Technical Annex

Neil R. Bramley

November 11, 2019

This document provides additional details about the models we use in this project, for reference. Specifically, it details the *Context-Free Grammar* we use to model normative inference in compositional spaces, and the *Instance Driven Generation* procedure we propose for modelling data-driven hypothesis generation and our approach to *Scene Similarity Analysis.*

"Top-down" Context-Free Generation

One solution to the problem of learning within an infinite hypothesis space is to sample hypotheses by composing them stochastically from an underlying grammar of sufficient expressivity. In our pilot work, we modelled compositional sampling using a *probabilistic context free grammar* (or PCFG) Ginsburg (1966) that can produce any rule that can be expressed with first-order logic and lambda abstraction. Note that lambda abstraction provides a simple general formalism for binding entities to variables Church (1932). If a broad enough sample of hypotheses are generated in this way and scored based on their ability to capture available data, it is possible to approximate Bayesian inference Goodman, Tenenbaum, Feldman, and Griffiths (2008). In the current context, binding sets of objects to different variables means that, in the limit, PCFG samples will make all possible logical assertions about the features of, and relations between, subsets of objects within a scene that are necessary and sufficient for a scene to be "rule following". For instance, grammatical statements include things like "blue objects may not be touching", or "three objects must be facing the same way", or "all objects must be green or horizontal but no more than one can be large", and so on.

Table 1: λ-abstraction Based Probabilistic Grammar.

Start	$S \rightarrow$	$\exists(\lambda x_i:\ A,\mathcal{X})$	$\forall(\lambda x_i:\ A,\mathcal{X})$	$N_I(\lambda x_i:\ A, J,\mathcal{X})$		
Bind additional	$A \rightarrow$	B	S			
Expand	$B \rightarrow$	C	$H(B,B)$	$\neg(B)$		
Function	$C \rightarrow$	$=(x_i, D1)$	$I(x_i, D2)$	$=(x_i, x_j, E1)^{\mathbf{a}}$	$I(x_i, x_j, E2)^{\mathbf{a}}$	$\Gamma(x_i, x_j, E3)^{\mathbf{a}}$
Feature/value	$D1 \rightarrow$	value$^{\mathbf{b}}$,	feature			
(numeric only)	$D2 \rightarrow$	value$^{\mathbf{b}}$,	feature			
Feature	$E1 \rightarrow$	feature				
(numeric only)	$E2 \rightarrow$	feature				
(relational)	$E3 \rightarrow$	feature				
Boolean	$H \rightarrow$	$\wedge$	$\vee$	...		
Inequality	$I \rightarrow$	$\leq$	$\geq$	$>$	$<$	
Number	$J \rightarrow$	$n \in \mathcal{Z}_5$				

Note: Each line maps a capital letter (column 2) to several possible expansions (right hand columns). Context-sensitive aspects of the grammar: [a]Bound variable(s) sampled uniformly without replacement from set; expressions requiring multiple variables censored if only one is bound. [b] The value is always sampled uniformly from the support of the feature selected in D.

To draw a sample hypothesis using the PCFG in Table 1, one starts with a string containing a single *non-terminal symbol* (here, S). Non-terminal symbols in the string are then iteratively replaced using expansions drawn from the options in the requisite row of Table 1. Each expansion replaces non-terminal symbols with a mixture of other non-terminal symbols and terminal fragments of first order logic, until no non-terminal symbols remain. The expansions are so designed that the final string is guaranteed to be a valid grammatical expression. Figure 1a gives two example samples from the PCFG detailed in Table 1.[1] Because some of the expansions involve branching (e.g., $B \rightarrow H(B,B)$), the resultant string can become arbitrarily long and complex, involving multiple Boolean functions and complex relationships between bound variables.

Followin Piantadosi, Tenenbaum, and Goodman (2016), we assume the latent space of possible concepts in our task to be those expressible in first-order logic combined with lambda abstraction and full knowledge of the objects' relevant features. However, our choice of expansion rules is one of many that can produce this class of expressions. Furthermore, the probabilities for each expansion can be set separately. This all means that the PCFG framework is a flexible way of encoding a learner's inductive bias, or prior, over a compositional

[1]Note that this grammar is not strictly context free because the bound variables (x_1, x_2, etc.) are shared across contexts (e.g. x_1 is evoked twice in both expressions generated in Figure 2a).

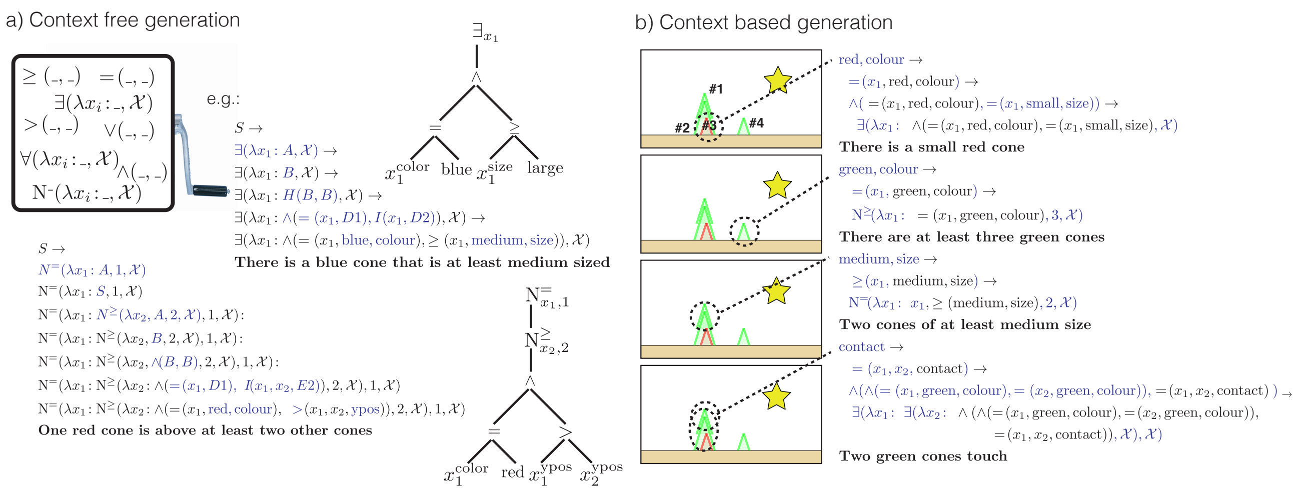
a) Context free generation
$\geq(\_,\_)$ $=(\_,\_)$ $\exists(\lambda x_i : \_, \mathcal{X})$ $>(\_,\_)$ $\vee(\_,\_)$ $\forall(\lambda x_i : \_, \mathcal{X})$ $\wedge(\_,\_)$ $\mathrm{N}^{\text{-}}(\lambda x_i : \_, \mathcal{X})$
e.g.:
$S \rightarrow$
$\exists(\lambda x_1 : A, \mathcal{X}) \rightarrow$
$\exists(\lambda x_1 : B, \mathcal{X}) \rightarrow$
$\exists(\lambda x_1 : H(B,B), \mathcal{X}) \rightarrow$
$\exists(\lambda x_1 : \wedge(=(x_1, D1), I(x_1, D2)), \mathcal{X}) \rightarrow$
$\exists(\lambda x_1 : \wedge(=(x_1, \text{blue}, \text{colour}), \geq(x_1, \text{medium}, \text{size})), \mathcal{X})$
There is a blue cone that is at least medium sized
$\exists_{x_1}$ $\wedge$ $=$ $\geq$ x_1^{color} blue x_1^{size} large
$S \rightarrow$
$N^{=}(\lambda x_1 : A, 1, \mathcal{X})$
$\mathrm{N}^{=}(\lambda x_1 : S, 1, \mathcal{X})$
$\mathrm{N}^{=}(\lambda x_1 : N^{\geq}(\lambda x_2, A, 2, \mathcal{X}), 1, \mathcal{X})$:
$\mathrm{N}^{=}(\lambda x_1 : \mathrm{N}^{\geq}(\lambda x_2, B, 2, \mathcal{X}), 1, \mathcal{X})$:
$\mathrm{N}^{=}(\lambda x_1 : \mathrm{N}^{\geq}(\lambda x_2, \wedge(B,B), 2, \mathcal{X}), 1, \mathcal{X})$:
$\mathrm{N}^{=}(\lambda x_1 : \mathrm{N}^{\geq}(\lambda x_2 : \wedge(=(x_1, D1), I(x_1, x_2, E2)), 2, \mathcal{X}), 1, \mathcal{X})$
$\mathrm{N}^{=}(\lambda x_1 : \mathrm{N}^{\geq}(\lambda x_2 : \wedge(=(x_1, \text{red}, \text{colour}), >(x_1, x_2, \text{ypos})), 2, \mathcal{X}), 1, \mathcal{X})$
One red cone is above at least two other cones
$\mathrm{N}^{=}_{x_1,1}$ $\mathrm{N}^{\geq}_{x_2,2}$ $\wedge$ $=$ $>$ x_1^{color} red x_1^{ypos} x_2^{ypos}
b) Context based generation
#1 #2 #3 #4
red, colour $\rightarrow$
$=(x_1, \text{red}, \text{colour}) \rightarrow$
$\wedge(=(x_1, \text{red}, \text{colour}), =(x_1, \text{small}, \text{size})) \rightarrow$
$\exists(\lambda x_1 : \wedge(=(x_1, \text{red}, \text{colour}), =(x_1, \text{small}, \text{size}), \mathcal{X})$
There is a small red cone
green, colour $\rightarrow$
$=(x_1, \text{green}, \text{colour}) \rightarrow$
$\mathrm{N}^{\geq}(\lambda x_1 : =(x_1, \text{green}, \text{colour}), 3, \mathcal{X})$
There are at least three green cones
medium, size $\rightarrow$
$\geq(x_1, \text{medium}, \text{size}) \rightarrow$
$\mathrm{N}^{=}(\lambda x_1 : x_1, \geq(\text{medium}, \text{size}), 2, \mathcal{X})$
Two cones of at least medium size
contact $\rightarrow$
$=(x_1, x_2, \text{contact}) \rightarrow$
$\wedge(\wedge(=(x_1, \text{green}, \text{colour}), =(x_2, \text{green}, \text{colour})), =(x_1, x_2, \text{contact})) \rightarrow$
$\exists(\lambda x_1 : \exists(\lambda x_2 : \wedge(\wedge(=(x_1, \text{green}, \text{colour}), =(x_2, \text{green}, \text{colour})), =(x_1, x_2, \text{contact})), \mathcal{X}), \mathcal{X})$
Two green cones touch

Figure 1: a) Example generation of hypotheses using PCFG in Table 1. b) Examples of IDG hypothesis generation based on an observation of a scene that follows the rule. New additions on each line are marked in blue.

hypothesis space with different choices of expansion and different weights shaping the frequency with which a particular expression will be produced. Provided the expansions that involve branching do not occur with too high a probability, shorter expressions will naturally be favoured, organically capturing Ockham's razor.

"Bottom-up" Instance-Driven Generation

Our Instance-Driven Generation (IDG) proposal (Bramley, Rothe, Tenenbaum, Xu, & Gureckis, 2018) is related to the PCFG idea but with one major difference. Rather than generating guesses entirely context-free, before checking them against the data, we propose that people generate guesses *inspired* by an encountered positive example Minton et al. (1989). Concretely, we propose that learners start by observing the features of objects in a rule-following scene and use these to back out a true logical statement about the scene in a stochastic but truth preserving way. In this way the learner does not generate uniformly from all possible logical statements, but directly from the restricted space of statements true of the current observation. Figure 1b motivates this approach. Here a learner begins their hypothesis generation with an observation of a scene that follows the hidden rule. To generate hypotheses as candidates for the hidden rule, we assume the learner uses the following procedure:

1. **Observe.** With uniform probability, either:
   (a) Sample a object from the observation, then sample one of its features — e.g., #2:[2] "medium, size" or {#3}: "red, colour".
   (b) Sample two objects uniformly without replacement from the observation, and samples any shared or pairwise feature — e.g., {#1,#2}: "size", or "contact"
2. **Functionise.** Bind a variable for each sampled object in Step 1 and sample a true (in)equality statement relating the variable(s) and feature:
   (a) For a statement involving an unordered feature there is only one possibility — e.g, {#3}: "$=(x_1, \text{red}, \text{color})$", or for {#1,#2}: "$=(x_1, x_2, \text{color})$"
   (b) For a single object and an ordered feature this could also be a nonstrict inequality ($\geq$ or $\leq$). We assume a learner only samples an inequality if it expands the number of objects picked out from the scene relative to an equality — e.g., in Figure 1b, there is also a large object {#1} so either $\geq(x_1, \text{medium}, \text{size})$ or $=(x_1, \text{medium}, \text{size})$ might be selected.
   (c) For two objects and an ordered feature, either strict or non-strict inequalities could be sampled if the objects differ on the sampled feature, equivalently either equality or non-strict inequality could be selected if the objects do not differ on that dimension — e.g., {#1,#2} $>(x_1, x_2, \text{size})$, or {#3,#4} $\geq(x_1, x_2, \text{size})$.
3. **Extend.** With probability $\frac{1}{2}$ go to Step 4, otherwise sample uniformly one of the following expansions and repeat. For statements with two bound variables Step 3 is performed for x_1, then again for x_2:
   (a) **Conjunction.** A object is sampled from the subset picked out by the statement thus far and one of its features sampled — e.g., {#1} $\wedge(=(x_1, \text{green}, \text{color}), \geq(x_1, \text{medium}, \text{size}))$. Again, inequalities

[2]Numbers prepended with # refer to the labels on the objects in the example observation in Figure 1b.

are sampled only if they increase the true set size relative to equality — e.g., "$\wedge(\leq(x_1, 3, \text{xposition}), \geq(x_1, \text{medium}, \text{size}))$", which picks out more objects than "$\wedge(=(x_1, 3, \text{xposition}), \geq(x_1, \text{medium}, \text{size}))$".

   (b) **Disjunction.** An additional feature-value pair is selected uniformly from *either* unselected values of the current feature, *or* from a different feature — e.g., $\vee(=(x_1, \text{color}, \text{red}), =(x_1, \text{color}, \text{blue}))$ or $\vee(=(x_1, \text{color}, \text{blue}), \geq(x_1, \text{size}, 2))$. This step is skipped if the statement is already true of all the objects in the scene.

4. **Quantify.** Given the contained statement, select true quantifier(s):
   (a) For statements involving a single bound variable (i.e., those inspired by a single object in Step 1) the possible quantifiers simply depend on the number of the objects in the scene for which the statement holds. The quantifier is selected uniformly from the existential $\exists(\_)$ or numerical "exactly $N^{=}(\_, J)$", "at least $N^{\geq}(\_, J)$", or "at most $N^{\leq}(\_, J)$" and J is set to match number of objects for which the statement is true. If it it is true for all objects, the universal quantifier $\forall(\_)$ may also be selected.
   (b) Statements involving two bound variables in lambda calculus have two nested quantifier statements each selected as in (a). The inner statement quantifying x_2 is selected first based on truth value of the expression while taking x_1 to refer to the object observed in '1.'. The truth of the selected inner quantified statement is then assessed for all objects to select the outer quantifier — e.g., {#3,#4} "$\wedge(=(x_2, \text{green}, \text{color}), \leq(x_1, x_2, \text{size}))$" might become "$\forall(\lambda x_1: \exists(\lambda x_2: \wedge(=(x_2, \text{green}, \text{color}), \leq(x_1, x_2, \text{size})), \mathcal{X}), \mathcal{X})$". The inner quantifier $\exists$ is selected because three of the four objects are green {#1, #2, #4}, and the outer statement is selected because all objects are less than or equal in size to a green object.

One way to think of the IDG procedure is as an inversion of a PCFG. As illustrated by the blue text in the examples in Figure 1, while the PCFG starts at the outside and works in, the IDG starts from the central content and works outward out to a quantified statement, ensuring at each step that this final statement is true of the scene. Note also that the specific algorithm we provide here is only an illustrative example that we expect to refine on the basis of our experiments.

Scene Similarity Analyses

The scenes we explore in Bramley et al. (2018) involve differing numbers of objects that also vary across five basic dimensions: [colour, size, x-position, y-position, rotation]. Two are continuous (x-position, y-position), one is ordinal (size), one is categorical (colour) and one is cyclical (orientation).

To establish the overall similarity between two scenes we need to map the objects in a given scene to the objects in another scene (for example between the scenes in Figure 2a and b) and establish a reasonable cost for the differences between objects across dimensions. We also need a procedure for cases where there are objects in one scene that have no analogue in the other. We approach the calculation of similarity via the principle of minimum edit distance (Levenshtein, 1966). This means summing up the elementary operations required to convert scene a into scene b or visa versa. We assume objects can be adjusted in one dimension at a time (i.e. moving them on the x axis, rotating them, or changing their colour, and so on.

Before focusing on how to map the objects between the scenes we must decide how to measure the adjustment distance for a particular object in scene a to its supposed analogue in scene b. As a simple way to combine the edit costs across dimensions we first z-score each dimension, such that the average distance between any two values across all objects and all scenes and dimensions is 1. We then take the L1-norm (or city block distance) as the cost for converting an object in scene (a) to an object in scene (b), or visa versa.[3] Note this is sensitive the size of the adjustment, penalising larger changes in position, orientation or size more severely than smaller changes, while changes in colour are all considered equally large since colour is taken as categorical. Note also that for orientation differences we also always assume the shortest distance around the circle.

If scene a has an object that do not exist in scene b we assume a default adjustment penalty equal to the average divergence between two objects across all comparisons. We do the the same for any object that exists in a but not b.

Calculating the overall similarity between two scenes involves solving a mapping problem of identifying which objects in scene (a) are "the same" as those in scene b. We resolve this charitably, by searching exhaustively for the mapping of objects in scene a to scene b that minimises the edit distance. Having selected the mapping, computed the edit distance including any costs for additional or removed objects, we divide by the number shared cones, so as to avoid penalising larger more complex scenes. Figure

[3]We appreciate that this is a simplifying assumption, however believe it is a reasonable one for incommensurable dimensions (i.e. colour and size do not meaningfully trade off).

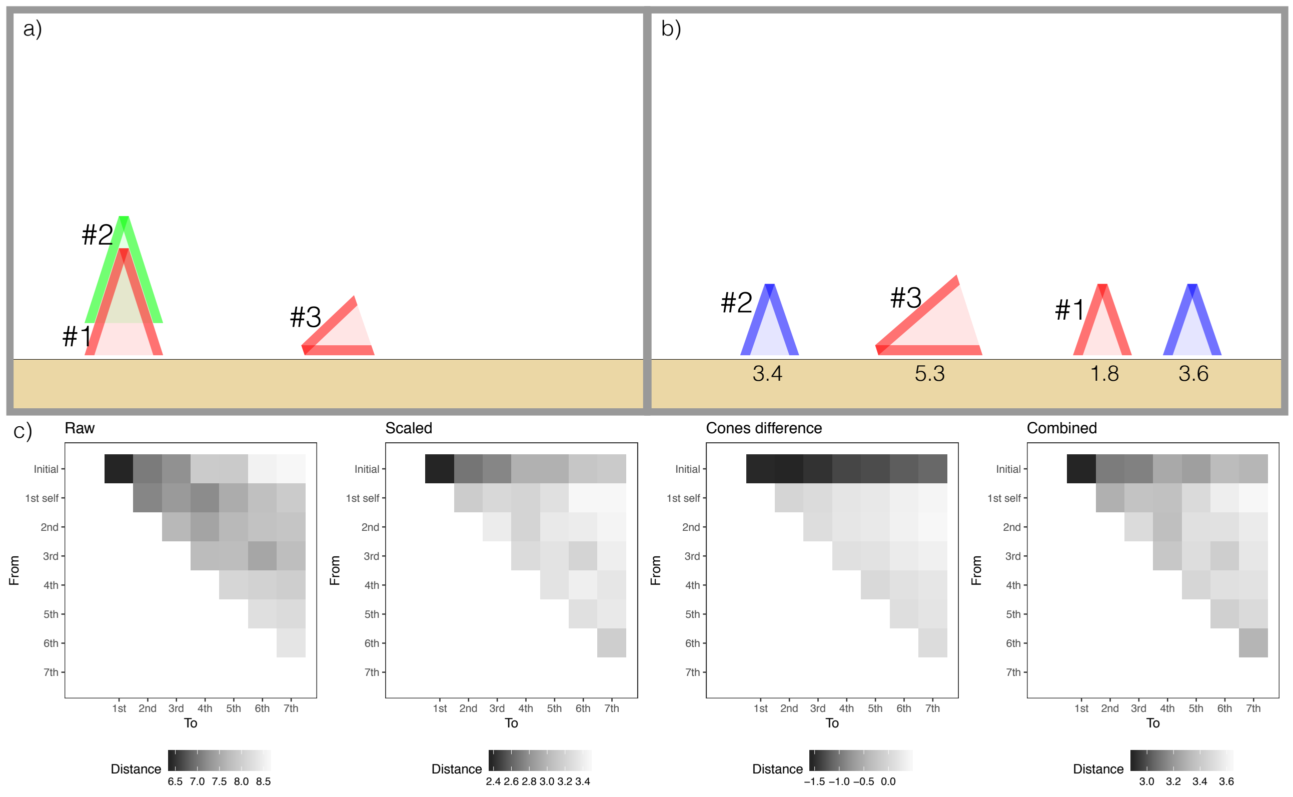

Figure 2: Calculating similiarity. a & b) Two example scenes. Objects indices link the most similar set of objects in b to those in a. Numbers below indicate the edit distance for each object (i.e. the sum of scaled dimension adjustments). The unique object in b is then penalised with the mean edit distance of 3.6. c) Visualises the components of the overall similarity measures computed across all data in Bramley et al. (2018). "Raw" simply sums the differences across the shared objects. Scaled divides this by the total number of objects. Cones difference computes the average difference in the number of cones in the scene and combined adjusts scaled by cones difference to give the measure we adopt.

References

Bramley, N. R., Rothe, A., Tenenbaum, J. B., Xu, F., & Gureckis, T. M. (2018). Grounding compositional hypothesis generation in specific instances. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society.* Austin, TX: Cognitive Science Society.

Church, A. (1932). A set of postulates for the foundation of logic. *Annals of mathematics*, 346–366.

Ginsburg, S. (1966). *The mathematical theory of context free languages.* McGraw-Hill Book Company.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, *32*(1), 108–154.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, pp. 707–710).

Minton, S., Carbonell, J. G., Knoblock, C. A., Kuokka, D. R., Etzioni, O., & Gil, Y. (1989). Explanation-based learning: A problem solving perspective. *Artificial Intelligence*, *40*(1-3), 63–118.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, *123*(4), 392.