California Engineer

OSPREY

also in this issue

Semantic Priming   Flutes   Ultra-Wideband Technology

FEATURE ARTICLE

Also in this issue:

California Engineer

Student Journal of the UC Engineering
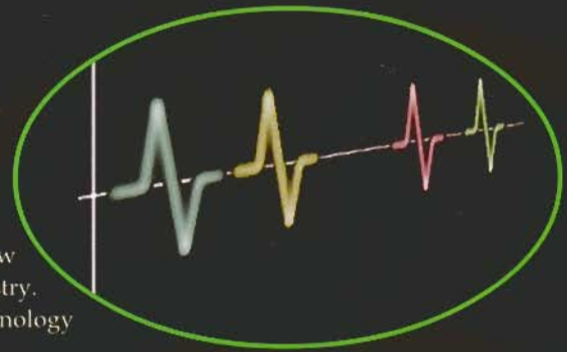
learning categories using

Semantic Priming
in a Bayesian Framework

by Ray Luo

Human beings are capable of learning nouns with just a few examples and almost no explicit feedback. But a simple explicit memory model cannot account for either the rapidity or the accuracy of learning. Thus, reliance on an implicit memory model that accounts for the effects of past experiences is necessary. One approach is to represent nouns as basic categories with uncertain feature values. As more data are incorporated, the categories become more and more like what the nouns represent in the real world. Since categories are built up from experience, it is expected that physical interaction will improve category acquisition. Current research suggests that implicit category comparisons (via semantic priming) model this physical interaction.

Much of the modeling ideas presented here are inspired by Schooler et. al. [5]. As in Schooler et.al., rational analysis [1] is applied to the modeling of results from psychological experiments. The REM model of Schooler et. al. was an attempt to account for episodic retrieval via a model that assumes that priming acts "alter the word's lexical-semantic memory representation" [5]. Justification for direct alteration of the representation

is given in Schooler et. al. Most objections to Bayesian models point to the lack of plausibility for the human mind to perform probabilistic computations during experimental tasks. As such, this experiment will make simplifying assumptions on the REM model by using simpler Bayesian methods that are assumed to be within reach of the neural computation.

The experiment focused on animal categories so as to simplify direct comparison with psychological experiments. Ideas suggested here are relevant for text categorization as well. In particular, any domain that can be decomposed into hierarchical categories can be modeled. The algorithm uses a simple representation of features and categories to facilitate comparison. In a real-world application, Bayesian network representations should be constructed for both basic and parent categories. A suggestion on how this can be done is presented later, although the implementation is only at an experimental phase.

Assumptions

First, it is assumed that no perceptual errors are made during learning and testing. There is no probability distribution associated with input animal features for each example. The lack of errors in perception does not, however, imply that the input data is noiseless. It is assumed that the distributions of attributes of an animal are independent and jointly Gaussian. Data are generated using normal distributions with pre-determined means and variances that reflect the attribute domain for each animal. Noisy examples with random features are also fed into the model periodically. The assumption suggests that the model captures the only probabilistic aspect of category learning. Thus, the model itself (not the perceptual system) is responsible for handling noisy data.

Second, only two levels of categories are relevant to the task. The basic cat-

egories consist of animal types. In particular, data for ant, snail, frog, cat, wolf, cow, and whale were given as input. The model forms abstract models of the data, so no actual label is given to any category. The idea is that a category is described by average feature values and a measure of the spread of feature values. It is the model's task to minimize the complexity of sets of categories while maximizing the probability of correctly identifying the category of any example given the feature values. This can be done using inference on a belief network. The parent categories consist of graded conceptual categories involving the animal types. For example, the large animal category should include whale and cow as salient examples, with wolf possibly having graded membership. Graded membership is not represented explicitly, however. The model will keep track of one best prototype, which can be updated with a given probability during comparison tasks. Conceptual categories are constructed using the different features associated with animals. In particular, large, small, cute, hideous, ferocious, and tame animal were used. Ideally, other objects could be modeled as well, allowing the conceptual categories to be simply large, small, cute, hideous, etc. This makes the model more general, less domain-dependent, but harder to assess.

Third, inferences and comparisons done on particular members of basic categories are necessary for learning and structuring both basic and parent categories. Here the focus is on feature value comparisons. A question that can be

asked is, for example, "which animal is larger, cat or frog?" Luo [4] reviews these tasks. In modeling early development, the semantic priming that results from querying the subject is associated with changes in the underlying categories. The claim is that these changes result from comparison of basic category feature values with prototype feature values. In the example above, size feature values for cat and frog are compared with the prototype large animal feature value. Changes are made to the abstract categorical representations of cat and frog size values and spreads. Prototype values are updated as needed. One natural question that arises is: if feature value estimates are already kept by our basic category representation, could not one just compare the estimates and return the appropriate category (in this case, cat for the larger animal)? There are three problems with this approach. First, the approach predicts that all comparisons should take the same amount of time on

average. This has been shown to be false [2]. For example, it is much faster to decide between whale and ant than cat and frog. Second, the approach predicts that no errors could occur. For example, frogs would always be judged smaller than cats. This does not hold experimentally [4]. What is necessary is a model that is generally true, but can give wrong answers in some contexts. Third, there are uncertainties associated with feature estimates and this is why a measure of spread is needed. As more examples of cats are shown to the system, the model becomes better at estimating the relative size of cats, but it is never absolutely sure that cats are smaller than wolves. The approach is to compare cat and wolf sizes with the prototypical size. Relative distances between category and prototype size feature values can then model differences in judgment reaction times. This approach can be thought of as a variance reduction technique. Knowl-

edge about prototypical cases, which also have associated uncertainty values, is leveraged.

Fourth, there exist abstract category representations in the mind independent of but influenced by linguistic labels. This model keeps abstract categories with no particular linguistic interpretation. This model of the mind relies on generalization over similar word senses. Hence, frog and toad may refer to relatively similar animals that have similar values with respect to the set of features. A child may not be able to distinguish (at least at first) between these similar animal types. Thus a structure is needed that represents the features of a frog-toad without labeling it. It is possible then to compare these abstract categories amongst themselves and perform inferences purely on the basis of feature structures. Words with different senses (cat, for example) are described by different categories while different

words denoting animals with similar features (e.g. frogs and toads) are described by the same category. In the future the linguistic label associated with an abstract category can, hopefully, be shown as a probability distribution over words or phrases acquired from experience. Instead of saying, for example, that the word cat has two senses (kitty and tiger), it'll be possible to say that abstract category A has a probability of 0.95 of being labeled "cat" and abstract category B has a probability of 0.5 of being labeled "cat." With experience, the abstract categories become finer because the variances of the feature values become smaller. In the limit of infinite experience, the two word senses of "cat" will be described by separate abstract categories with labeling probability 1.0, because a separate abstract category will describe "tiger" with probability 1.0. It is also hoped that someone will prove the following claim: given a pre-defined,
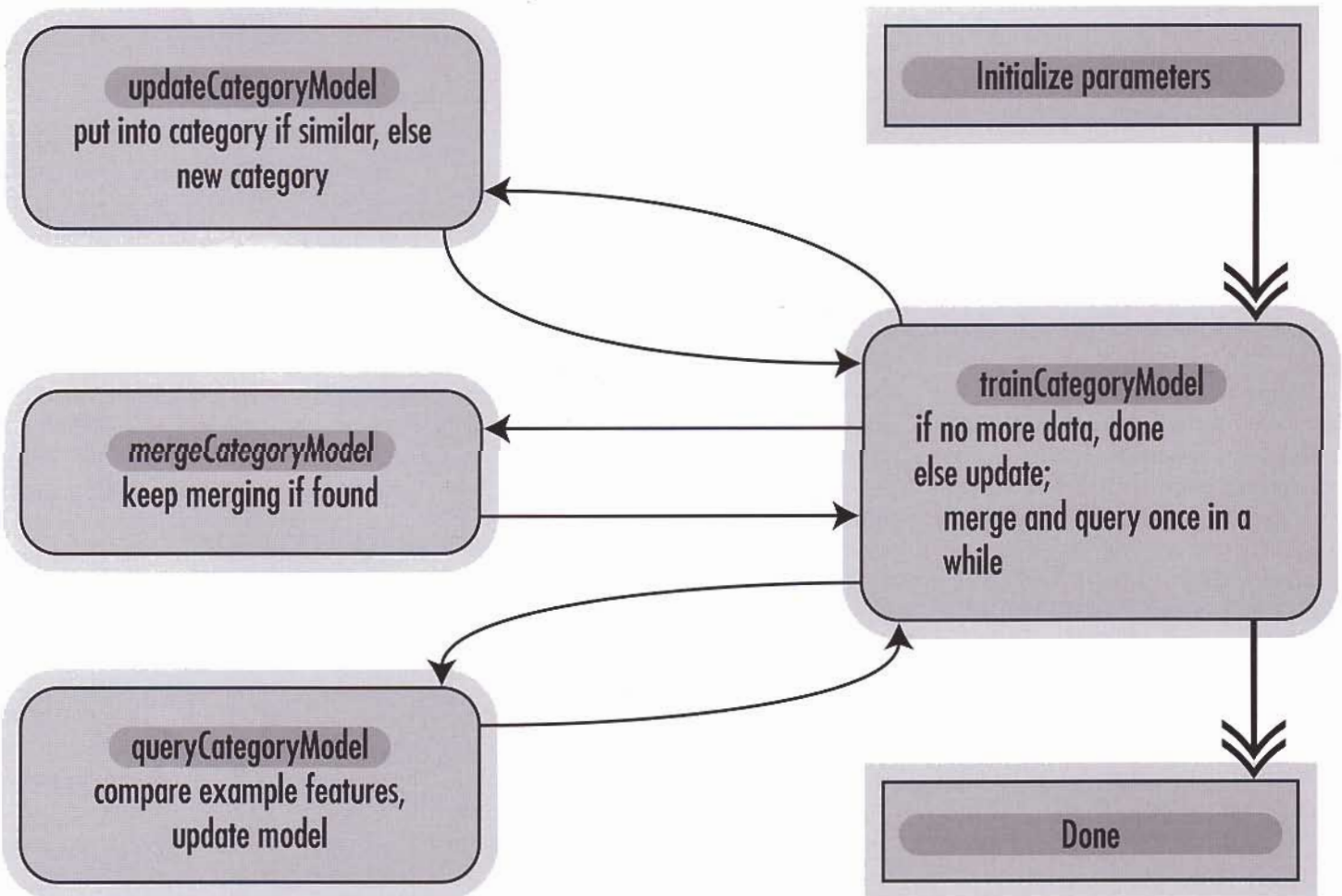
finite, sufficiently "nice" set of training examples, the category partitions formed by the algorithm will converge in probability to a set of non-overlapping category distributions (in the sense of zero variance feature values) as the number of sweeps goes to infinity. This statement would then imply that each category would eventually be labeled by its own word. For now, it is assumed that most likely labels are found magically by some other mechanism. It is not necessary to worry about possible ambiguity here because the comparison simulations are done with abstract categories and the prototypes generalize over the entire domain independently of particular word labels. Note, however, that a prototype is a single instance from the basic categories. When humans are confronted with a question regarding size they don't automatically think of a whale. The assumption here is that humans perform subconscious comparisons with a vague superordinate prototype whose variance is averaged over a few salient examples.

Lastly, it is assumed that this direct representation of human semantic processing can be efficiently implemented neurally. One approach is to translate the mean-variance representation into a simple Bayesian network. This will be described after the model and the algo-rithm.

Model

The basic claim of the model is that the meaning of a noun in some restricted domain is its abstract representation in relation to all other nouns in the domain[1]. Moreover, this relationship is captured by comparisons with sets of superordinate prototypes that capture general characteristics of objects referred to by basic terms.

The model acquires basic category representations by merging abstract models of category partitions and by performing inferences and comparisons using these abstract partitions. It is assumed that implicit reinforcements for correct and incorrect responses have been provided internally. This model controlled experimental paradigms in which the subject is given a pair of objects whose feature values fall on some subjective scale. Errors are frequently caused by priming and going too fast. Moreover, learning probability distributions for category labels is assumed to be incremental, so it is not necessary to tell the subject not to say something, because it will not be said as predicted.

Thus, the model has abstract categories that model the internal representations of nouns and learning mechanisms that takes knowledge from the real world, and reconstructs, as best as they can, the entities that the nouns represent, using comparison-based implicit learning [5]. Note that for the domain presented here, this consists only of two hierarchical levels. Extensions beyond animal types can be constructed by treating each superordinate category as also a basic category.

Algorithm

Both model merging and prototype formation depend on order of data presentation. Since any possible merge with lower cost will be performed, there is a chance that the optimal merge will not be performed. Similarly, since early experience is weighted more heavily in prototype formation, it is possible to get unrepresentative prototypes fairly late into the learning process.

Prototype acquisition is modeled as "Markov." That is, the mean feature value for a prototype is drawn from either the current value or the maximum (or minimum) value over subordinate categories. As the model accumulates experience, it tends to stay with the current value; initially, it tends to choose the optimal value. The idea is that a single basic category instance serves as the prototype, but that the variance associated with the prototype is estimated from the current variance and the difference between the optimal value and the current value. Hence, the model forgets the past because the current prototype value variance captures all that is needed to know about past prototype values. The reason for this design decision is the seemingly transient nature of a prototype. When people think of a cat, they are basing their estimates on prior knowledge about distributions of feature values of a cat. When people think of a large animal, however, they first think of the idea of large before thinking of a prototype. People do not know everything about this prototype animal but they do know that it is large. After using this information, the prototype is put away. The claim here is that not all the feature value distributions of any prototype of any superordinate category are known. Given this assumption, one sensible model would be to base the proto-

Left: High-level overview of the algorithm. Right: A Bayesian network implementation.
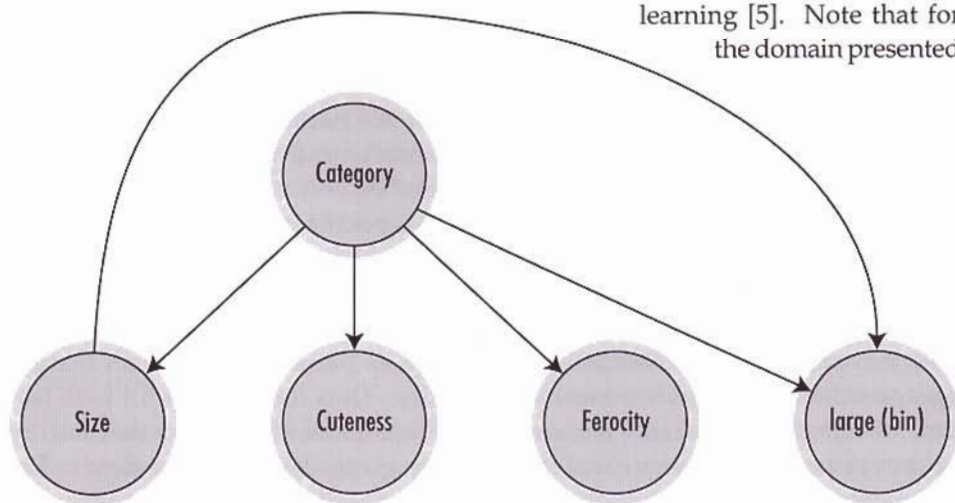
type on either the last prototype or the best possible prototype. The probability of choosing between the two depends on experience. As more and more examples are observed, it becomes less likely to switch to the best possible prototype, because the current prototype must already be pretty good.

Here is a walkthrough of the algorithm in detail. First, random examples are created consisting of size, cuteness, and ferocity values for our animal types (ant, snail, frog, cat, wolf, cow, and whale), with noisy examples given random features values. Next, the data is given to the model incrementally. The model first scans through the categories to see if the new example can be put into an existing category. If not, a new category is constructed. After a few data incorporation steps, the merging algorithm is invoked. The cost of every pair of possible merges is computed and the best merge to perform is chosen. The first merge is performed with a higher probability (lower cost) than the probability

category feature values. The idea is that for a set of data, a smaller model is a better, more succinct description of an underlying representation. Therefore, it is desirable to minimize the size of the partition by merging categories together. But merging requires the re-computation of the variance of feature values for the resulting category, which can increase because the two categories to be merged are distinct. This should be kept low if good predictions are to be made about which category a novel set of feature values belongs to. Thus there is a trade-off between compactness and representational power. The algorithm will keep finding and merging categories until a merge with a lower cost cannot be found.

Between data incorporation steps, the query mechanism is periodically invoked, which models implicit comparisons that facilitate category learning. The prototypes are first updated probabilistically by taking on the parameters modeling the experience of a child. A past experience factor (pef) between 0

egories that correspond to the querying examples are found. These categories are compared with the prototype categories for each feature. The categories are explicitly changed by a small amount that is related to pef. The larger category is moved toward the value of the larger prototype, the smaller category toward the value of the smaller prototype, and similarly for cuteness and ferocity. The idea is that given a pair of examples, the model is asked to choose, for example, which animal is larger. The target example is coded as being large and the rejected example as being small. This primes future judgments on both of the examples [4]. Specifically, the internal representation of the target and rejected examples are changed by some small factor determined by experience and by the distance between feature values of examples and prototypes. Note that the prototype representation can also be changed if the target example feature value is "more optimal" than the prototype feature value. This can happen be-
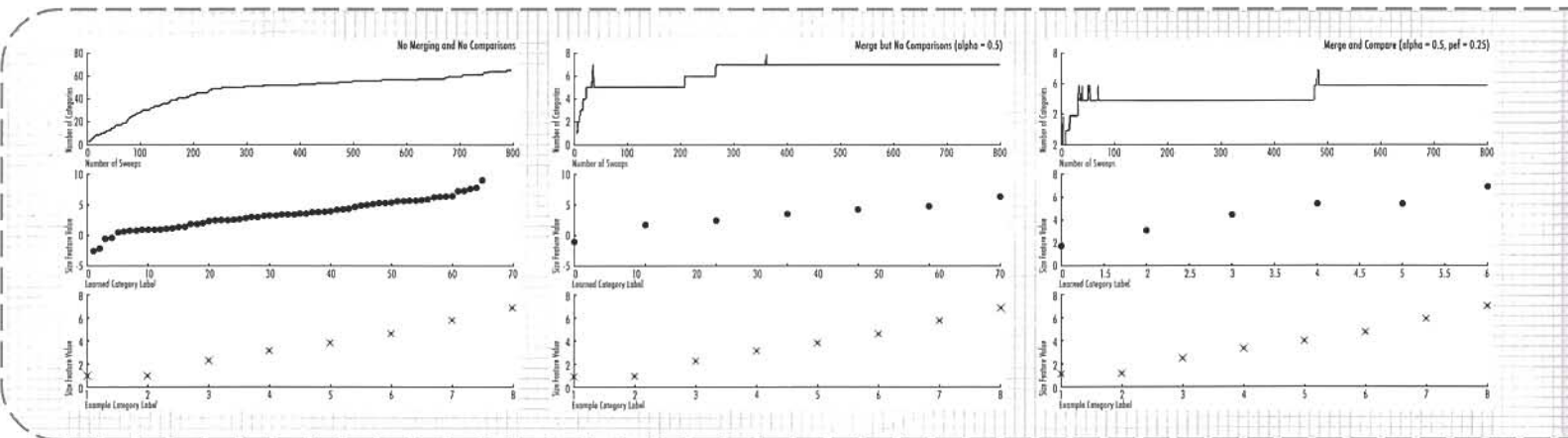
(cost) of the original model. The basic idea is to try and maximize

$P(Model \mid Data)$ which is equivalent to $P(Data \mid Model) \cdot P(Model) / P(Data)$.

Here $P(Data)$ can be ignored since it is the same for any merge. This equation is then translated to:

$cost(Model)$ being equal to
$\alpha \cdot size(Model) + (1 - \alpha) \cdot var(Model)$,
where $size(Model)$ is the total number of categories (i.e. the size of the partition), and $var(Model)$ is the total variance of all

and 1 is defined that is initially small, but becomes larger as the number of sweeps increases. With probability 1 minus pef, the superordinate prototypes are replaced (large animal, small animal, cute animal, hideous animal, ferocious animal, tame animal) with the example with the optimal feature value in the category partition. Since pef increases with time, the model is less likely to change prototypes as it obtains more and more experience. Next, the best matching cat-

cause the prototypes are only updated probabilistically so the prototypes may not be the best.

It is possible to translate the model to a Bayesian network. As suggested by Koller & Sahami [3], the categories can be represented as discrete nodes that serve as parents of Gaussian feature nodes. Thus the statistics for each feature are specified and query the category node given the feature evidence. The problem is to implement an unsuper-

vised learning algorithm that updates the values that the category node can take on as more evidence arrives. Here, independence of each feature value is assumed. The superordinate category node "large" is a binary representation of a "decision" based on current feature estimates. Note that prototype values are kept implicitly within the distribution for the superordinate node. It may be better to use decision nodes for the superordinate category nodes, but whether inference would work is still in question.

Results

In general, category partitions learned using feature comparisons tend to generalize a bit more, so that a smaller number of categories results. Note that one obvious problem with the model merging algorithm is that it tends to give inaccurate values for animals whose size is small.
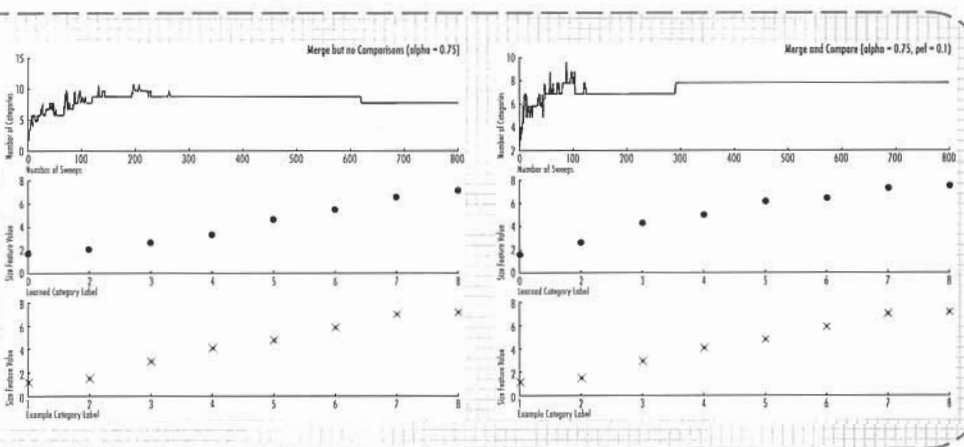
It is possible to change the parameters of both the model merging and the feature comparison algorithms. Over many iterations of the experiment, it is discovered that feature comparison tends to do wonders when pef is set to be small (R 0.1). In that case, feature comparison tends to generalize more than model merging alone.

Discussion

This Bayesian model-merging algorithm applied to category-acquisition is not without its faults. For example, it is

assumed that in the presence of small sample sizes, humans still believe that the distribution of feature values in the real world is approximately normal for each distinct animal. More reasonable approaches involving the Dirichlet density can be found in Anderson [1]. It is also assumed that each feature comparison task occurs regularly and that all features of a category are compared. In the real world, this is rarely the case. A person might be led into comparing, for example, sizes more than cuteness, and cats and dogs more often than polar bears and lady bugs. A proper model for comparison tasks in early language acquisition would involve modeling the world in which the child lives, which is beyond present scope and capabilities. Note also that the independence of each feature given the category is assumed, which cannot be a true model of the world. Finally, the model assumes that model merging and feature comparison happen independently. Note, however, that depending on the current comparison task, it might be more or less suitable to merge disjoint categories. For example, consider the comparison of cardinals and blue jays. The model does not have much of an idea of what they are and might consider them to be close to each other in size. Are they also in one category?

One other proposal would be to treat the problem as minimization of Boolean complexity given features and examples. The best model is that which minimizes

the minimal formula associated with the disjunctive normal form of the evidence written in conjunction with features and disjunction among examples. This has not been worked out yet.

The Bayesian network representation still needs to be explored thoroughly. A way in which learning of category values will work for a Bayes net still needs to be worked out. Superordinate category representation can also be improved. One idea is to use separate nodes for each category value (ie. animal name). Then all the model has to do is prune the network as more evidence arrives. The problem is that each node would be the parent of the feature values. Hence each augmentation of a feature adds $O(k)$ links, where k is the number of values a category can take on. The model should also allow feature values to link with each other, thereby removing the feature value independence assumption.

This model has presented a Bayesian approach to learning category representation. It can be seen that learning is facilitated by implicit interactions with the real world. Here, the focus is on feature comparison and priming. The principle, however, is more general. In modeling domain knowledge, the various contexts in which a name can occur and the various interpretations the name can take on with respect to superordinate category prototypes need to be considered.

Merge but no Comparisons (alpha = 0.75)

Merge and Compare (alpha = 0.75, pef = 0.1)

References

1. Anderson, J. R. 1991. The adaptive nature of human categorization. Psychological Review. 98(3):409-429.

2. Banks, W. P.; Fujii, M.; and Kayra-Stuart, F. 1976. Semantic congruity effects in comparative judgments of magnitude es of digits. Journal of Experimental Psychology: Human Perception and Performance, 2:435-447.

3. Koller, D. and Sahami, M. 1997. Hierarchically classifying documents using very few words. In Proc. ICML-97, 170-178.

4. Luo, R. 2000. Response interference in categorical judgment tasks. Online http://inst.eecs.berkeley.edu/~rluo/categ_learn_doc/perc&cog_np&sc-revised-text.doc.

5. Schooler, L. J.; Shiffrin, R. M.; and Raaijmakers, J. G. W. 2001. A Bayesian model for implicit effects in perceptual identification. Psychological Review, 108(1):257-272.

Matlab code for the algorithm can be found at http://inst.eecs.berkeley.edu/~rluo/categ_learn_doc/categ_learn.zip.

Run the file learn_animal_categories in Matlab to begin experimenting. You should have the BNT toolkit installed http://www.cs.berkeley.edu/~murphyk/Bayes/request.html