Nonsense and Sensibility: Inferring Unseen Possibilities

Lauren A. Schmidt, Charles Kemp & Joshua B. Tenenbaum
Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology
{lschmidt, ckemp, jbt}@mit.edu

Abstract

How do we distinguish the sensible yet unlikely (blue bananas) from the nonsensical (hour-long bananas), given observations only of what is true in the world (e.g., yellow bananas)? Judgments like these may be supported by the M constraint: the assumption that ontological categories are organized into a predicability tree, and that properties apply to different subtrees within this hierarchy. We provide a computational theory that shows how the M constraint can be used to acquire predicability trees given observations only of what is true. We also suggest how the M constraint itself could be learned.

A friend comes home from the market exclaiming that she has just seen the most interesting banana. She tells you to guess what was so interesting about it. Which of the following questions are you most likely to ask? "Was the banana blue?" "Was the banana the shopkeeper's fault?" "Was the banana an hour long?" Probably you are more likely to ask the first question. You may never have seen or heard of a blue banana before, but it seems like one might exist somewhere, and such a banana would indeed be noteworthy. The other two questions, on the other hand, do not make sense. Bananas are not the kind of thing that can be someone's fault or an hour long.

How can we judge so quickly that it is far more sensible for bananas to be blue than an hour long, when we have seen no direct evidence in the world regarding either blue or hour-long bananas? We may never have seen a blue banana, but we have seen many other blue things—blueberries, bicycles, and bedspreads, for example—which share many properties with bananas. They are all visible objects, and we can touch them, move them, or give them to our friends. If these objects can sensibly be blue, then perhaps it would be sensible for bananas to be blue as well. None of the other blue objects is an hour long, however, which might make us more doubtful that bananas could sensibly have this property.

Kinds such as physical objects and events are examples of ontological categories. The properties that apply to the members of the ontological categories ("blue," "hour long") are also known as predicates. When a predicate can sensibly be applied to an object, and a truth value can be assigned ("Bananas are usually blue" is sensible, though false), that predicate is said to span the object (Keil, 1979).

Predicates do not appear to span objects arbitrarily. Instead, predicates seem to cluster together and apply to whole categories of objects. In addition, these categories do not overlap arbitrarily, but also seem to follow structural rules. Sommers (1971) proposed "the M constraint": categories of objects are organized in a strict hierarchy, and predicates must span subtrees of the hierarchy (thus preventing any "Ms" within the tree). See Figure 1 for an example of such a predicability tree, and how it governs the set of pairs which are sensible together, which in turn governs which pairs may be true.

This hierarchical constraint could be extremely useful in making inferences about what is sensible based on limited evidence. If we know that soccer games cannot sensibly be blue but can be an hour long, and bicycles can be blue but cannot sensibly be an hour long, then according to the M constraint, bananas cannot sensibly have both properties. Knowing this, all we need to do to make a quick inference is figure out whether bananas are more like bicycles or soccer games. Observations that bananas and bicycles can both be yellow and green, as well as sharing other properties, can help us infer that bananas could sensibly be blue, but not an hour long.

Keil (1979) provided some evidence that people do follow the M constraint in reasoning about the world. When asked to judge which statements about predicate-object pairs were sensible, adults provided sets of answers which showed a strict hierarchical organization. This was true for children as well, though their predicability trees were far simpler than those of adults. Keil also showed that children can make quick inferences based on hierarchical relationships of categories and predicates. Most interestingly, Keil proposed that the M constraint is a part of the innate core knowledge that guides children in learning about the world.

If the M constraint (or a weaker statistical bias) does exist as a psychological reality, how could we use it to learn what is sensible? We propose a formal model that incorporates the M constraint and can discover a predicability tree given limited evidence about what is true in the world. Our model takes the M constraint as given; however, we also describe how Bayesian model selection can be used to infer that a model with the M constraint accounts better for the observed data than an alternative model with no hierarchical constraint. We therefore suggest that the M constraint need not be innate.

A Structured Approach to Sensibility

Assume that we are working with a fixed collection of objects and predicates. We develop a computational theory that assumes that the objects and predicates are

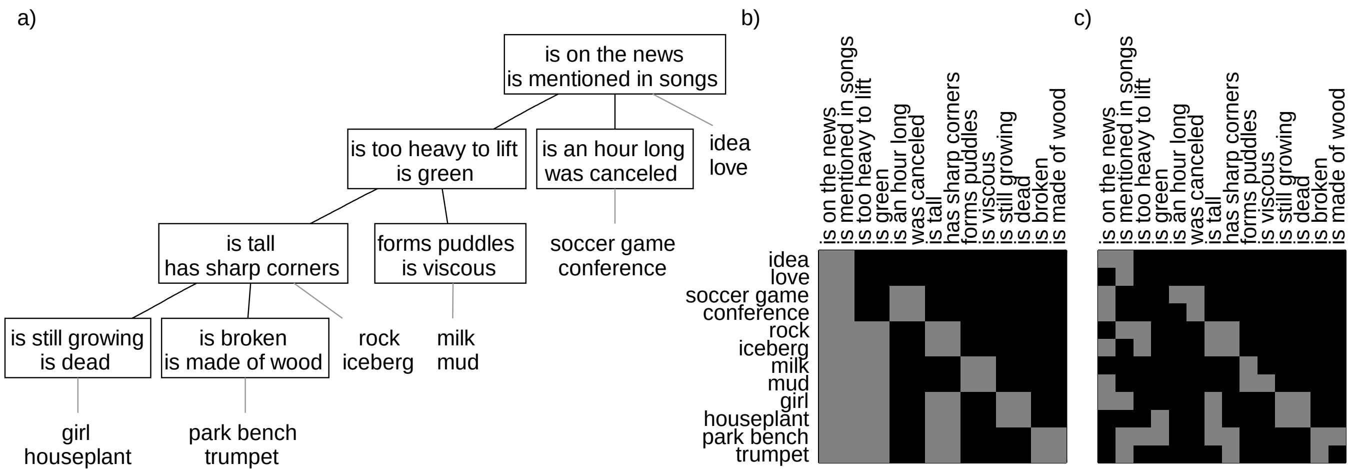

Figure 1: (a) A predicability tree. The predicates located at a node of a tree span all the objects at that node and all those in the subtree below it. (b) The corresponding predicability matrix, R. Squares in gray indicate predicable pairs. (c) A truth matrix, T, indicating the predicable pairs that are actually true.

organized into a predicability tree (Figure 1a). Each predicability tree can be represented as a predicability matrix, R (Figure 1b). Some of the predicable pairs are also true, and we represent these pairs using a truth matrix, T (Figure 1c). We assume that a learner observes a set of predication events (such as a yellow banana or blue bicycle), where each event is an occurrence of a predicate-object pair. The data can be arranged into a frequency matrix D, where D_{ij} indicates the number of times that predicate i was paired with object j. Our model assumes that the large entries in D correspond mainly to pairs that are true: more precisely, our model assumes that each predicate-object pair is observed with a frequency that depends on the truth of the pair and the individual popularities of both the predicate and the object. Our ultimate goal is to work backward from the data D to learn the predicability matrix R and truth matrix T.

We take a generative approach, and define a joint distribution $P(D, T, R) = P(D|T, R)P(T|R)P(R)$. This generative model can then be inverted to search for the R and T with maximum posterior probability:

$$P(T, R|D) \propto P(D|T, R)P(T|R)P(R). \tag{1}$$

To apply Equation 1, we need a prior $P(R)$ on predicability matrices. We assume that predicability satisfies the M constraint, and that each candidate R corresponds to a tree like the example in Figure 1. Each predicability tree can be parameterized as a triple (z_o, B, z_p) where z_p is a partition of the predicates, B is a tree built using the predicate clusters in z_p, and z_o is an assignment of the objects to the nodes in B. Each triple (z_o, B, z_p) uniquely specifies a tree with objects and predicates attached, which in turn uniquely specifies a predicability matrix R. The prior probability of any matrix R is

$$P(R) = P(z_o, B, z_p) = P(z_o|z_p)P(B|z_p)P(z_p)$$

We compute $P(z_p)$ by assuming that z_p is generated by a Chinese restaurant process with concentration parameter γ (Aldous, 1985). This prior on z_p assigns some probability to all possible partitions of the predicates, including the partition where each predicate is assigned to its own cluster, and the partition where all the predicates belong to the same cluster. The prior, however, favors partitions that use small numbers of clusters.

Suppose that $|z_p|$ is the number of clusters used by partition z_p. We assume that B is drawn uniformly from all of the $|z_p|^{|z_p|-1}$ possible labeled trees with $|z_p|$ nodes, and z_o is generated by dropping each object at random onto one of the nodes of B:

$$P(B|z_p) = \frac{1}{|z_p|^{|z_p|-1}} \tag{2}$$

$$P(z_o|z_p) = \frac{1}{|z_p|^n} \tag{3}$$

To apply Equation 1, we also need to calculate $P(T|R)$, the probability of the truth matrix given the predicability matrix. We assume that the truth of each predicable pair is determined by flipping a coin with bias η:

$$P(T|R) = \begin{cases} \eta^{|T|}(1-\eta)^{|R|-|T|}, & \text{if } T \subset R \\ 0, & \text{otherwise} \end{cases}$$

where $|R|$ is the number of predicable pairs and $|T|$ is the number of true pairs.

Finally, we assume that each predication event v, consisting of an occurrence of an object-predicate pair (o, p), is drawn from a distribution given by

$$P(o, p) \propto e^{\pi_p + \pi_o + \lambda T_{po}} \tag{4}$$

where π_p and π_o are parameters representing the popularities of predicate p and object o, and λ represents the extent to which we penalize violations of the truth matrix T. If T is uniformly one (i.e. every object-predicate pair is both sensible and true) or λ is zero (i.e. we do not penalize violations of matrix T, then the model reduces to a simple joint distribution where $P(o,p) \propto P(o)P(p)$, $P(o) = e^{\pi_p}$ and $P(p)$ is similarly defined.

The popularity parameters could potentially be learned, but we fix them using the frequencies of objects and predicates in the matrix D:

$$\pi_o \propto \log |\pi_o| \tag{5}$$

where $|\pi_o|$ is the proportion of events in D that involve object o. π_p is defined similarly.

The probability of the entire dataset is the product of the probabilities of the individual predication events, v :

$$P(D|T) = \prod_{v \in D} P(v|T). \tag{6}$$

Searching for predicability matrices

We run a stochastic search to identify the $R = (z_o, B, z_p)$, T, λ, and η that maximize $P(R,T|D)$. The search problem is difficult, since the score for a matrix R couples z_o and z_p: in other words, changing z_p is unlikely to improve the score unless z_o is changed as well. We overcome this issue by integrating out the object locations z_o, and searching for the B, z_p and T that maximize

$$P(B, z_p|D) \propto P(D|B, z_p)P(B, z_p) \tag{7}$$

Suppose that D_j is the set of predication events that involve object j. Then

$$P(D|B, z_p) = \prod_j P(D_j|B, z_p)$$

and

$$P(D_j|B, z_p) = \sum_{k=1}^{|z_p|} \sum_{t_j} P(D_j, t_j, z_o^j = k|B, z_p) \tag{8}$$

where z_o^j is the location of object j in the tree, and t_j is a vector indicating which predicates are true of objects. Computing the sum in Equation 8 is intractable, and we approximate it as follows:

$$P(D_j|B, z_p) \approx \sum_{k=1}^{|z_p|} P(D_j|t_j^*(k))P(z_o^j = k|B, z_p) \tag{9}$$

where $t_j^*(k)$ is the truth vector that maximizes $P(t_j|z_o^j = k, B, z_p)$. Equation 3 implies that $P(z_o^j = k|B, z_p) = \frac{1}{|z_p|}$. We do not include the details here, but if we condition on the number of times each object appears in the dataset, it is straightforward to compute $P(D_j|t_j^*(z_o^j = k))$: in particular, we avoid having to compute the normalizing constant of the distribution in Equation 5.

Let us call (B, z_p) an *incomplete* tree: that is, a predicability tree without the objects attached. Using Equation 7, we run a search by starting from a random incomplete tree and considering proposals that move the predicates around the tree, possibly creating new nodes in the process. For each incomplete tree, we use an approximation similar to the idea behind Equation 9 to compute a candidate pair (T, R), where T and R are the matrices that maximize $P(T, R|B, z_p, D)$. At the end of the search, we return the best candidate pair encountered, where each pair is scored according to Equation 1.

Learning the right tree

We compared the performance of our model and two alternate approaches on data sets meant to approximate real world data.

The data sets We generated two data sets of observed predication events based on a tree with the same structure as that shown in Figure 1. Our tree had six predicates located at each node, and three objects located at each node except for one internal node, which was empty, yielding a total of 42 predicates and 18 objects. This structure yielded the predicability matrix shown at the top of Figure 2.

We then set parameters π, λ, and η, and sampled data according to our model. We sampled $|\pi|$ from a uniform distribution. λ was set to 10. We used truth matrices with $\eta = \{0.3, 0.5, 0.7, 0.9\}$. For each truth matrix, we generated three different data sets, differing in number of samples drawn, N, where N took the values $\{1000, 10000, 100000\}$. 1% of the samples were drawn uniformly over all pairings, creating some noise in the data. The average number of times each true predication event was seen ranged from 3.63 to 1089 across conditions; values are shown in Table 1. The generative process and resulting data sets are illustrated in Figure 2.

The comparison methods We sought a comparison model that would also learn a tree structure based on clusters of objects and predicates and allow for inference about predicabilities of unseen pairs. Hierarchical clustering is a popular method for learning tree structures, however, the standard algorithm is insufficient for our needs. The same is true of Bayesian tree learning techniques proposed in the past (Kemp, Perfors & Tenenbaum, 2004). Instead of finding hierarchies based on object categories, the trees recovered by both methods branch maximally such that each object is alone at a leaf node. Therefore, neither method can be used to identify the large-scale ontological categories that we hope to recover. Additionally, the predicates are not clustered together or placed at nodes within the tree.

Our comparison method is a modified version of hierarchical clustering which overcomes some of the above shortcomings and generates predicability predictions that serve as a comparison for those of our model. We used a standard hierarchical clustering algorithm to find a hierarchical organization of objects based on the frequency vectors of their occurrences with predicates in the data. We then developed a metric for scoring possible predicate locations and then placed each predicate at the best scoring node in the tree.

To score a potential predicate location, we looked at what data would be predicted by that predicate being placed at that node. Any objects not spanned by the predicate were predicted never to occur. Any objects spanned by the predicate were predicted to occur with that predicate a number of times proportional to the product of the object frequency and the predicate frequency in the data set. The score of the location was proportional to the inner product of the normalized predicted and actual data vectors.

In addition to this comparison method, we also tried a

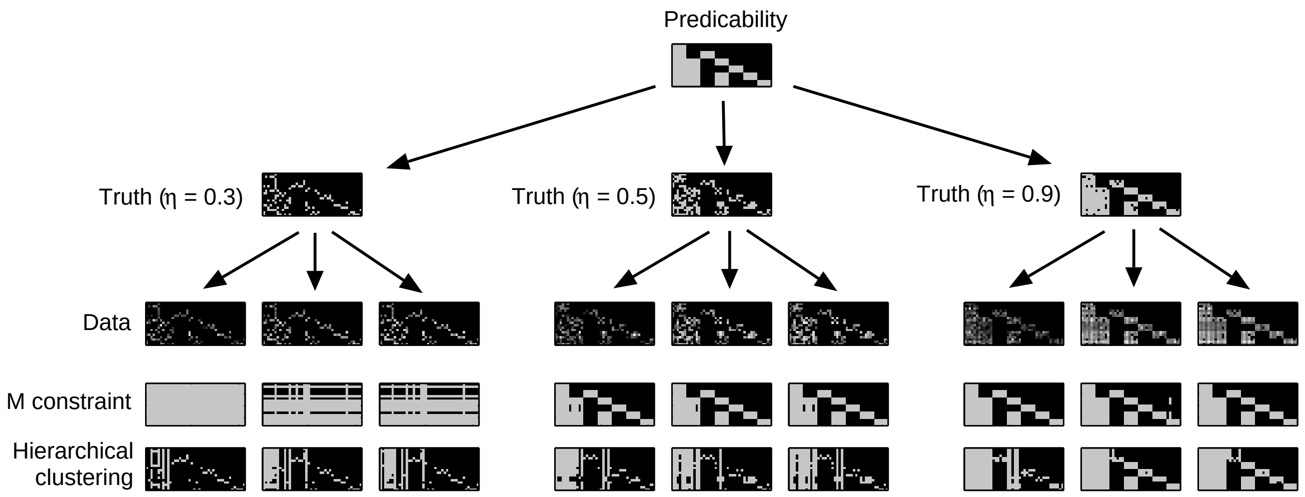

Figure 2: Data sets and results for the M constraint model and the hierarchical clustering model. The three truth matrices were generated by selecting random subsets of predicable pairs. The input for each model is a frequency matrix generated by sampling pairs from the truth matrix. Three data sets with increasing numbers of observations are shown for each truth matrix. The final two rows show predicability judgments made by the two models.

simple thresholding method which predicted that exactly those pairs that had been seen in the input data one or more times were predicable.

Results The predicability matrices recovered by the M constraint and hierarchical clustering models are shown in Figure 2 (the predicability matrix from the other comparison method is not shown, as it is simply a thresholded version of the data matrix). The binary matrices recovered can be said to have high precision whenever most of the predicability predictions made are correct:

$$\text{precision} = \frac{H}{H + FA}$$

where H is the number of "hits", or pairs that were correctly predicted to be predicable, and FA is the number of "false alarms", or pairs that were incorrectly predicted to be predicable. The recovered matrices have high recall when most of the actual predicable pairs have been successfully recovered:

$$\text{recall} = \frac{H}{H + M}$$

where H is again the number of "hits", and M is the number of "misses", or pairs that are actually predicable but were not predicted. We measured overall success of the models using the F-measure, the harmonic mean of precision and recall. The F-measures of these results with the true predicability values are shown in Table 1 (results are averaged across the two data sets). The truth matrices recovered by our model were very close to the actual truth matrices, having F-measures of 0.967 or higher in all cases; the details of those results are omitted.

With relatively dense input data, the M constraint recovers a perfect or nearly perfect predicability matrix. The M constraint model outperforms both the hierarchical clustering method and the thresholding method on all data sets; however, with the sparsest data sets none of the algorithms perform well. In those cases, the M constraint drastically overgeneralizes, while the other models do the opposite. Interestingly, the simple thresholding model also outperforms the hierarchical clustering model on all but some of the sparsest data sets.

We would suggest that the M constraint model is doing something both intelligent and psychologically plausible in the cases where it overgeneralizes based on sparse data. When someone learning about the world has only made a few observations, then her confidence about the structure of the world should be very low. Such a learner cannot yet distinguish between occurrences that are likely, but have not yet been observed, and ones that are extremely unlikely. She also does not know with any confidence which of the observations she has made so far are due to noise. Because of all this uncertainty, it

Table 1: F-measures when the results of the three models are compared with original predicability matrix. SPT is samples per true pair.

η	Samples	SPT	M con.	H.clus.	Thresh.
0.3:	1000	10.89	**0.58**	0.45	0.47
	10000	108	**0.61**	0.58	0.48
	100000	1089	**0.61**	0.58	0.48
0.5:	1000	6.54	**0.98**	0.64	0.66
	10000	65.4	**0.99**	0.65	0.66
	100000	654	**0.98**	0.66	0.67
0.7:	1000	4.67	**0.86**	0.68	0.82
	10000	46.7	**0.97**	0.72	0.83
	100000	467	**0.97**	0.75	0.81
0.9:	1000	3.63	**0.99**	0.75	0.94
	10000	36.3	**0.98**	0.83	0.95
	100000	363	**1.00**	0.83	0.94

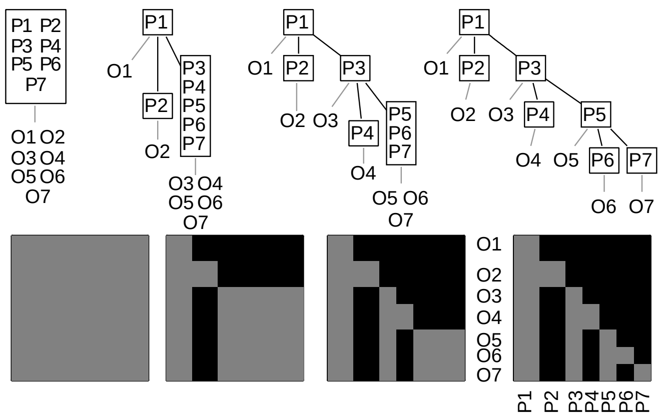

Figure 3: A developmental progression showing the predicability trees and corresponding predicability matrices learned by the M constraint model as it receives increasing amounts of data. The labels P1...P7 and O1...O6 represent clusters of predicates and objects, each of which has two or three members.

would be unwise for the learner to speculate that the structure of the world is some highly complicated tree-structure that fits the handful of observations she has made. It is better to postulate a simpler structure until the learner has more information. Our M constraint model captures this intuition, choosing relatively simple trees until the data warrant otherwise. The hierarchical clustering method, on the other hand, is like a learner who creates maximally complex tree structures to fit every data set, regardless of how much evidence she has seen. The thresholding method does not learn a tree structure to fit the data set, but it also undergeneralizes in its predicability predictions in a way that does not match the human developmental data.

To demonstrate in a more controlled way the behavior of the M constraint model when operating on sparse data, we generated simpler data sets. In these data sets, $\eta = 0.85$. Every true pair is observed a fixed number of times (N). Thus, for $N = 1$, every true pair has been seen one time.

We ran the M constraint model on data sets generated for $N = 1$, 2, and 3, and 6. The best scoring predicability trees and matrices can be seen in Figure 3. The developmental progression of our model is similar to the human developmental progression reported by Keil. When the M constraint model has seen very little evidence, it behaves like the younger learners, choosing simpler trees. As more data are provided, the trees recovered by the model grow to look more like those of Keil's older subjects, who also had more data about the world. The similarity between these developmental curves argues for the psychological plausibility of a model that develops more complex theories only when sufficient evidence is available.

Learning the M constraint

We have shown that our M constraint model can discover predicability structures given only limited evidence. Our model, however, assumes that the M constraint is true: in other words, it assumes that the matrix R is tree-structured. The corresponding cognitive assumption is that the M constraint is innate, an assumption we may not wish to make.

In previous work we have argued that Bayesian model selection helps explain how learners can discover the structural properties that best characterize a domain (Kemp et al., 2004). Here we demonstrate that the M constraint could be learned by comparing our model to a closely related clustering model that does not include the M constraint. The same method could be used to compare a wider set of possible models, but here we choose only between clustering with and without the M constraint.

A flat clustering model Our alternative model clusters predicates and objects, as the M constraint model does, but it does not impose any structural restrictions (such as the M constraint) on how predicate clusters relate to object clusters.. For instance, predicate cluster 1 could span object clusters 1, 2, and 3, and predicate cluster 2 might span object clusters 2, 3, and 4. This model, which we refer to as the flat model, is identical to the M constraint model except that it uses a different prior on the predicability matrix R. We parameterize each matrix as a triple (z_o, C, z_p). As before z_p is a partition of the predicates. C is a $|z_p|$ by $|z_p|$ matrix of constraint vectors drawn uniformly from the space of binary matrices of this size. Each row of the matrix corresponds to a predicate cluster, and z_o is a random assignment of the objects to the columns of this matrix:

$$P(C|z_p) = \frac{1}{2^{|z_p|}} \tag{10}$$

$$P(z_o|z_p) = \frac{1}{|z_p|^m} \tag{11}$$

Each triple (z_o, C, z_p) uniquely determines a predicability matrix R where R_{ij} takes the same value as the entry in C corresponding to predicate i and object j. Note that R need not satisfy the M constraint: the model captures the idea that predicates and objects cluster, but little more. As in the M constraint model, the flat clustering model represents truth as a matrix T that is a subset of R.

Bayesian model selection Using Bayesian model selection, we can discover which of these models is best supported by a given dataset. Let M_{tree} indicate the M constraint model, and M_{flat} indicate the alternative. Given data D, we search for the model M and predicability matrix R that have maximum joint posterior probability:

$$P(R, T, M|D) \propto P(D|M, R, T)P(T|R, M)P(R|M)P(M)$$

We use equal priors on the two models: $P(M_{tree}) = P(M_{flat}) = 0.5$. Let R_{tree} and T_{tree} indicate the predicability matrix and truth matrix that maximize $P(R, T|D, M_{tree})$, and R_{flat} and T_{flat} indicate the matrices that maximize $P(R, T|D, M_{flat})$. If the data are consistent with a tree structure, then R_{tree} and R_{flat}

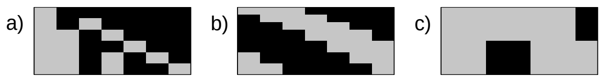

Figure 4: a) The tree-consistent data used for model selection. b) The tree-inconsistent data. c) The tree-consistent predicability matrix recovered by the M constraint model, given the data in b).

should be identical (as well as T_{tree} and T_{flat}), since the flat model is capable of representing all tree-consistent predicability matrices. $P(R_{tree}, T_{tree}|M_{tree})$, however, will be greater than $P(R_{flat}, T_{flat}|M_{flat})$, since the flat model needs to assign some probability mass to the many matrices that are not tree-structured. As a consequence, Bayesian model selection will favor the tree model if the data are consistent with a tree structure. If the data are consistent only with the flat model, then the tree model will be unable to represent a predicability matrix that accounts well for the data, and the flat model will be favored by model selection.

Model selection simulation We generated data sets from two different predicability matrices. Each contains the same number of predicates and objects, but one set is consistent with a tree structure, and the other violates the M constraint (see Figures 4a and 4b). Within the second data set, predicates span overlapping sets of objects. The frequencies in the dataset were generated by sampling each predicable pair 100 times.

Table 2 compares posterior probabilities $P(R_{flat}, T_{flat}, M_{flat}|D)$ and $P(R_{tree}, T_{tree}, M_{tree}|D)$ for the two datasets. As expected, the M constraint model scores better than the flat model on the tree-consistent data. The difference in performance may appear subtle, but the scores are log-posteriors and represent a difference of 22 orders of magnitude. For the tree-inconsistent data set, the flat model performs better than the M constraint model, because the M constraint model is unable to find a tree that produces predicability patterns that match the data. The M constraint model recovers the appropriate truth matrix, but must overgeneralize in order to find a tree that could produce the observed data (see Figure 4c). These results confirm the intuition that a learner with a hypothesis space including several different representations could choose the representation best supported by a given dataset.

Table 2: Log-posterior scores for the best possible configuration each model recovered in each condition.

Model	Tree-consistent	Tree-inconsistent
M constraint	**-94726**	-94255
Flat	-94748	**-93933**

Conclusion

We have shown how a generative Bayesian model incorporating the M constraint can be used to learn a predicability tree and infer what is sensible about the world given sparse observations of what is true in the world. Additionally, we have demonstrated how Bayesian model selection can be used to learn the M constraint given a hypothesis space including alternate models. If people do organize predicates and objects hierarchically, this result suggests that the hierarchical bias could be learned rather than innate.

Much work remains to be done in exploring this model further. Measuring model performance on real world datasets is an important future step. As Carey (1983) and others have pointed out, real world data may contain many exceptions to the M constraint. One advantage of a probabilistic model is that it can tolerate noise and exceptions; similarly, it may be possible that learners still have a strong hierarchical bias, but can make exceptions when there is sufficient evidence. Our model could be adapted to do the same. The M constraint may also be violated by cross-cutting systems of categorization (e.g., taxonomic vs. ecological categories in biology), but previous work suggests that multiple context-sensitive models, each representing a different hierarchical (or other) structure, could be learned to capture reasoning about different aspects of a complex domain (Shafto et al., 2005). We have only begun to explore the learning problem of distinguishing nonsense from sensibility, but our Bayesian model demonstrates that this distinction is a statistically learnable one, even in a case with no direct evidence, like that of the blue banana.

Acknowledgments

We thank Tom Griffiths for his help in developing this work. This work is supported by an NSF Graduate Research Fellowship (LAS), the William Asbjornsen Albert memorial fellowship (CK), and the Paul E. Newton chair (JBT).

References

Aldous, D. (1985). Exchangeability and related topics. In *Ecole d'Ete de Probabilites de Saint-Flour, XIII–1983*, 1-198. Springer, Berlin.

Carey, S. (1983). Constraints on natural kind terms. In T. Seiler and W. Wannenmacher (eds.), *Concept Development and the Development of Word Meaning.* Springer, 126-146.

Keil, F. (1979). *Semantic and conceptual development: an ontological perspective.* Cambridge, MA: Harvard University Press.

Kemp, C., Perfors, A., & Tenenbaum, J. B. (2004). Learning domain structures. *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society.*

Shafto, P., Kemp, C., Baraff, L., Coley, J., & Tenenbaum, J. B. (2005). Context-sensitive induction. *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society.*

Sommers, F. (1971). Structural ontology. *Philosophia*, *1*(1), 21-42.