Knowledge transfer in a probabilistic Language Of Thought

Samuel J. Cheyette, Steven T. Piantadosi
{scheyett, spiantad} @ ur.rochester.edu
Department of Brain and Cognitive Sciences
University of Rochester, Rochester NY, 14627 USA

Abstract

In many domains, people are able to transfer abstract knowledge about objects, events, or contexts that are superficially dissimilar, enabling striking new insights and inferences. We provide evidence that this ability is naturally explained as the addition of new *primitive* elements to a compositional mental representation, such as that in the probabilistic Language Of Thought (LOT). We conducted a transfer-learning experiment in which participants learned about two sequences, one after the other. We show that participants' ability to learn the second sequence is affected the first sequence they saw. We test two probabilistic models to evaluate alternative theories of how algorithmic knowledge is transferred from the first to second sequence: one model rationally updates the prior probability of the primitive operations in the LOT based on what was used in the first sequence; the other stores previously likely hypotheses as new primitives. Both models perform better than baselines in explaining behavior, with the human subjects appearing to transfer entire hypotheses when they can, and otherwise updating the prior on primitives.

Keywords: Knowledge transfer; Concepts; Language Of Thought; One-shot learning

Introduction

One of the most remarkable capabilities of human cognition is the ability to rapidly create algorithms that are applicable to a new situation. For instance, an adult can quickly pick up a new card game, absorbing the rules and intuiting the strategy. Yet even the simplest card game is complex: it requires knowledge of basics like moves and turns; and it requires complex reasoning abilities, such as general-purpose strategic maneuvers in games. It seems more generally that humans' capacity to infer a lot from sparse data must be undergirded by a flexible array of useful concepts about many domains developed over a lifetime. For example, knowing about strategy in Texas Hold 'Em makes it possible to quickly pick up many other types of poker without reverting to a novice level, because requisite concepts across types of poker share similarities – betting, bluffing, winning hands. Yet this still leaves open the question: what are the representations and computations that make such effective transfer of abstract knowledge in this and myriad other domains possible?

Part of humans' adeptness in learning about new domains quickly may lie in their ability to map old conceptual structures to new ones, allowing them to infer abstract knowledge. This relational reasoning ability has often been characterized as "analogical" in nature (Markman, 1997), and many theories of analogical inference have been proposed on this basis (e.g. Gick & Holyoak, 1980; Gentner, 1983; Holyoak & Thagard, 1989; Hummel & Holyoak, 1997). Gentner's 1983 theory of "structure-mapping", formalized later as the "Structure Mapping Engine" (SME) (Falkenhainer et al., 1989), is an influential framework for describing analogical inference. On this account, situations or facts are given descriptions in predicate logic, the components of which are either objects, relations, or attributes. The goal of a learner when presented with two situations is to make a mapping between these components by finding structural correspondences, and then inferring facts about one situation from the mapping to the other.

A commonality among SME and other theories of analogical transfer is their assumption of static knowledge representations. But structure only captures a limited subset of human knowledge. Other kinds of knowledge, such as learned processes or algorithms are untouched by these theories. In the poker example above, the algorithm of shuffling or bluffing may be transferred whole cloth to a new kind of poker. These abilities may be borrowed and incorporated into the algorithms that reason strategically. This kind of reuse would be much more like a programming language library—a location from which pieces of algorithms can be copied and reused—than just a recognition of a correspondence of pieces. Indeed, in the same way that SME allows for powerful new inferences based on structure, transfer of algorithmic pieces could be part of the answer to how children eventually acquire algorithmically sophisticated representations: learners who can transfer algorithmic pieces need not construct entirely new representations each time they encounter a new domain.

Here, we experimentally and computationally test transfer of algorithmic components of representations by modeling concept learning as program induction over compositional functions, a system often called a "Language Of Thought" (Fodor, 1975). Under the LOT, a learner's job is to induce simple generative programs from primitive functions that match their observations of the world. In essence, this model treats learning as programming: there are a small set of "built in" operations that must be composed correctly in order to express richer algorithmic knowledge. This family of models has successfully been applied to explain human behavior in many rule-learning domains (Piantadosi & Jacobs, 2016), including kinship and taxonomies (Kemp et al., 2008; Katz et al., 2008; Mollica & Piantadosi, 2015), number (Piantadosi et al., 2012), causality, (Goodman et al., 2011), and words (Siskind, 1996; Piantadosi et al., 2008), among others.

Unlike structure mapping theories, LOT models are able to account for concept learning without requiring a significant amount of pre-developed knowledge. On the other hand, LOT models do not provide an account of humans' ability to transfer abstract knowledge between already-learned concepts. In general, it is an open question how LOT models can adapt their inductive biases and primitive representations through experience.

One possibility is that primitives are weighted in their prior according to their past utility as in the "Rational Rules" model

(Goodman et al., 2008). On this account, the prior is computed integrating out the production probabilities, allowing for a reduction in the penalty for repeated use of the same production rule. Among other things, this model has been used to explain selective attention effects, the finding that people tend to focus on as few features as possible to explain an observation.

Another possible way of explaining knowledge transfer in a LOT model is that upon learning a useful program, people store that program as a primitive for later re-use. This approach seems potentially more powerful than only updating priors over primitives themselves, as it could provide a basis for building increasingly complex, hierarchical conceptual structure. Indeed, Dechter et al. (2013) demonstrated how program recombination and re-use can facilitate and improve learning in the domains of both arithmetic and Boolean logic, using program induction over combinatory logic expressions. Others have explored models of sub-program re-use in Probabilistic Context Free Grammars, such as adaptor grammars (Johnson et al., 2006) and fragment grammars (O'Donnell et al., 2011). However, it has yet to be determined empirically if any of these models can explain human transfer of knowledge.

We ran a sequence-learning experiment to test human knowledge transfer, training people on one sequence and then testing them on a transfer sequence. We manipulated the congruity of the sequence pairs, corresponding to the abstract similarity of the training and transfer sequences. The results from our experiment suggest that having seen a congruous sequence in the past has a significant beneficial effect on accuracy. We modeled participants' learning curves in a probability-matching model and three probabilistic LOT models: a Rational Rules-type model that updates the prior of production rules in previously useful concepts; a model that adds previously useful concepts in full to its set of production rules; and a baseline model LOT model that does not update between training and transfer sequences. We compared the fit of each model to human data from our experiment. We found that the LOT model that re-uses high probability hypotheses from training provides the best fit to the data in the congruous condition, and the Rational Rules model provides the best fit in the incongruous condition. These findings suggest that learners transfer entire concepts when they can, and otherwise prefer previously used primitives.

Experiment

We used a one-shot transfer learning paradigm in which participants were shown pairs of sequences which could either have come from similar LOT programs or not. To determine effects of knowledge transfer, we tested whether participants' overall accuracy on the the second sequence varied as a function of the first.

Participants 360 participants were recruited from Amazon Mechanical Turk, whose ages varied from 20 to 67. They were paid 50 cents to complete the experiment, which took roughly 3-5 minutes.

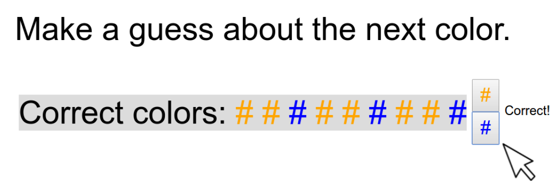

Figure 1: Example of display participants saw in the experiment.

Method

Design The task involved a repeated binary choice, in which participants had to pick between two colored symbols (orange and blue) 15 times in learning both the training and transfer sequence. There were a total of 12 stimuli of which 6 were designated training sequences and 6 were designated transfer sequences. The manipulation was a full-factorial between-subjects design with respect to the stimuli, so every possible combination of these sequences was tested, with only two shown to any given subject. An example of the display shown to participants is given in Figure 1. Note that every participant in both conditions saw the exact same training sequences — the differences in stimuli between conditions were only in the transfer sequence (the second of the two).

Stimuli The particular stimuli we chose were partly designed to allow for differing levels of compression in encoding in the LOT model. Some pairs of stimuli involve very simple repetitions, e.g. $((A^2B)^N)$ and $((A^3B)^N)$ [1], which in our model are expressible in short hypotheses. Other patterns are not as efficiently compressible in our model, such as the repetition of $((AB)^2B)$. But, more importantly, they were designed such that the congruous pairs had abstract similarity, such that learning the first might help with learning the second. For instance, a congruous counterpart of the sequence $(A^2B^3)^N$ is the sequence $(A^2B^4)^N$, since a simple change to the description of one would result in the other. Every sequence in the first set had a congruous counterpart in the second set. The full set of stimuli is shown in Table 1, with congruous pairs adjacent.

Procedure Participants each saw two sequences, one after the other. Starting with no information about each sequence and ending with the entire sequence displayed on the screen, participants chose the symbol they thought was most likely given the previous values of the sequence they could see. After each guess, feedback appeared on the screen as to whether or not they were correct, and the correct symbol was placed at the end of the sequence on the screen. After participants completed the first sequence, it was erased from the screen, and they then completed the same task for the second sequence, starting from the beginning.

[1] $((A^2B)^N)$ and $((A^3B)^N)$ are, in full, N repetitions of AAB and AAAB, respectively. In general X^N means the symbol or sequence X repeated N times.

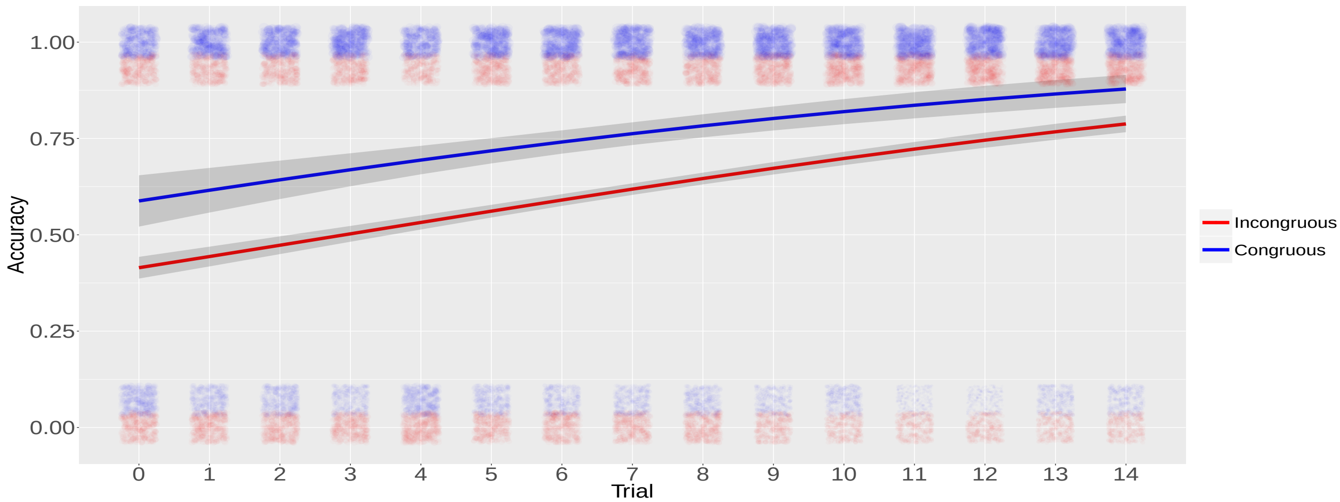

Figure 2: This plot shows participants' accuracy in the transfer sequence over all 15 trials in the congruous (blue) and incongruous (red) conditions. The dots on top and bottom represent participants responding correctly or incorrectly at that trial, respectively. The decreasing transparency of squares of dots on top shows increasing numbers of correct responses, and the fact that the blue squares on top are less transparent than the red squares on top represents better learning in the congruous than incongruous condition. The two curves are the best-fit logistic regression predictions for the congruous and incongruous conditions.

Training	Transfer
$(A^2B^3)^N$	$(A^2B^4)^N$
$B^5(AB)^N$	$A^4(BA)^N$
$(A^2B)^N$	$(A^3B)^N$
$(ABAB^2)^N$	$(BABA^2)^N$
B^6A^N	A^4B^N
$BA^2BA^3BA^4 \ldots BA^{i-1}BA^i$	$ABAB^2AB^3 \ldots AB^{i-1}AB^i$

Table 1: The full set of stimuli in our experiment is comprised of the first 15 symbols of each of these sequences. The congruous pairs are adjacent, and any non-adjacent pair is considered incongruous. The notation X^N used here can be understood as N repetitions of sequence X. The bottom-most congruous pair is not as easily expressible in this way, but can be understood as incrementally increasing runs of one symbol interspersed by the other.

Participants were instructed to make their best guess about the next value of each sequence, even if they were unsure. They were told nothing about whether the two sequences were related, only that they both involved strings of colored symbols.

Results

Our primary concern in analysis is to determine both the effect of sequence step and the effect of congruity on learning the transfer sequence. To determine both in a single analysis, we ran a logistic regression with both factors as fixed effects as well as random subject and sequence intercepts.

The results of this analysis revealed both a main effect of sequence step ($\beta = 0.12, z = 15.8, p < 0.001$) and congruity ($\beta = 0.70, z = 4.69, p < 0.001$). The interaction between congruity and sequence step was not significant ($z = -0.12$). The fits from this analysis are shown in Figure 2, with the curves representing the best-fit regression lines for the congruous (blue) and incongruous (red) conditions.

Collapsing over all sequences and sequence steps in the transfer sequence, and just considering the average correct response given condition, those in congruous condition responded correctly more often ($M = 0.75$) than those in the incongruous condition ($M = 0.61$). The lack of interaction between congruity and sequence step implies that there is a lingering but constant beneficial learning effect in the congruous condition compared to the incongruous condition, but that the speed of learning in the two conditions is roughly the same.

Model

The general modeling framework we used is a probabilistic Language Of Thought. In this approach there are a set of primitive, typed, and compositional operations, analogous to the statements that define programming languages (e.g. for Python *'if'*, *'elif'*, *'while'*, *'True'*, etc... would be considered primitive operations). The set of operations defines the "grammar", and the allowed rules for composing them are the rules for a Probabilistic Context Free Grammar (PCFG). The list of all possible compositions of production rules defines the entire hypothesis space. Since the number of possible hypotheses produced in our grammar is infinite, we use Metropolis-Hastings, a Markov Chain Monte Carlo (MCMC) sampling method, to provide a finite approximation to the entire space.

Each hypothesis H can be assigned a probability for any observed data D, which is computed via Bayesian inference: $P(H|D) \propto P(D|H)P(H)$. The likelihood, $P(D|H)$ is determined by how well the output of the hypothesis matches the data. The prior probability $P(H)$ is computed according to the prior rule for PCFGs, which is the product of the prior probability of each primitive production rule R composing H: $P(H) = \prod_{R \in H} P(R)$. The highest posterior probability hypothesis is therefore the most concise one that fits the data.

In the likelihood, we assume that hypotheses' output may be slightly noisy, giving each digit in the output sequence a 0.01 chance of being flipped. This likelihood formulation weights generated sequences higher in the likelihood in proportion to their similarity with the observed data. In addition to the intuitive plausibility of a similarity-weighting likelihood metric, this likelihood helps MCMC learn correct hypotheses by providing a graded (non-modal) posterior space. We performed no model fitting, and all parameters were used "out-of-the-box".

We ran a Metropolis-Hastings sampler for 100,000 steps and stored the top 100 hypotheses with the highest posterior found on each incremental prefix of the sequence.

Hypotheses

In our model, hypotheses output binary sequences, corresponding to the binary colored symbols in the experiment. The production rules — which are the same across models — are themselves operations on sequences and integers that return sequences. The production rules we chose were simply chosen to roughly be the minimal set necessary to concisely represent the sequences humans saw:

- A^∞. Returns the symbol A repeating unboundedly.
- B^∞. Returns the symbol B repeating unboundedly.
- $Alternate(INT_1, INT_2)$. Returns the sequence of alternations of INT_1 and INT_2. E.g. $Alternate(2,3) \Rightarrow (A^2B^3)^\infty$.
- $Increment(INT_1)$. Returns the sequence of alternating repetitions of increasing length, starting from length INT_1. E.g. $Increment(2) \Rightarrow A^2B^3A^4B^5...A^{N-1}B^N....$
- $Append(SEQ_1, SEQ_2)$. Returns SEQ_2 on SEQ_1. E.g. $Append(A^2, B^2) \Rightarrow A^2B^2$.
- $Weave(SEQ_1, SEQ_2)$. Returns SEQ_2 weaved between SEQ_1. E.g. $Weave(A^2, B^2) \Rightarrow (AB)^2$.
- $Take(SEQ_1, INT_1)$. Returns the first INT_1 items from SEQ_1. E.g. $Take((AB)^5, 2) \Rightarrow AB$.
- $Invert(SEQ_1)$. Returns the inversion of SEQ_1. E.g. $Invert(B^3A) \Rightarrow A^3B$.

In these rules, INT could expand to the integers 1...10.

Models of Learning

We implemented three different LOT models to test various possibilities about human concept learning from experience: a baseline model which does not update; a model that updates the prior of primitives; and a model that adds previous high-posterior programs to its set of primitives. Each model was run on all 36 conditions in the experiment. Additionally, we implemented a unigram model of the sequence to compare against the LOT models. The LOT models all started with the same production rules, which we assumed to have a uniform prior probability. All models were implemented using a freely available software package called LOTlib (Piantadosi, 2014).

Non-Updating Model In the baseline model, the primitives and their priors were fixed between the first and second sequence, and did not change.

Rational Rules Model We implemented a version of the Rational Rules model (Goodman et al., 2008), which updates the priors over primitives according to their posterior-weighted production rule count. This corresponds to a Dirichlet-Multinomial model, in which counts of each production rule in the Maximum A Posteriori (MAP) hypothesis from the training sequence are summed and subsequently used in computing the primitives' priors when learning the transfer sequence. Since a higher count corresponds to a decreased penalty for use in a tree, this is essentially a way of increasing the prior for primitive production rules useful in learning the training sequence. We assumed a uniform prior over production probabilities in the training sequence.

Re-Use Model Upon learning a concept, people may store and re-use this concept as a primitive. The way we captured this idea in our model was by placing the MAP hypothesis from the end of the training sequence as a primitive for generating hypotheses in the transfer sequence. The hypothesis space over primitives was re-normalized such that the primitives retained a uniform prior probability after this primitive was added.

Unigram Model We implemented a unigram model that responds proportionally to the probabilities of previous symbols. More specifically, we modeled this as a beta-binomial over the counts of the digits with a uniform prior. The counts were updated starting on the first sequence and continued through the second sequence. This is a baseline comparison, as it implements (smoothed) probability matching without taking into account any contingency.

Results

Figure 3 shows the model's performance (with human data for comparison) at each step, collapsed over all sequences. The top panels display performance in the congruous condition and the bottom four show performance in the incongruous condition. It's worth noting again that these are predictions made with no model parameter tuning, but the rank-order speed of learning between models is unlikely to be affected by this. The first interesting thing to note is how well, and how quickly, each of the models learns in the congruous and incongruous conditions. The Re-Use model shows the greatest disparity between conditions, guessing accurately on average 66% of the time in congruous case and 54% of the time in the incongruous case, a difference of 12%. This is substantially higher than the difference in the Rational Rules model (4%), the unigram model (1%), and the no-updating

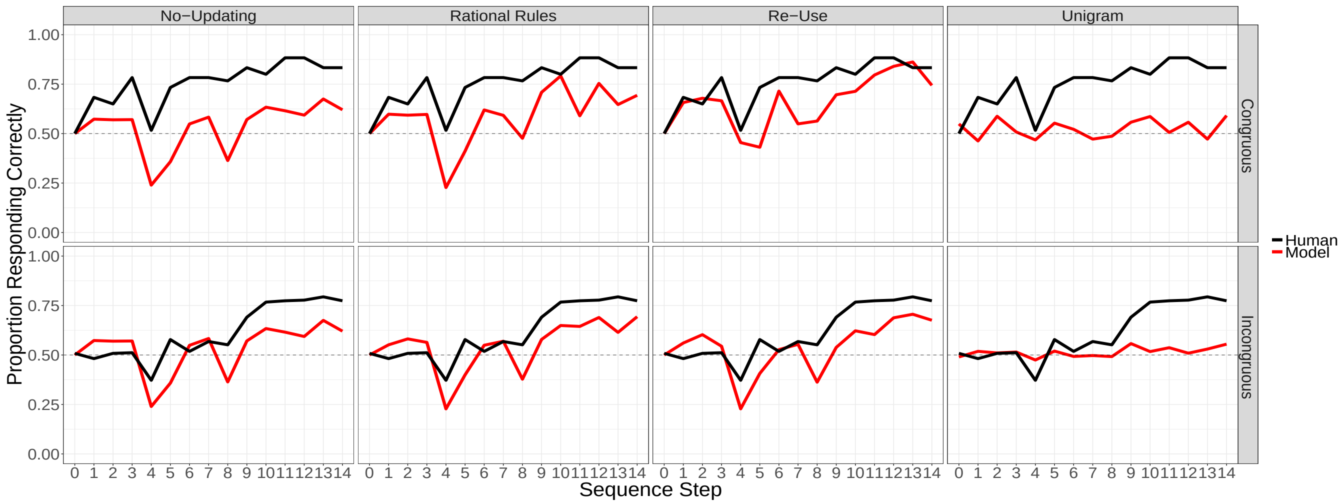

Figure 3: Overall model correctness overtime in the congruous (top) and incongruous (bottom) conditions, collapsed overall all sequences. The human data is shown in red in each plot, as comparison. The dashed line is the just the constant of y=0.5, for comparison.

Condition	Analysis	No Update	Rat. Rules	Re-Use	Unigram
Congruous	Mean Squared Error $(x10)$	0.57	0.37	**0.17**	0.65
Congruous	R^2	0.60	0.70	**0.72**	0.13
Congruous	Log Likelihood	−853	−796	**-736**	−870
Incongruous	Mean Squared Error $(x10)$	0.16	**0.12**	0.13	0.23
Incongruous	R^2	0.67	**0.75**	0.75	0.56
Incongruous	Log Likelihood	−4322	**-4268**	-4280	−4412

Table 2: Overall performance measured in Mean Squared Error, R^2, and Log Likelihood, for each of the models in both the congruous and incongruous condition. The best fit for each metric is bolded. Note that the log likelihoods can be compared like AIC values since there are no free parameters.

model (0%). This difference in the re-use model is most similar to humans, who responded correctly 75% of the time in the congruous congruous and 61% of the time in the incongruous condition, a change of 14%.

To more precisely compare the model and human fits for each sequence, we report the Mean Squared Error (MSE), R^2, and Log Likelihood to aggregate human responses, for each sequence and condition in Table 2. In both conditions, all the LOT models were significantly better fits than the unigram model. In the congruous condition, the Re-Use model was clearly a better fit than any other LOT model or the unigram model. The reason it out-performs all the other models in this case is primarily that none of the others learn the sequences fast enough. In the incongruous condition, the LOT models in this case perform more similarly than in the congruous condition, but the Rational Rules model provides a slightly better fit of the three according to each metric.

Discussion

The fact that the Re-Use model has the highest accuracy in the congruous condition (and closest to human-level) suggests that it is a better model of how humans' inferences benefit from helpful experience. The Re-Use model also displays the greatest disparity in accuracy between the two conditions, though still not quite as large in the gap in human performance between conditions (12% versus humans' 14%). Interestingly, the models display much more similar learning curves in the incongruous case. This means that the disparity in performance in the two conditions may be entirely due to the relative benefit of congruous experience – insofar as it changes primitives or their priors beneficially – but not as much to hindrance from incongruous experience. If true, this would predict that humans would perform about as well on the transfer sequence with no training sequence at all as with an incongruous training sequence.

To understand the Re-Use model's performance, it is informative to look at the actual representations that allow it to learn more quickly than the other models in the congruous condition. For each sequence, the MAP hypothesis from the first sequence is used in the MAP representation of the second sequence by the final step. Indeed, it is often orders of magnitude higher in the posterior than any other hypothesis. For instance, consider the case where the model sees:

$$((AB)^2B)^3$$

as training followed by:

$$((BA)^2A)^3$$

as transfer. The MAP hypothesis for the training sequence is displayed in orange in Figure 4.

This hypothesis gets added as a primitive, which we can call MAP_1. The shortest program on the transfer sequence that fits the data by the final step (and before), is simply invert(MAP_1), which is the entirety of the tree in Figure 4. This, of course, generates the inverse sequence generated by MAP_1, which is a simple and low-cost transformation when treating MAP_1 as a primitive. The tree representing the MAP hypothesis for the transfer sequence in the Re-Use model is much higher in the prior than the MAP representation both the Rational Rules model and the No-Update model construct, since it only uses two primitives, compared to their use of eight.

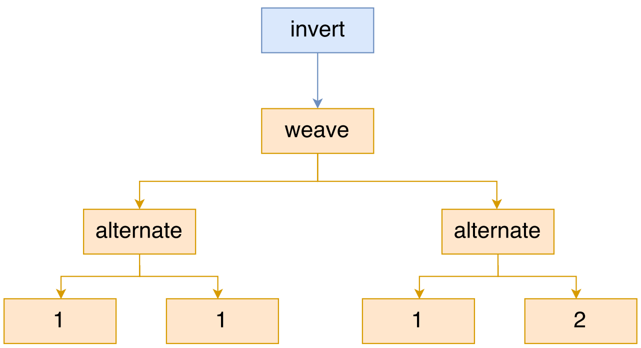

Figure 4: The Re-Use model's MAP hypothesis for generating repetitions of $((BA)^2A)$ in the congruous condition. The part in orange is the MAP hypothesis from learning the training sequence $((AB)^2)B$, and the blue is a transformation on it, treating it as a primitive production rule.

It is also interesting that the Rational Rules model provides the best fit in the incongruous condition, closely followed by the Re-Use model. This suggests that even when people can't transfer a whole concept, they still prefer using primitives of past hypotheses. One possibility to explore in the future is combining the Rational Rules and Re-Use models. Another potentially powerful model could account for partial sub-tree re-use. This would reflect the possibility that people not only store useful programs in their entirety, but store useful sub-programs. This added flexibility in recombination has been modeled using adaptor grammars (Johnson et al., 2006) and fragment grammars (O'Donnell et al., 2011). But inference in these models is substantially more complicated than models considered in this paper, and the extent of human flexibility in this regard remains an open question.

Conclusion

Our experiment showed that people benefit in learning a sequence given prior experience with an abstractly congruous sequence. By considering congruity as a function of similarity in LOT program-space, we can understand human knowledge transfer as changes in the representations and biases of LOT models. We showed that a LOT model that treats previously learned programs as primitive rules is the best fit to human data in the congruous condition. On the other hand, we found that the LOT model that rationally updates the prior on existing production rules is the best fit in the incongruous condition. This provides evidence that people spontaneously transfer knowledge of both whole programs and their subcomponents when learning.

Acknowledgements We would like to thank Jenna Register, Fred Callaway, Frank Mollica, and colala for helpful comments and conversations.

References

Dechter, E., Malmaud, J., Adams, R. P., & Tenenbaum, J. B. (2013). Bootstrap learning via modular concept discovery. In *23rd international joint conference on artificial intelligence.* Beijing, China.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, *41*(1), 1–63.

Fodor, J. A. (1975). *The language of thought*. Harvard University Press.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155 - 170.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, *12*(3), 306 - 355.

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

Goodman, N. D., Ullman, T. D., & Tenenbaum, J. B. (2011). Learning a theory of causality. *Psychological review*, *118*(1), 110.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive science*, *13*(3), 295–355.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological review*, *104*(3), 427.

Johnson, M., Griffiths, T. L., & Goldwater, S. (2006). Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In *Advances in neural information processing systems* (pp. 641–648).

Katz, Y., Goodman, N. D., Kersting, K., Kemp, C., & Tenenbaum, J. B. (2008). Modeling semantic cognition as logical dimensionality reduction. In *Proceedings of the cognitive science society* (Vol. 30).

Kemp, C., Goodman, N., & Tenenbaum, J. (2008). Learning and using relational theories. In *Advances in neural information processing systems* (Vol. 20, p. 753-760).

Markman, A. B. (1997). Constraints on analogical inference. *Cognitive science*, *21*(4), 373–418.

Mollica, F., & Piantadosi, S. (2015). Towards semantically rich and recursive word learning models. In *Proceedings of the cognitive science conference* (Vol. 37).

O'Donnell, T. J., Snedeker, J., Tenenbaum, J. B., & Goodman, N. D. (2011). Productivity and reuse in language. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*.

Piantadosi, S. (2014). *LOTlib: Learning and Inference in the Language of Thought.* available from https://github.com/piantado/LOTlib.

Piantadosi, S., Goodman, N., Ellis, B., & Tenenbaum, J. (2008). A Bayesian model of the acquisition of compositional semantics. In *Proceedings of the cognitive science society* (Vol. 30).

Piantadosi, S., & Jacobs, R. (2016). Four problems solved by the probabilistic language of thought. *Current Directions in Psychological Science*, *25*(1), 54–59.

Piantadosi, S., Tenenbaum, J., & Goodman, N. (2012). Bootstrapping in a language of thought: a formal model of numerical concept learning. *Cognition*, *123*, 199-217.

Siskind, J. M. (1996, Oct-Nov). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, *61*(1-2), 1-38.