plans are all qualitatively similar since they consist of combinations of LOAD-DRIVE-UNLOAD or LOAD-FLY-UNLOAD subplans. SD was able to recognize some of the similarities, but it still had a significant variance over plan pairs. NCD, on the other hand, recognized the similarities among the plans.

Effect of Plan Encodings. One concern is that NCD might simply be exploiting the textual conventions of planning domains written to be human-understandable. So, for example, object are typically in a style like `TRUCK1`, `TRUCK2`, etc. Good naming conventions simply make obvious the type structure, and so give NCD a shortcut to a valid feature of the problem. However, it is obviously a problem if naming is critical to the performance of the NCD metric: that would mean that the it would be likely to behave poorly, e.g., when handling problems generated by programs, for programs. So we investigate whether our NCD metric behaves correctly, using only available information, when naming is purposely obscured.

We conjectured that, if names in the plans are not meaningful, we could apply a preprocessor, before the compressor, that uses the domain description to build a codebook. This codebook would expose type information to the compressor in a principled way, instead of relying on the natural language textual structure. To investigate this hypothesis, we conducted a small renaming experiment. We took a domain and problem and randomly renamed all the operators and all the types. Then we renamed all the objects as $\langle typename \rangle \langle index \rangle$. We compared the performance of NCD on plans in their original, meaningfully named, form, and on plans that had been renamed in a way that was meaningful, but not natural language.

We generated a pair of plans whose normalized AD, CLD, and SD were all 1: we did this by taking one plan from a blocks world problem and one from a logistic problem. NCD measured the difference between the plans as being 0.535. When we applied our renaming, the new NCD measure was only 0.481, showing that the renaming did not change NCD's behavior significantly.

As another extreme, we compared that with the case of a pair of logistics plans from our lifted comparisons, described above. For these two plans, CLD was 0.921 and AD was 0.857, because these metrics are very sensitive to the grounding in the plans. SD was only 0.459, because it was able to recognize similarities in the way the plans induce similar state trajectories (the moving agents are different, but the packages go to the same places). We measured NCD as 0.124, which was capturing correctly the similarities between the lifted plans. After random renaming, the NCD was 0.112.

Simply garbling the names of the operators and actions in a typed PDDL domain, then, does not impede the use of NCD to measure diversity, since the damaging effects of the garbling can be undone both efficiently, and in a principled way. This shows that NCD is not simply recognizing regularities in natural language names.

Parameter Shuffling. Although renaming may not damage NCD, the compressors exploit adjacency relations, so disrupting these relations while maintaining the same causality can hide similarities from NCD. To investigate this challenge, we did another experiment in which we shuffled the parameters of operators: e.g., if there are two operators, o an o' that differ only in the ordering of their parameters, plans where o' is substituted for o will have diversity scores that are, intuitively speaking, too high. The results were in terms of the diversity values each measure generated: AD = 0.956, CLD = 1.0, SD = 0.0, and NCD = 0.259.

As expected, this kind of modification confuses AD and CLD because the actions are part of the causal links. SD correctly detects that the plans have the same structure. NCD, on the other hand, was fooled by the fact that different orders of the parameters in the plans break the syntactic similarities between them.

Use Case: Diversity for Learning. We have also performed experiments to investigate how diversity, and in particular correct and meaningful plan-plan distance measures, may help in other applications and problems. For these experiments, we chose Hierarchical Task Network (HTN) learning, where learning takes as a set of plans as input and generates a library of HTNs for planning.

Our experimental hypothesis was that the more diverse the input training plans to HTN learning, the better the learning coverage will be (i.e., the more test planning problems an HTN planner would be able to solve). We used HTN-Maker, a modern HTN learning system (Hogg, Muñoz-Avila, and Kuter 2008; Hogg, Kuter, and Muñoz-Avila 2009; 2010), and the SHOP2 HTN planner (Nau et al. 2003).

We wrote an abstract planning domain for these experiments, in order to carefully probe the effects of diversity in the input. Adapted from the Logistics domain (Veloso 1992), our domain involves taking an object from one location to another on a network. In this network, each edge between locations has a particular color; in our experiments, we had five distinct colors for labeling the edges. There are five actions in the domain, each of which moves the object over a specifically-colored edge.

We generated 30 problems in this domain with varying colorings and different bounds on the lengths of the plans. We compared NCD with AD in terms of their diversity values over each plan set by giving those sets as training examples to HTN-Maker. We also have randomly generated 10 distinct test problems. In runs where AD returned 1.0 values for the plan-plan distance, labeling them most diverse, SHOP2 with HTN-Maker's learned HTNs was only able solve 20% of the test problems, because it was labeling plans incorrectly as being diverse. By comparison, NCD predicted the quality of learning: given a low-diversity training set, SHOP2 was able to solve 20% of the test planning problems. When we gave HTN Maker a set of training plans that NCD classified as being of higher diversity, SHOP2 solved 80% of the test planning problems using the learned library.

Aggregation

As Roberts, *et al.* (2014) point out, there are issues in aggregating plan diversity measures over sets of plans. Perhaps surprisingly, obvious aggregation methods tried by previous researchers give bad results, primarily because of conflating search and pruning heuristics with measures. We review these here, illustrate the problems and propose a simple solution: mean and variance.

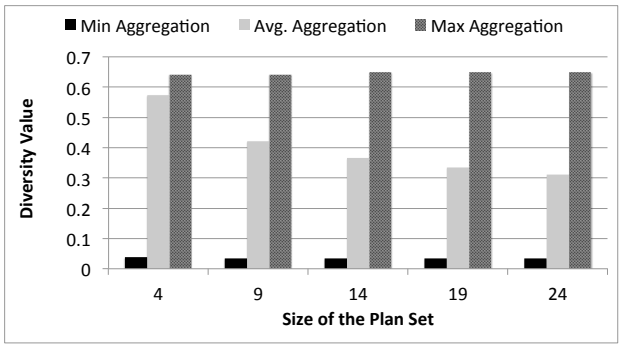

Figure 6: Comparisons of different aggregation techniques for diversity metrics over plan sets. Min aggregation is used by (Nguyen et al. 2012a). Max aggregation corresponds to set diameter. We investigate how average aggregation compares against these previously-known techniques.

Srivastava, *et al.* (2007) generalize their distance measures from pairs to sets of plans by minimizing the distance measure from each plan to the other plans in the set: for a set, S, where the pairwise diversity measure is δ, the distance metric would be:

$$D(S) = \min_{p \in S} \left(\min_{p' \in S | p' \neq p} \delta(p, p') \right) \quad (8)$$

This has the obvious problem that as a set of diverse plans grows, its measured diversity can plummet abruptly: adding a single, less diverse plan to the set causes the whole set's diversity to drop sharply. This behavior is reasonable, however, when you consider how they were using this measure: not as a measuring tool, but as a pruning tool. It's reasonable for the search to prune candidates causing diversity to drop.

Metric theory (Searcóid 2007) suggests a different measure: set diameter $\mathrm{dia}(A) = \sup\{\delta(x, y) | x, y \in A\}$. This avoids the falling diversity problem of (8), above. Unfortunately, it still doesn't provide a good aggregate measure of the set, since any elements of less than or equal diversity are essentially invisible.

Once one views this as a question of characterizing the entire set, rather than the extent of the set, the use of mean and variance (and possibly other measures such as median) is the obvious choice. In the following, we illustrate the problems of previous aggregation measures, and compare them to the use of mean and variance.

Figure 6 shows the weaknesses of previous aggregation measures. It was generated by taking four maximally-diverse plans, and then gradually introducing 5 new plans at each datapoint that are all strictly less diverse (because they contain parts of the original plans). We used NCD with pac8 as the underlying diversity measure. Our results show that neither max-aggregation (diameter), nor min-aggregation (the measure suggested by Srivastava, *et al.*) is sensitive to the internal structure of the set of plans.

Related Work

Some much earlier work by Myers and Lee (1999) provided an alternative definition of diversity based on Euclidean distance. They define diversity as a function of the *dispersion* and *proximity* of plan sets. Dispersion measures plan distances in a Euclidean feature space. Proximity measures how close a plan is to the boundaries of the plan set. This metric is not easy to compare to NCD, since it requires domain-dependent feature definitions.

Roberts, *et al.*. (2014) present a number of methods for computing diverse plan sets. They describe their work as proposing new diversity metrics, but those metrics measure diversity of *planners*, rather than of plans themselves.

Conclusions and Future Work

In this paper we have described a Kolmogorov complexity based plan-plan distance measure for plan diversity. We showed that it provides a sound, domain-independent theoretical basis for conditional information. Since Kolmogorov complexity is not computable, we also presented normalized compression distance as an approximation. We conducted a number of studies, which show strengths and weaknesses of the approach. A number of the issues with previous diversity measures are handled nicely by NCD, although there are still some vulnerabilities to syntactic aspects of the plan representation, as displayed in parameter shuffling. Our experiment with HTN Maker provides an example of the utility of a plan diversity measure.

NCD opens a number of areas for future research. One is to overcome expressive limitations: for example, NCD does not obviously adapt to plans with costs; doing so would require a principled way of fusing cost distance with information distance. Our NCD scheme also does not handle temporal plans. It would also be interesting to consider a plan representation that exposed the causal structure of the actions, for example treating an action as a collection of causal links, giving access to the causal dynamics. One question that we have just begun to probe has to do with how sensitive NCD will be to naming: two of our experiments addressed this, but more work needs to be done here. We expect to extend this work to incorporate learned knowledge in the compression distance. For example, if HTN methods are learned, they can be used to pre-compress the plan before feeding it to a conventional compressor.

Fox, *et al.* (2006) propose to measure *stability* under replanning and plan repair by (roughly speaking) measuring action distance between initial and repaired (replanned) plans. However, the notion of stability from control theory is a relationship between the input and output of the system. Where we do not have a domain-specific measurement, NCD should give us a way to measure distance between both the input (the problems) and the output (the plans).

Acknowledgments Thanks to the anonymous reviewers for many helpful suggestions. Thanks to Peter Keller for help building the PPMZ compressor. This paper was supported by DARPA and the U.S. Air Force under contract number FA8650-11-C-7191. Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Bloom, C. 2014. Ppmz website. www.cbloom.com/src/ppmz.html. Date is retrieval date.

Boddy, M. S.; Gohde, J.; Haigh, T.; and Harp, S. A. 2005. Course of action generation for cyber security using classical planning. In Biundo, S.; Myers, K. L.; and Rajan, K., eds., *International Conference on Automated Planning and Scheduling*, 12–21. AAAI.

Bryce, D. 2014. Landmark-based plan distance measures for diverse planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*.

Coman, A., and Muñoz-Avila, H. 2011. Generating diverse plans using quantitative and qualitative plan distance metrics. In *Proceedings AAAI*.

Fox, M.; Gerevini, A.; Long, D.; and Serina, I. 2006. Plan stability: Replanning versus plan repair. In Long, D.; Smith, S. F.; Borrajo, D.; and McCluskey, L., eds., *ICAPS*, 212–221. AAAI.

Gailly, J., and Adler, M. 2014. Gzip website. www.gzip.org. Date is retrieval date.

Goldman, R. P.; Kuter, U.; and Schneider, A. 2012. Using classical planners for plan verification and counterexample generation. In *Proceedings of AAAI Workshop on Problem Solving Using Classical Planning*.

Hogg, C.; Kuter, U.; and Muñoz-Avila, H. 2009. Learning hierarchical task networks for nondeterministic planning domains. In *IJCAI-09*.

Hogg, C.; Kuter, U.; and Muñoz-Avila, H. 2010. Learning methods to generate good plans: Integrating htn learning and reinforcement learning. In *AAAI-10*.

Hogg, C.; Muñoz-Avila, H.; and Kuter, U. 2008. HTN-MAKER: Learning HTNs with minimal additional knowledge engineering required. In *AAAI-08*.

Kuter, U.; Burstein, M.; Benton, J.; Bryce, D.; Thayer, J.; and McCoy, S. 2014. HACKAR: helpful advice for code knowledge and attack resilience. Under review.

Lempel, A., and Ziv, J. 1976. On the complexity of finite sequences. *IEEE Transactions on Information Theory* 22(1):75–81.

Li, M., and Vitányi, P. 2008. *An introduction to Kolmogorov complexity and its applications*. Springer, third edition.

Li, M.; Chen, X.; Li, X.; Ma, B.; and Vitányi, P. M. B. 2004. The similarity metric. *IEEE Transactions on Information Theory* 50(12):3250–3264.

Mahoney, M. 2014. The PAQ data compression programs. cs.fit.edu/~mmahoney/compression/paq.html. Date is retrieval date.

McDermott, D. 1998. PDDL, the planning domain definition language. Technical report, Yale Center for Computational Vision and Control.

Myers, K. L., and Lee, T. J. 1999. Generating qualitatively different plans through metatheoretic biases. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 570–576. Menlo Park, Cal.: AAAI/MIT Press.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, W.; Wu, D.; and Yaman, F. 2003. SHOP2: An HTN planning system. *JAIR* 20:379–404.

Nguyen, T. A.; Do, M. B.; Gerevini, A.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012a. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence* 190:1–31.

Nguyen, T. A.; Do, M. B.; Gerevini, A.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012b. Generating diverse plans to handle unknown and partially known user preferences. In *Artificial Intelligence* (2012a) 1–31.

Roberts, M.; Howe, A.; and Ray, I. 2014. Evaluating diversity in classical planning. In *Proceedings ICAPS*.

Sayood, K. 2012. *Introduction to data compression*. Morgan Kaufmann, fourth edition.

Searcóid, M. O. 2007. *Metric Spaces*. SUMS. Springer.

Seward, J. 2014. Bzip2 website. www.bzip.org. Date is copyright.

Srivastava, B.; Nguyen, T. A.; Gerevini, A.; Kambhampati, S.; Do, M. B.; and Serina, I. 2007. Domain independent approaches for finding diverse plans. In Veloso, M. M., ed., *Proceedings of the International Joint Conference on Artificial Intelligence*.

Thayer, J.; Burstein, M.; Goldman, R. P.; Kuter, U.; Robertson, P.; and Laddaga, R. 2013. Comparing strategic and tactical responses to cyber threats. In *SASO Workshop on Adaptive Host and Network Security AHANS*.

Veloso, M. M. 1992. Learning by analogical reasoning in general problem solving. PhD thesis CMU-CS-92-174, School of Computer Science, Carnegie Mellon University.