STAT840 - Word Learning

Sugandha Sharma (s72sharm@uwaterloo.ca)
Center for Theoretical Neuroscience, 200 Unversity Ave. W.
Waterloo, ON N2L3G1 Canada

Abstract

This paper presents a computational model of word learning. A model of such kind can give us insight into how humans learn words and also help us design artificial intelligence systems that can learn words based on similar computational principles. The challenge is to implement a model which learns the meanings of words and is also able to generalize them to other words. In this paper, I have used the Bayesian Framework to accomplish this by replicating the model proposed by Xu & Tenenbaum (2007). I found that the model's behaviour is qualitatively and quantitatively similar to people's patterns of generalization when an appropriate hypothesis space and prior are used.

Keywords: Word Learning; Markov Chain Monte Carlo; Rejection Sampling; Metropolis Algorithm

Introduction

Word learning is an important area of research since it can give us insights into how our brain is able to learn word extensions so effectively and efficiently. This can further lead us to understand how we learn language and answer questions like "why are childrent better at learning languages than adults?". Finding the computational principles behind word learning can also help us design and improve artificial intelligence systems which are adept at using and learning language.

Background

Researchers have been trying to explore computational approaches to how humans learn the meanings of words since long. *Hypothesis elimination* and *associative learning* are the two theories which have been dominant in the literature about how word learning works (Fazly et al., 2010). In the hypothesis elimination approach, learning process involves eliminating incorrect hypothesis about word meanings until convergence to a single consistent hypothesis. For example, Siskind (1996) presented an efficient algorithm which kept track of just the necessary and possible components of word meaning hypothesis which were consistent with a set of examples. A weakness of this approach is that some logically possible hypothesis or concepts cannot be recovered once they are eliminated. Moreover, even after ruling out all hypothesis inconsistent with a given labelled example, a learner will still be left with many consistent hypothesis. For example if someone says to a child while pointing at a Dalmatian dog "look Max is running away", how would the child know whether Max refers to only that particular dog, to all dogs, or to all animals?. This is the problem of inference in a hierarchical taxonomy which poses the problem of learning with overlapping hypothesis about a given word and what it refers to. The hypothesis elimination approach fails to solve this problem of learning words with overlapping hypothesis. An example of a hierarchical taxonomy is shown in Figure 13 in the appendix.

Another approach to word learning is associative learning, an example of which was shown in a model by Yu (2005). They studied a word-object association model in a unified framework of lexical and category learning and their model demonstrated the emergence of patterns observed in early word learning. Most of the people following the associative learning approach use connectionist networks (Burns et al. (2003); Smith (2000)). Through the use of internal layers of hidden units and appropriately designed representations, these models are able to produce generalizations of word meanings that go beyond the direct word-percept associations. It is not clear however that the associative models can solve the overlapping hypothesis problem mentioned before (Xu & Tenenbaum, 2007). For example, given a word 'animal' it can refer to a dog, a particular kind of dog e.g., 'dalmatian' or to another animal. However, one standard mechanism in associative models is that the models implicitly assume that a positive example of one word is a negative example of every other word. Some of these models also fail to explain how multiple words can apply to a single object, since they use competition among outputs to implement the implicit negative evidence e.g., MacWhinney (1998).

Computational approach followed

Xu & Tenenbaum (2007) have argued that the problem of learning overlapping word meanings from sparse positive examples can be solved by a Bayesian approach to word learning which combines prior knowledge with the observed examples of a word's referents.

It has been found that adults show a basic level bias i.e., they map common nouns preferentially to basic level categories (Rosch et al., 1976). Basic level categories are clusters of intermediate size eg., category of dogs, that maximize many different indices of category utility relative to subordinate (e.g., dalmatians) or superordinate (e.g., animal) categories that contain them. However, children do not show a strong basic-level preference when taught unfamiliar words (Xu & Tenenbaum (2007); Callanan et al. (1994)). This suggests that a basic level bias might not be a part of foundations of word learning, but such a bias might develop as children learn more about general patterns of word meanings and how words tend to be used.

To explain this further, lets assume a learner has a taxonomic hypotheses space with basic, superordinate and subordinate categories (shown in Figure 13 in the appendix). Lets also assume that this learner has a preference for labeling basic level categories. If this learner is shown Max the Dalma-

tian, labelled as 'fep', he might reasonably guess that 'fep' refers to all dogs. Now, lets say that the learner observes three more objects labeled as 'feps' each of which is also a Dalmatian. After seeing these three additional examples, no potential hypothesis can be ruled out as inconsistent that were not inconsistent after seeing the first one. However, these additional examples make the word 'fep' seem relatively more likely to refer to just Dalmatians than to all dogs. It would be surprising to observe only Dalmatians called 'feps' if the word referred to all dogs and if the four examples were a random sample from the world. This intuition can be captured by a Bayesian inference mechanism that scores alternative hypothesis about a word's meaning based on how well they predict the observed data and how they fit with the learner's expectations.

The aim of my project is to replicate the Bayesian model for word learning proposed by Xu & Tenenbaum (2007). My focus is on replicating the foundations of word learning i.e., learning without any basic level bias, and then adding in the bias to see its effect. I propose to first build a relatively simple model with a small taxonomic hypothesis space shown in Figure 1, and then extend the model to the full taxonomic hypothesis space shown in Figure 2 (Xu & Tenenbaum, 2007). In the rest of this paper, I will present the methodology I used to implement the model including the algorithms used and design decisions made. This will be followed by describing the results obtained and their comparison to the experimental data and to the model built by Xu & Tenenbaum (2007). Lastly, I will finish with a discussion on the model performance and potential future work.

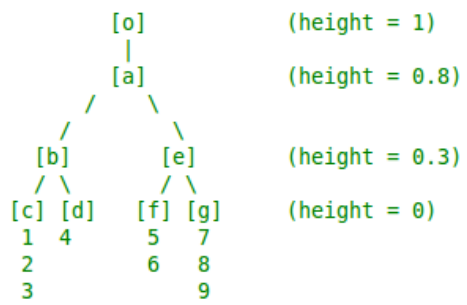

Figure 1: A small taxonomic hypothesis space. Letter codes refer to specific clusters i.e., hypothesis for word meanings.

Methodology

The aim of this project was to build a computational model that is able to learn a single novel word C from a few examples. Lets assume that $X = x^1, ...x^n$ are a set of n observed examples of the novel word C (the examples are drawn from a known domain of entities). The two goals then are:

- Given X examples of word C, the model should be able to figure out which hypothesis is the true meaning of the word C.
- The model should be able to generalize what it learns. For example, once it has found the true meaning of word C, given another example 'y' it should be able to decide whether 'y' belongs to the extension of C.

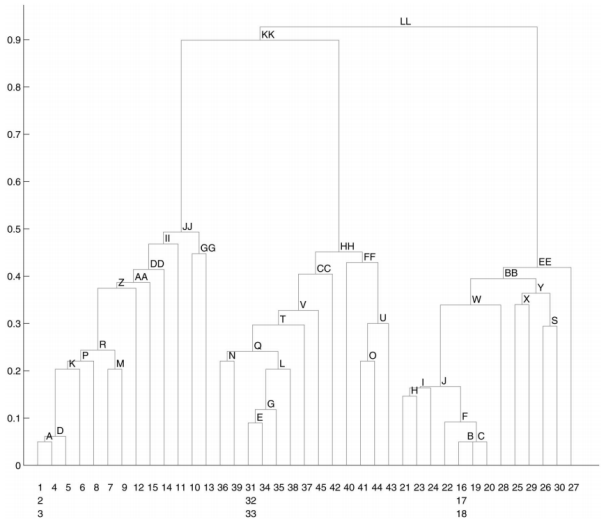

Figure 2: The full taxonomic hypothesis space obtained by hierarchical clustering of human similarity judgments. Letter codes refer to specific clusters i.e., hypothesis for word meanings: vegetables (EE), vehicle (HH), animal (JJ), pepper (J), truck (T), dog (R), green pepper (F), yellow truck (G), and Dalmatian (D). The numbers below the x-axis indicate the domain of entities from which the examples are drawn. The position of these examples on the plot indicates the hypotheses/categories to which they belong. (Xu & Tenenbaum, 2007)

I assume that the learner has access to a hypothesis space H of possible concepts and a probabilistic model relating hypotheses $h \in H$ to data X. This hypothesis space is shown in Figure 2 and was obtained experimentally by Xu & Tenenbaum (2007). Each internal node of the tree corresponds to a cluster of entities that are more similar to each other on average, than to nearby objects. The height of each node represents the average pairwise dissimilarity between the objects in that cluster and the length of the branch above each node indicates cluster distinctiveness i.e., how much more similar are that cluster's members to each other on average than to objects in the nearest cluster. Each hypothesis h points to some subset of entities in the domain that is a candidate extension for C. I also assume that the learner is capable of identifying which entities fall under each hypothesis i.e., extension of each hypothesis. The Bayesian learner evaluates all hypotheses according to Bayes' rule by computing their posterior probabilities, given examples X. The posterior probabilities are proportional to the product of the prior probabilities and likelihoods and are given by equation 1.

$$p(h|X) = p(X|h)p(h)/p(X) \quad (1)$$

Prior

The prior *p(h)*, in combination with the hypothesis space itself represents the learner's expectations about the plausible meanings of the word C, independent of the observed examples X. It is easier for the learner to distinguish between the clusters that are relatively more distinct. For example, it is easier to distinguish a cat from a dog than to distinguish two dogs of different types (e.g., a Labrador and a Dalmatian). Thus this preference for cluster distinctiveness is captured in the prior by taking the prior to be proportional to the branch length of each node as given by equation 2.

$$p(h) \propto height(parent[h]) - height(h) \quad (2)$$

Likelihood

The likelihood $p(X|h)$ captures the statistical information from the observed examples i.e., the expectations about which entities are likely to be observed as examples of C given a particular hypothesis h about C*'s* meaning.

Likelihood is computed based on the size principle as given by equation 3 if $x_i \in h$ for all i, and 0 otherwise. The assumption here is that the observed examples are independent and each hypothesis has a finite size i.e., finite number of entities that belong to it. Consider a hypothesis which consists of K entities. The likelihood of picking any one entity at random from this set of size K would be $1/K$, and $1/K^n$ for n objects sampled with replacement.

The likelihood is thus based on the size of extension of each hypothesis. Though we do not have access to the "true" size of the hypotheses, e.g., set of all dogs in the world, we can use the within cluster dissimilarity i.e., the cluster height in the tree as a psychologically plausible substitute. Thus the likelihood is then given by equation 4 if if $x_i \in h$ for all i, and 0 otherwise. Note that ε is not a parameter, but just a constant which is added to prevent the likelihood from going to infinity for the nodes with height zero. Thus a fixed value of $\varepsilon = 0.05$ is chosen and hence all nodes in Figure 2 are shown at a height of 0.05 above their true heights reflecting this value.

$$p(X|h) \propto [1/size(h)]^n \quad (3)$$

$$p(X|h) \propto [1/(height(h)+\varepsilon)]^n \quad (4)$$

Posterior

The posterior reflects the learner's belief that h is the true meaning of C given observations X and the prior knowledge about plausible word meanings. It is thus given by combining the prior and the likelihood according to the the Bayes' rule as given in equation 5 if $x_i \in h$ for all i, and 0 otherwise.

$$p(h|X) \propto [1/(height(h)+\varepsilon)]^n [height(parent[h]) - height(h)] \quad (5)$$

Generalization

Xu & Tenenbaum (2007) defined a way to relate learner's beliefs about the word meaning encoded in $p(h|X)$ to generalization behaviour. Once the learner has found the true meaning of word C, given another example 'y', it needs some way to decide whether 'y' belongs to the extension of C. If $p(X|h) = 1$ for exactly one hypothesis ($h = h^*$) and 0 for all others, then C applies only to those new objects $y \in h^*$. However in a more general case, the learner should compute the probability of generalization by averaging the predictions of all hypothesis weighted by their posterior probabilities as given by equation 6.

$$p(y \in C|X) = \sum_{h \in H} p(y \in C|h)p(h|X) \quad (6)$$

Note that in equation 6, $p(y \in C|h) = 1$ if $y \in h$ and 0 otherwise. Moreover, $p(h|X) = 0$ unless the examples X are all contained within hypothesis h. Thus the generalization probability gets reduced to the sum of posterior probabilities of all hypotheses that contain both the new example 'y' and the old examples X as given by equation 7.

$$p(y \in C|X) = \sum_{h \supset y,X} p(h|X) \quad (7)$$

Comparison of Sampling Techniques

Rejection Sampling : First, I built a small model using a variant of rejection sampling on the hypothesis space shown in Figure 1. Following approach was used to implement rejection sampling:

1. Calculate the prior weights of all nodes (nodes represent hypothesis) and normalize them such that they lie between $[0,1]$.
2. Generate a multinomial distribution over the nodes based on the prior weights and sample a node from the it.
3. Check whether all the examples provided are children of this sampled node. If not reject this node sample and draw another node from the multinomial.
4. If the node is the parent of all examples, then compute the likelihood of the node.
5. Flip a biased coin with probability of heads equal to the likelihood of the node.
6. If the coin lands head, accept the sample, otherwise reject it and draw another node from the multinomial.
7. Repeat the above process, until the required number of samples are obtained and generate a plot of the posterior.

The samples should converge to the posterior distribution and the node h^* having the maximum probability is the maximum a posteriori (MAP) estimate. Thus hypothesis h^* is the inferred meaning of the examples observed.

Markov Chain Monte Carlo : After I got the model working with rejection sampling on the small hypothesis space (Figure 1), I decided to implement Markov Chain Monte Carlo (MCMC) sampling. This was because I knew that rejection sampling will not scale very well to the large hypothesis space in Figure 2. Following approach was used to implement MCMC:

1. Define the target distribution which is given by equation 5.
2. Define a proposal function and a proposal distribution.
3. Create a node map to map the nodes to numbers between $[0, numberofnodes - 1]$. This was done to facilitate sampling from the proposal and to easily capture dependence of new samples on previous states.
4. Set the initial state 'x' to 0.
5. Draw a node sample 'y' from the proposal function and check whether the sample lies in the valid range i.e., $[0, numberofnodes - 1]$. If not, set the probability $p = 0$.
6. If 'y' lies in the valid range, compute probability p as a product of the target ratio and the proposal ratio (as taught in class, refer to the code in the Appendix).
7. Then generate a random sample u from a uniform distribution and set the new state to 'y' if $u < min(p, 1)$, otherwise, set the new state to the previous state.
8. Repeat steps 5-7 until the required number of samples are obtained.
9. Remove the first 10,000 samples from the total samples obtained (i.e., $burn - in = 10000$) and use a lag of 50. Use the remaining samples to plot the posterior distribution.

Again, the samples should converge to the posterior distribution and the node h^* having the maximum probability is the maximum a posteriori (MAP) estimate. However, in this case none of the samples are rejected and hence the markov chain converges to the posterior distribution much earlier than rejection sampling. For example, in an experimental, I found that obtaining 50,000 samples required 50,000 iterations of MCMC, but 173,967 iterations of rejection sampling. Thus in this case, MCMC provided around 3.47 times speedup.

Comparison of Proposals : Since I was trying to estimate the posterior distribution through MCMC, the state space of my model was discrete and not continuous. In other words, the state space consisted of the set of nodes in the hypothesis space (Figure 2). This meant that my proposal needed to return discrete values. Hence I decided to split my proposal into a proposal function and a proposal distribution. Proposal function was used to propose a new state and proposal distribution was used to compute the density of the proposal for a given state.

Figure 3 explains the three different proposals that I tried. First, I experimented with them using the small hypothesis space (Figure 1). Figure 4 shows the results obtained for each of the three proposals with the small hypothesis space when only one example i.e., 4 is provided to the model. From the hypothesis space, you can see that 4 is the immediate child of node d, but node b and node a are also its ancestors. Thus node d should be the MAP and node b should have the next highest probability followed by node a. From Figure 4A, it is clear that the symmetric random proposal converges to the correct posterior distribution. Moreover it has a good mixing shown by the trace-plot and a good auto-correlation plot.

On the other hand, Figure 4B shows that the symmetric equally likely proposal doesn't converge to the target distribution. The reason for this is clear from its trace-plot which shows that the Markov chain gets stuck between states 0 and 1 (which corresponds to nodes a and b) and never reaches the state 3 (node d). Thus this proposal doesn't seem suitable for our target distribution as it carries the risk of getting stuck in a local subspace without being able to span the entire state space. Figure 4C shows that a variant of normal proposal also does not converge to an accurate posterior distribution, although it seems to be doing better that the equally likely proposal. There is no evidence of it being stuck in a local subspace in this case (it has good mixing), however it doesn't generate the correct probabilities over the nodes involved. Its auto-correlation plot also looks worse than the other two cases.

Based on the above analysis it seemed like that symmetric random proposal would be the best to use, however I also tried using all three of them with the large hypothesis space (Figure 2) to see how they scale. Figure 5 shows the results obtained when three examples (16, 17, 18) from subordinate category B are provided to the model as input. Figure 5A shows that with the symmetric random proposal, the model converges to the right posterior distribution with B having the highest probability followed by F and J which are its basic level and superordinate level categories respectively. Figure 5B and Figure 5C show that mixing and auto-correlation is still good with the large hypothesis space when using the symmetric random proposal. On the other hand, the equally likely and variant of normal proposals do worse with the large hypothesis space. Both of them get stuck on the starting state (Figure 5D, E). This is probably because the distance between the starting state (which is set to 0 or node 'LL' be default) and the acceptable states is too large that these two proposals are unable to get there. Hence, both these proposals pose a possibility of getting stuck in a local space without spanning the entire state space.

As a result of the above analysis, I decided to use the symmetric random proposal as my proposal function. I also tried to think about what other proposal function might be suit-

able for this application. Usually, we prefer to pick proposal functions that are close to our target distribution. However, for this application, the target distribution is dependent on the observed examples and thus will be different for every different set of examples (since the examples will have different hypothesis as their meanings). Thus, the proposal distribution needs to be very generic to be able to approximate all of these distributions. The way the node map (mapping of nodes to numbers from [0, number of nodes -1]) is constructed can also impact which proposal works better especially in case of dependent proposals (where the next state depends on the current state). I constructed the node map in a depth first way from the hypothesis space, such that the neighboring nodes belong to the same superordinate category and each node is closer to its children and its parent. However, this might not be the most optimal way to construct the node map. Regardless, since I ended up using he symmetric random proposal, the way the node map is constructed doesn't matter since this proposal function leads to independent sampling.

Results

All the results reported in this section were obtained through MCMC, using the symmetric random proposal function with the number of samples = 50000, burn-in = 1000, lag = 50.

Prediction results

As mentioned before, the main goal of the project was that the model should be able to estimate which hypothesis is the true meaning of word C when a set of examples of word C are provided to it.

The prediction results of the model on the small hypothesis space (Figure 1) are shown in Figure 6. You can see that the model predicts the correct category ('c') as the MAP estimate even when only one example from a subordinate category is provided to the model (Figure 6A). However its' confidence level is relatively lower and distributed across the basic ('b') and superordinate ('a') categories to which the example belongs. When we increase the number of examples shown to the mode to three (Figure 6B), the model becomes more confident of the subordinate category ('c') which the examples belong to. Similarly, the model predicts the MAP estimate with high confidence when three examples from basic and superordinate categories are provided to it as shown in Figure 6C and Figure 6D respectively.

The prediction results of the model on the large hypothesis space (Figure 2) are shown in Figure 7. You can see that in this case, when only one example from a subordinate category 'F' was provided, the model predicts the superordinate category 'EE' as the MAP estimate instead of 'F' which is predicted as the second most probable category (Figure 7A). However, if we increase the number of examples to three, then the model is able to accurately predict the subordinate category 'F' as shown in Figure 7C. Additionally, Figure 7B and Figure 7D show that the model correctly predicts the basic level ('J') and superordinate ('EE') categories when three examples of each of those categories are provided to the model. Note that I expected the posterior distributions to show a graded probability for the ancestors of the examples provided. For example in Figure 7C, I expected the probability of category 'F' to be the highest, followed by categories 'J', 'W', 'BB' and 'EE' (due to the structure of the hypothesis space). This trend was observed in the result, except for the probability of the superordinate category 'EE' which was higher than expected. This preference for the superordinate category 'EE' might be caused by a strong prior shown in the Figure 2 (recall that the prior is proportional to the branch length on the top of each node). It is important to point out that I constructed the hypothesis space by reading off the values from the graph in Figure 2, hence this difference might be caused due to inaccurate hypothesis space as well. Furthermore, in their paper (Xu & Tenenbaum, 2007) mention that the hypothesis space should be from [0,1] which is inconsistent with the hypothesis space constructed by them in the figure (which has an upper bound of around 0.925).

Generalization results

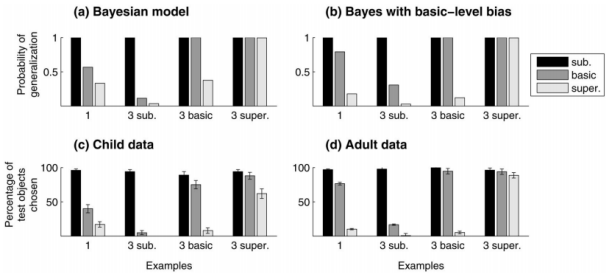

Figure 8: **a, b:** Predictions of the Bayesian model by (Xu & Tenenbaum, 2007) without and with basic-level bias respectively. **c:** Data from children in a psychological experiment by Xu & Tenenbaum (2007). **d:** Data from Adults in a psychological experiment by Xu & Tenenbaum (2007)

Another stretch goal of my project was that the model should be able to generalize what it learns i.e., given an additional example 'y', it should be able to decide whether 'y' belongs to the extension of C. In order to validate the generalization results, the $p(y \in C|X)$ computed from the model was compared to the generalization judgments of children and adults obtained through psychological experiments reported by Xu & Tenenbaum (2007). Note that this comparison was done by comparing graphs since I did not have access to the actual results for a more rigorous comparison. Moreover, I used the same examples which Xu & Tenenbaum (2007) used in their experiments, to get the experimental data.

Generalization results from the small hypothesis space (Figure 1) are shown in Figure 10. These results are averaged over two sets shown in Figure 9A, which cover this hypothesis space. For example to generate Figure 10A, one subordinate example was provided to the model from each case 1 and case 2 (column 2), and the results obtained were averaged. You can see that these results follow the same trend

A

```
def normal_pfun(sigma, mu):
    return int(sigma * np.random.randn() + mu)

def normal_pdist(x, mu, sigma):
    return ( 1/np.sqrt(2*np.pi*sigma**2) ) *  np.exp(-(x-mu)**2 / 2*sigma**2)

y = normal_pfun(sigma=sd, mu=x)
rproposal = normal_pdist(y, x, sigma=sd) / normal_pdist(x, y, sigma=sd)
```

B

```
def symm_pfun(x):
    if flip_coin(0.5):
        return x-1
    else:
        return x+1

def symm_pdist(x):
    return 0.5

y = symm_pfun(x)
rproposal = symm_pdist(x) / symm_pdist(y)
```

C

```
def symm_pfun():
    return np.random.choice(range(len(nodes)))

def symm_pdist(x):
    return 0.5

y = symm_pfun()
rproposal = symm_pdist(x) / symm_pdist(y)
```

Figure 3: Different proposal functions and proposal distributions used for MCMC. Each of A,B,C show proposal functions, proposal distributions and the way they are used in the MCMC algorithm (refer to the code in the appendix). **A:** A variant of normal proposal where the integer value of the sample drawn from a normal distribution is returned; the next state depends on the previous state (Metropolis Hastings). **B:** A symmetric proposal which is equally likely to propose one state higher or one state lower (Metropolis Algorithm). **C:** A symmetric random proposal which is equally likely to propose any state independent of the previous state (A combination of Metropolis Algorithm and Independence Sampler).

as in the experimental data shown in Figure 8c. The model shows graded generalization given one example, and more of all-or-none like generalization at the level of the most specific consistent hypothesis given three examples. However, quantitatively the probability values are different since we are just using a self-constructed sample hypothesis space in this case.

Generalization results for the large hypothesis space (Figure 2) shown in Figure 11, were computed in a similar manner using the data in Figure 9B, such that they covered examples from all the three main clusters (Vegetable, Vehicle and Animal). You can see that these results are similar to Figure 8c. The model captures most of the main qualitative and quantitative trends except that the graded generalization given one example is not accurate i.e., the probability of the superordinate category in Figure 11A is higher than that of the basic category, which is different from the experimental data. Earlier in the prediction results of the model, we saw that the superordinate category had higher probability than expected, and it seems that the discrepancy which we see here in generalization probability might be related to that. The results obtained by Xu & Tenenbaum (2007) with their model shown in Figure 8a for one example, are closer to the experimental data (Figure 8c) than my model. However, my results for three examples (especially for the basic level category i.e., Figure 11C) are closer to the experimental data than their results.

Sensitivity with respect to the prior

I tried two additional priors to study their effect on the generalization behaviour of the model.

Since the adults show a basic-level bias as discussed before, I added a bias in the prior that favors the three basic-level hypotheses (nodes 'J', 'R' and 'T'). The strength of the basic-level bias is a parameter which I set to $\beta = 40$ in order to fit the model results to the adult data shown in Figure 8d. On adding this parameter, the model captures most of main qualitative and quantitative trends in the adult data including graded generalization given one example (shown in Figure 12A). Note that similar to the predictions of the model with basic-level bias by Xu & Tenenbaum (2007) (Figure 8b), my model also predicts higher probability given three examples from subordinate category (Figure 12B), relative to the adult data shown in Figure 8d. However, my model seems closer to the experimental result than their model given three basic-level category examples (Figure 12C).

The second prior which I tried was an uninformative uniform prior and the generalization results obtained for the same are shown in Figure 14 in the appendix. The general qualitative trends can still be seen in the results but are much less pronounced. The quantitative results specially for the one example case are lost (do not match experimental data). This implies that the prior plays an important role in getting the same generalization behaviour as humans.

It was also found that the prior has the strongest effect

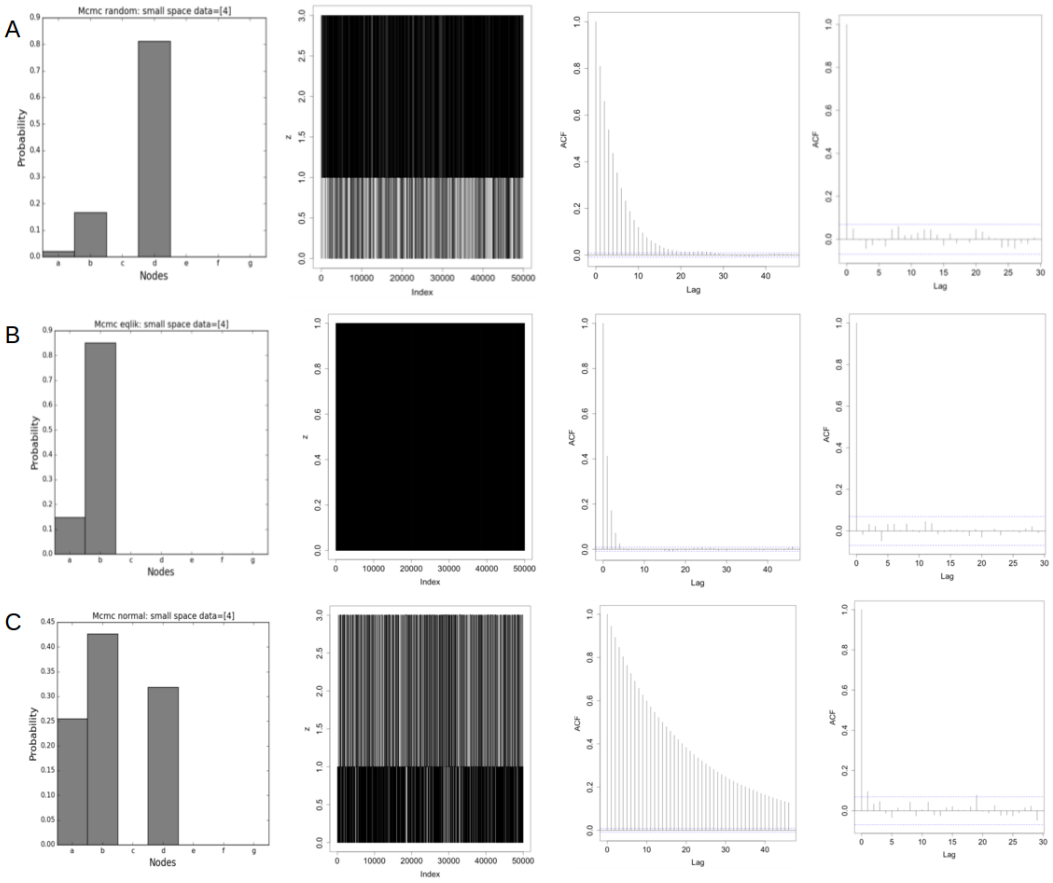

Figure 4: Results of comparing proposals using small hypothesis space. Posterior distribution, trace-plot, auto-correlation plot and auto-correlation plot after introducing lag are shown for the 3 proposals. Number of samples = 50,000, burn-in = 10,000 and lag = 50 for each of them. **A:** Symmetric random proposal. **B:** Symmetric equally likely proposal. **c:** Variant of normal proposal.

when only one example is observed. Figure 15 in the appendix shows the prediction results for the three priors:

- The original prior: Shows a preference for superordinate category (node 'EE') due to its' strong prior. The actual subordinate category (node 'F') gets the second highest probability.
- Prior with basic-level bias: Shows a basic level preference given one example (similar to adults) and predicts node 'J' to be the MAP with a high confidence.
- Uniform prior: Shows a graded probability distribution among the example's ancestors with node 'F' being the MAP as expected.

Discussion

We have seen the model was able to make accurate predictions and the results have shown that the model's behaviour is qualitatively and quantitatively similar to people's patterns of generalization when the right hypothesis space and prior are used. The similarity was higher in case of results for adults (model with basic-level bias included), than in case of children. This might be because the hypothesis space (Figure 2) was constructed using the similarity judgments from adults and its likely that children have a different hypothesis space which changes with experience (for example, it has already been shown that adults develop basic-level bias). Thus one of the limitations of this model is that it doesn't model the development of this basic-level bias with experience. I think it should be possible to model this as an instance of Bayesian learning such that Bayesian learner comes to realize that the basic-level object labels are used much more frequently than the subordinate or superordinate level labels. This would be an interesting area for future research.

We also saw the effect of priors on the model's predictions and generalization behaviour. We found that selecting the right prior is important for getting the same generalization behaviour as humans, and it affects the predictions of the model the most when only one example is provided. This is consistent with the fact that the word meanings learned previously can constrain the meanings of new words to be

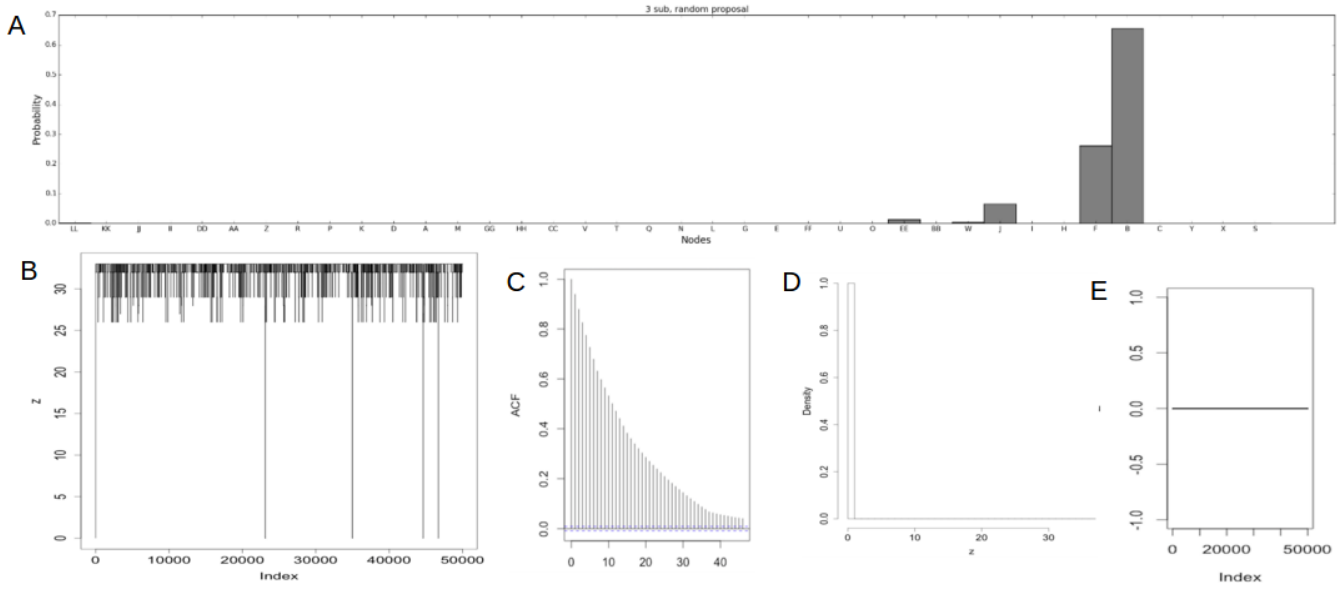

Figure 5: Results of comparing proposals using large hypothesis space. Number of samples = 50,000, burn-in = 10,000 and lag = 50 was used in each case. **A, B, C:** Posterior distribution, trace-plot and auto-correlation plot using symmetric random proposal. **D, E:** Posterior distribution and trace-plot using both equally likely and variant of normal proposals (both had same plots).

learned. However, I don't see any reason for a preference for a superordinate category which was found when using the original prior (equation 2). It is true that the superordinate categories like animal, vegetable etc. are highly distinctive, however there is no superordinate-level bias found in humans during word learning. Thus this preference might be due to the wrongly structured prior or because of manual error in reading the node heights as discussed before and needs to be further looked into.

Another possible area of future research is figuring out better proposal function which might perform better than the symmetric random proposal. Lastly, though the generalization results obtained were close to the experimental data, they might be improved by making the model more biologically plausible i.e., similar to how our brain works. This can be done by implementing the same Bayesian computations using a neural substrate such as the Neural Engineering Framework (Eliasmith & Anderson, 2004). Though the improvement is not guaranteed, this is an interesting avenue to explore in future.

References

Burns, B., Sutton, C., Morrison, C., & Cohen, P. (2003). Information theory and representation in associative word learning.

Callanan, M. A., Repp, A. M., McCarthy, M. G., & Latzke, M. A. (1994). Children s hypotheses about word meanings: Is there a basic level constraint? *Journal of Experimental Child Psychology*, *57*(1), 108–138.

Eliasmith, C., & Anderson, C. H. (2004). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press.

Fazly, A., Alishahi, A., & Stevenson, S. (2010). A probabilistic computational model of cross-situational word learning. *Cognitive Science*, *34*(6), 1017–1063.

MacWhinney, B. (1998). Models of the emergence of language. *Annual review of psychology*, *49*(1), 199–227.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive psychology*, *8*(3), 382–439.

Siskind, J. M. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, *61*(1), 39–91.

Smith, L. B. (2000). Learning how to learn words: An associative crane. *Becoming a word learner: A debate on lexical acquisition*, 51–80.

Xu, F., & Tenenbaum, J. B. (2007). Word learning as bayesian inference. *Psychological review*, *114*(2), 245.

Yu, C. (2005). The emergence of links between lexical acquisition and object categorization: A computational study. *Connection Science*, *17*(3-4), 381–397.

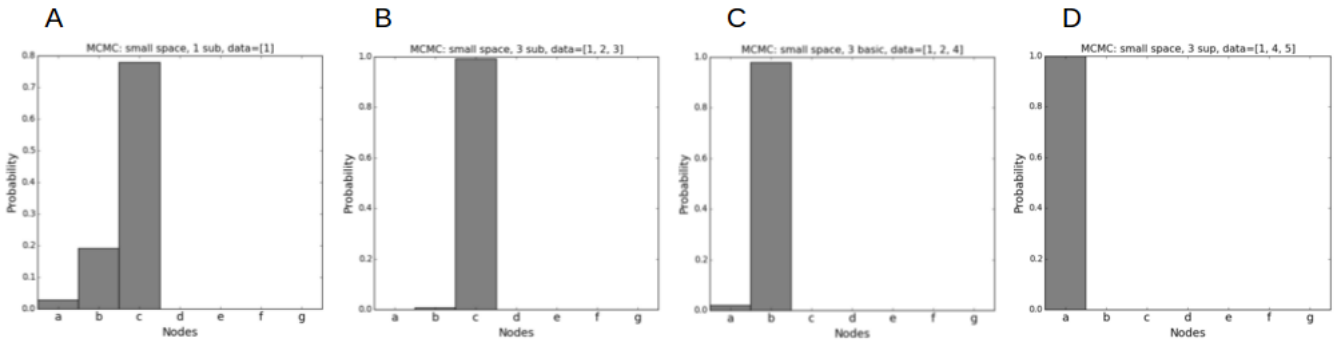

Figure 6: Prediction results for small hypothesis space. Each of the figures show the posterior distribution generated by the model when 1 or 3 examples from different categories are observed. **A**: Only one example (1) from subordinate category 'c' is observed. **B**: Three examples (1, 2, 3) from subordinate category 'c' are observed. **C**: Three examples (1, 2, 4) from basic level category 'b' are observed. **D**: Three examples (1, 4, 5) from the superordinate category 'a' are observed.

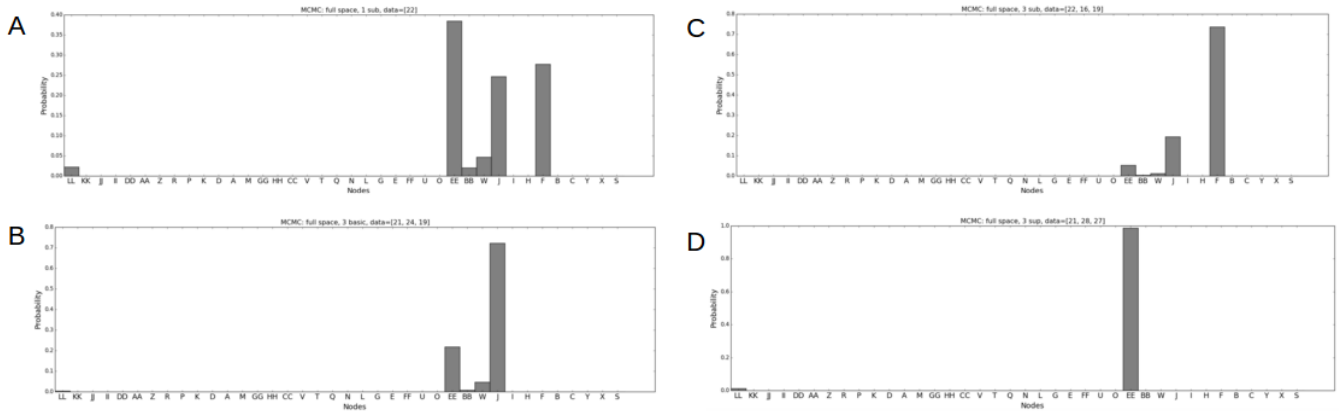

Figure 7: Prediction results for large hypothesis space. Each of the figures show the posterior distribution generated by the model when 1 or 3 examples from different categories are observed. **A**: Only one example (22) from subordinate category 'F' is observed. **B**: Three examples (21, 24, 19) from basic level category 'J' are observed. **C**: Three examples (22, 16, 19) from subordinate category 'F' are observed. **D**: Three examples (21, 28, 27) from the superordinate category 'EE' are observed.

A

	1 Subordinate	3 Subordinate	3 basic	3 superordinate
Case 1 (c, b, a)	[1]	[1, 2, 3]	[1, 2, 4]	[1, 4, 5]
Case 2 (g, e, a)	[7]	[7, 8, 9]	[7, 8 , 5]	[7, 5, 4,]

B

	1 Subordinate	3 Subordinate	3 basic	3 superordinate
Vegetable cluster (B, J, BB)	[16]	[16, 17, 18]	[16, 21, 22]	[16, 25, 26]
Animal cluster (A, R, JJ)	[1]	[1, 2, 3]	[1, 6, 7]	[1, 10, 11]
Vehicle cluster (E, T, HH)	[31]	[31, 32, 33]	[31, 36, 37]	[31, 40, 41]

Figure 9: Tables showing the input data used for generalization results. **A:** Input data for Figure 10. **B:** Input data for Figure 11. The categories in round brackets in the first column indicate (subordinate, basic, superordinate) categories for each set of data.

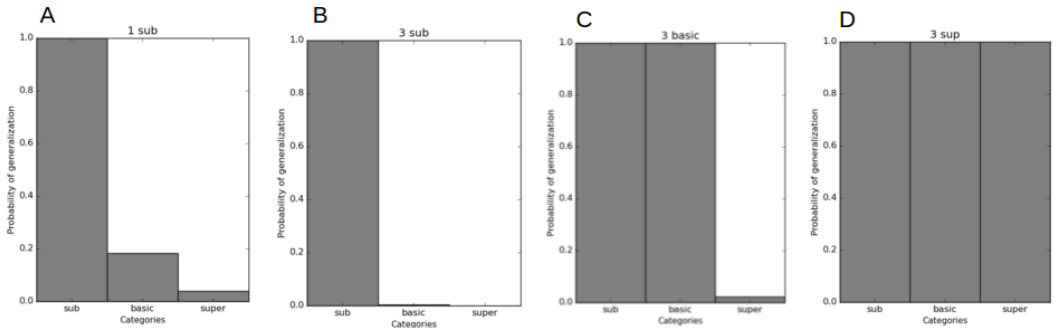

Figure 10: Generalization results for small hypothesis space. Each of the figures show probabilities of generalization when new example 'y' lies in the subordinate, basic or superordinate categories. **A**: Only one example from each subordinate category in Figure 9A is observed and the results are averaged over the categories. **B**: Three examples from each subordinate category are observed. **C**: Three examples from each basic level category are observed. **D**: Three examples from each superordinate category are observed.

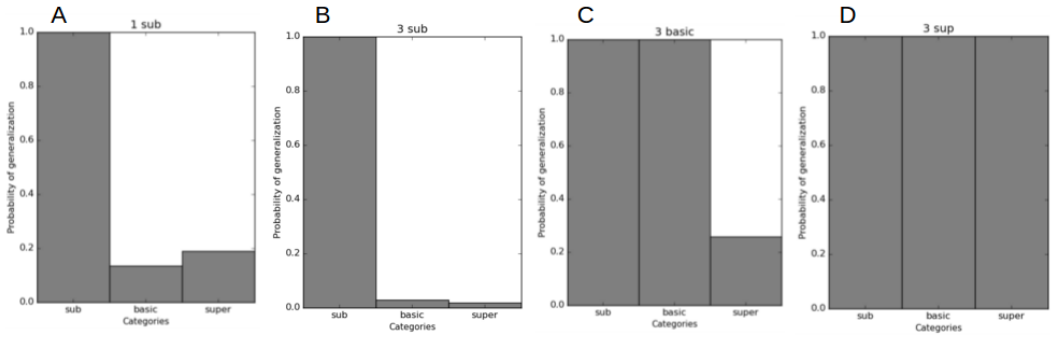

Figure 11: Generalization results for large hypothesis space. Each of the figures show probabilities of generalization when new example 'y' lies in the subordinate, basic or superordinate categories. **A**: Only one example from each subordinate category in Figure 9B is observed and the results are averaged over the categories. **B**: Three examples from each subordinate category are observed. **C**: Three examples from each basic level category are observed. **D**: Three examples from each superordinate category are observed.

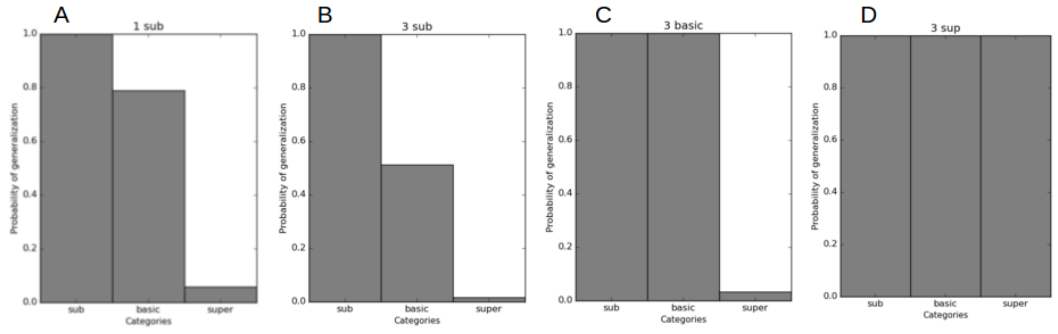

Figure 12: Generalization results for large hypothesis space with parameter β set to 40 (plots are generated in the same way as in Figure 11 with $\beta = 40$ instead of 1).

Appendix

All the code to implement this model was written using Python programming language. However, while comparing the proposal functions the results were output to numpy arrays which were read using the R programming language. This was done because it is easier to compare proposal functions in R language by plotting the trace-plots and autocorrelation plots. This can probably be done with Python too, however I was already familiar with doing it in R, so it saved some of my time. Both the python and R files are attached.

In order to generate the generalization plots, the results from each of the runs over the main clusters in the hypothesis space were written to python pickle files. The data stored in these pickle files was then averaged in a separate script and plotted. These pickle files and the python scrip to generate the plots from them are also attached. Additionally, two csv files were created to store the structure of the hypothesis spaces and these are also attached.

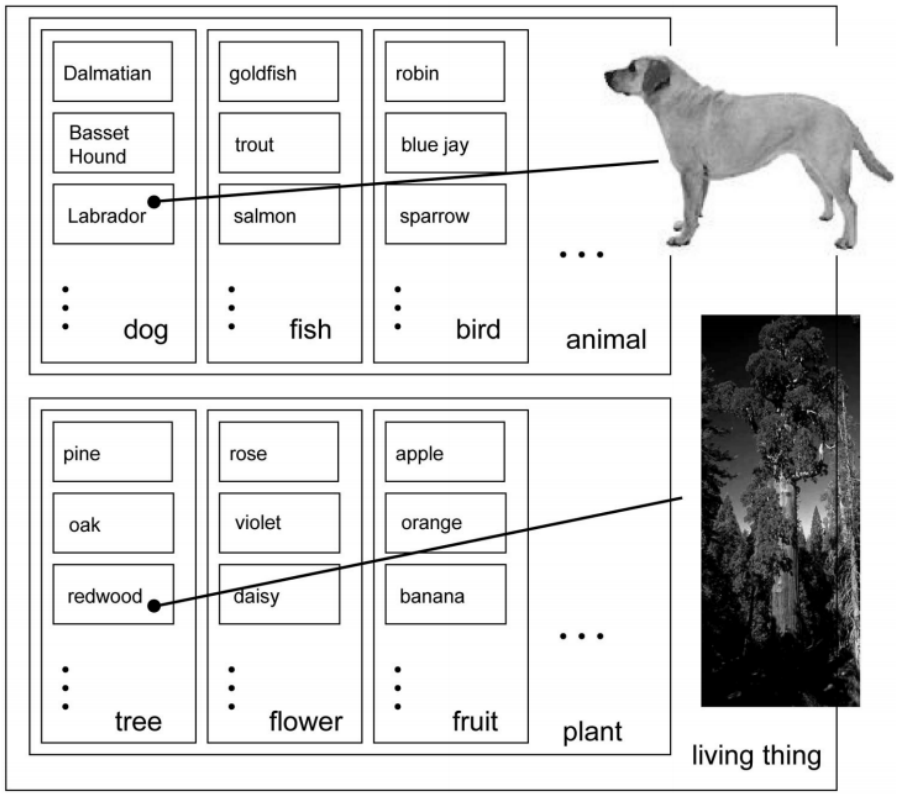

Figure 13: The extensions of words that label object-kind categories overlapping in a nested fashion, similar to a tree structured hierarchy of an object-kind taxonomy (Xu & Tenenbaum, 2007). Example: dog indicates a basic level category, animal is a superordinate category and dalmatian, labrador etc. are subordinate categories.

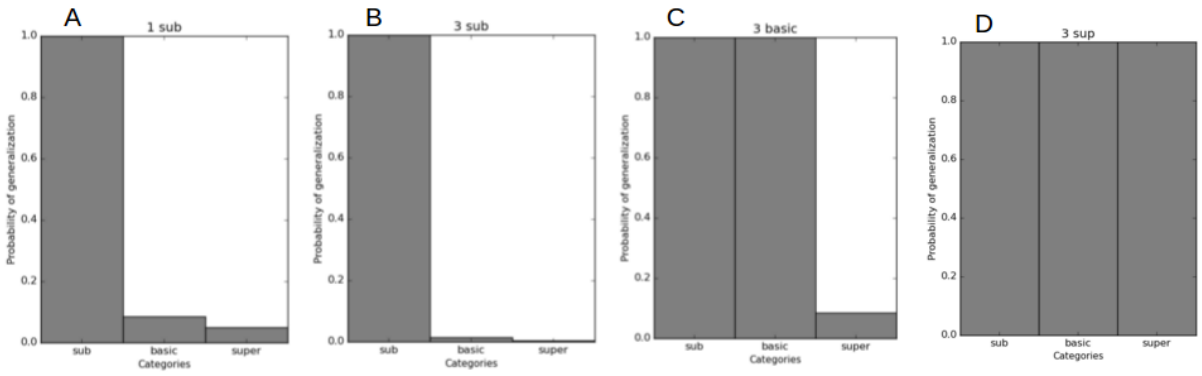

Figure 14: Generalization results for large hypothesis space with a uniform prior. Plots are generated in the same way as in Figure 11 but with a uniform prior.

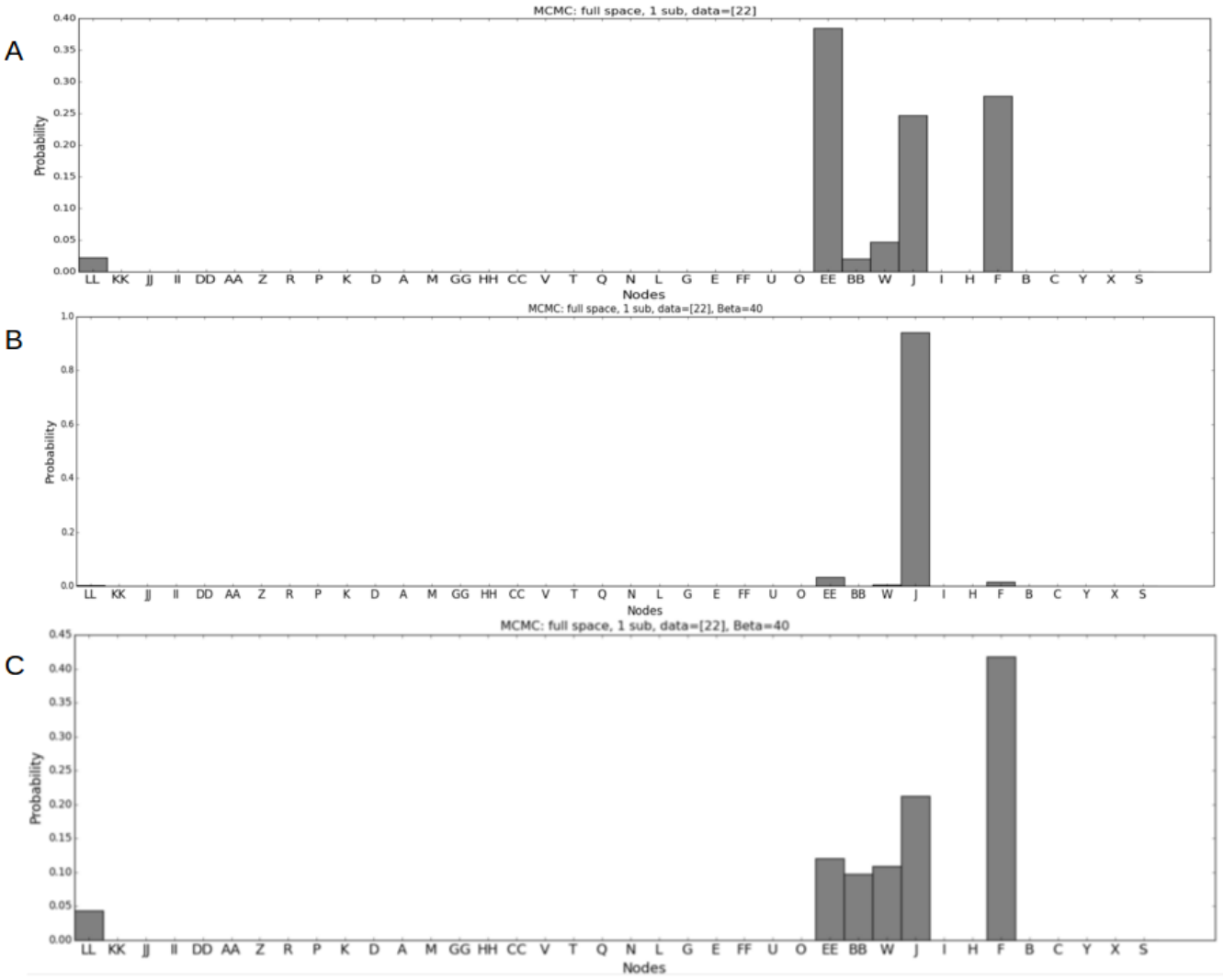

Figure 15: Prediction results for large hypothesis space for three different priors. Each of the figures show the posterior distribution generated by the model when only one example (22) from subordinate category 'F' is observed. **A:** Original prior defined in equation 2. **B:** Prior in A with basic-level bias ($\beta = 40$) added to it. **C:** Uniform Prior.