Title
Learning salience amoung [sic] features through contingency in the CEL framework

Authors
Granger, R. H., Jr.
Schlimmer, Jeffrey C.

Publication Date
1985

Peer reviewed

Learning Salience Amoung Features Through Contingency in the CEL Framework

R. H. Granger, Jr., and Jeffrey C. Schlimmer
University of California, Irvine

Technical Report No. 85-28

November, 1985

Learning Salience Among Features Through Contingency in the CEL Framework

Richard H. Granger, Jr.
Jeffrey C. Schlimmer

Irvine Computational Intelligence Project
Computer Science Dept.
Cognitive Sciences Program
and
Center for the Neurobiology of Learning and Memory
University of California
Irvine, California 92717

Abstract

Determining which features in an environment are salient given a task, *salience assignment*, is a central problem in Machine Learning. A related phenomenon, *contingency* (the conditions under which relative salience among environmental features is acquired), is central to learning and memory in animal psychology. This paper presents an analysis of a set of empirical data on contingency and an algorithm for the salience assignment problem. The algorithm presented is implemented in a working computer program which interacts with a simulated environment to produce contingent associative learning corresponding to relevant behavioral data. The model also makes specific empirical predictions that can be experimentally tested.

This research was supported in part by the Office of Naval Research under grant N00014-84-K-0391, the National Science Foundation under grant IST-81-20685, and by the Naval Ocean Systems Center under contract N66001-83-C-0255.

1 Introduction

A rat in a laboratory cage hears a tone. It also hears the air conditioning system start, and sees a lab assistant taking notes. Shortly afterwards, it feels an unpleasant electric shock. What does the rat learn? Since the late 1960s, psychological experiments have made it clear that what the rat learns from above episode depends on the relationship, over several trials, among the several plausible cues to the unpleasant event. In Machine Learning research in Artificial Intelligence, this corresponds to a *salience assignment* problem: which of the many possible cues are the predictive or salient ones, i.e., the ones to be learned? Rats solve the salience assignment problem under constraints more severe than those faced by most AI systems. For instance, learning must be *incremental*, for the rat in a natural setting must make good use of the experiential data already gathered while gathering more. Moreover, the environment may not provide perfect predictors; the animal must make predictions as best it can when cues indicate only a change in the probability of an event.

CEL (Components of Experiential Memory) [Granger 1983, Granger and McNulty 1984] is a framework for explaining a variety of learning phenomena. It consists of twelve processes, or operators, and a set of representations based on sequential traces of events. The operators are separable, by function, into five classes: reception, recording, retrieval, reconstruction, and refinement. Reception operators translate external perceptions into internal representations. Recording operators fix those representations into either short-term or long-term memory stores. Retrieval operators match incoming representations against traces stored in long-term memory and select one for the reconstruction operators to perform. Refinement operators modify existing long-term memory traces to accurately reflect learned concepts.

Using the CEL framework, we present a method of determining which features of an event are predictive of others and distinguishing useful cues from context and background noise. The method determines the relevance of individual predefined features and forms new feature descriptions by conjoining or negating existing features. The effectiveness of the method is due to taking into account, in addition to successful predictions and errors of commission, also errors of omission and events in which the absence of a cue correctly prevented prediction of a second event. This extension to the idea of strengthening and weakening corresponds to the distinction in psychology between learning based on number of pairings and learning based on contingency. Using this method the CAP-CEL (Contingent Associative Processes in Components of Experiential Learning) program exhibits contingency-based learning behavior, modeling the learning behavior of animals and humans in classical conditioning tasks. The program is able to function correctly even with a large number of erroneous training instances.

2 Contingency

2.1 The data

Imagine again a rat attempting to decide, over trials, which of several environmental features or events should be learned to be a predictor of a recurring shock event. We can view the animal's task as hypothesizing potential causal relationships between the shock and various features, individually and in various combinations, and weighing these relations against each other.

We might initially assume that a particular feature or event would be inferred to be the predictor of the shock depending on the number of times that feature actually occurred immediately before the shock. Each time the feature is paired with the shock, the 'association' between the feature and the shock might be *strengthened* (see e.g.,[Anderson 1983]). Extensive experimental evidence in the psychological literature shows this to be false. Over time, a particular feature, say the tone, may be paired with the shock more often than is some other feature (e.g., light). However, this condition is not by itself sufficient to warrant the animal's inference that the tone is more likely to predict the occurrence of shock. In particular, even if it happens over a number of trials that the tone precedes the shock more often than the light precedes the shock, say 7 times versus 4 times, *but* the shock also occurs a large number of times without having been preceded by the tone (say 8 times) while the shock only rarely occurs without the light (say 3 times), then an animal will learn to predict that the light, and not the tone, is the better predictor of the occurrence of shock. Hence, the naïve idea that the number of pairings alone determines the predictiveness (or salience) of candidate predictive features, or that strengthening alone could be the mechanism for learning associations, is false.

This result requires a somewhat counterintuitive computation on the part of the animal: the animal must be computing the probability of the shock occurring given the light and given the tone. Rescorla's [1966, 1967, 1968] formulation of the necessary computation is that the probability of the shock outcome (the US) given the conditional stimulus feature (the CS, e.g., the tone) must be greater than the probability of the outcome occurring without that feature having occurred, or $p(US|CS) > p(US|\overline{CS})$.

This measurement of relative probabilities is referred to as *contingency*; animals, and humans in analogous circumstances, exhibit *contingency-driven learning* in the sense that they somehow maintain incrementally-updated knowledge of the relative predictiveness or *salience* of features. Through experience the animal must pick out the relevant (salient) features from the background of uncorrelated features and use only these salient features to predict future events.

2.2 Required computation

We term the problem described here as *salience assignment*, i.e., the differential assignment of predictive value to the candidate predictive features (or combinations of features). All that the animal has available to it as input from the environment is the presence of features sensed over time. What it must compute from these inputs is the relative probability of some features relationships to others over time.

There are four logical categories of these relationships that can be computed: positive predictions, negative predictions, uncorrelated cues, and context; each has a behavioral correlate in animals. First, there are two types of what we term *predictive* cues: positive and negative predictions. Positive predictive cues are those that accurately predict the occurrence of an outcome. Negative cues accurately predict the absence of an outcome (e.g., these 'safety signals' might predict that the shock will not follow the cue, and therefore that the animal need not fear its coming). *Uncorrelated* cues are irrelevant and therefore not necessary for prediction of an event. Finally, an animal cannot readily evaluate the predictive importance of a *context* cue, i.e., one that occurs constantly in the background of a training session. It is impossible to know whether such a cue is a necessary precondition for predicting a shock, unless the shock has been predicted a few times in the absence of the context cue. When this happens, either the context cue will become a positive predictive cue (if the prediction was successful), or an uncorrelated cue (if the prediction failed).

The inputs to be categorized as positive, negative, context or uncorrelated are occurrences of features. For simplicity, we can categorize the logically possible pairwise combinations of two feature events F1 and F2: either F1 occurs and then F2 occurs (*prediction*), F1 occurs and then F2 does not occur (error of *commission*), F1 does not occur and then F2 does occur (error of *omission*), or F1 does not occur and neither does F2 (*non-prediction*). The first and last of these combinations strengthen the predictive value, or association, between F1 and F2, while errors of commission and of omission weaken the association.

	F2 present	F2 absent
F1 present	++ Prediction	+− Error of Commission
F1 absent	−+ Error of Omission	−− Non-prediction

Table 1: Possible combinations of F1 and F2

The proposed algorithm for the calculation of contingent predictiveness essentially just keeps a running count of each of these four categories of pairwise events;[1] these counts are used to calculate an estimate of the likelihood that one of these two features predicts the other.

3 An algorithm and implementation for contingent learning

3.1 The basis of the algorithm: sufficiency and necessity

Bayesian statistics (see e.g., Duda et. al. [1979]) provide formulae for the calculation of two values in inductive logic: *Logical Sufficiency (LS)*, which indicates the extent to which the presence of one event predicts, or increases the expectation of, another particular event; and, reciprocally, *Logical Necessity (LN)*, which represents the extent to which the *absence* of an event *decreases* expectation or prediction of the second event. LS and LN are each calculated by a simple formula composed of precisely the four possible categories of pairwise feature occurrence given above.

$$\mathrm{LS} = \frac{s(n+o)}{o(s+c)} \qquad\qquad \mathrm{LN} = \frac{c(n+o)}{n(s+c)}$$

where s is the count of successful predictions, c is errors of commission, o is errors of omission, and n denotes non-predictions.

Our proposed algorithm makes use of the calculation of LS and LN values to categorize the relationships between a pair of cues. The categorization is based on the interpretation of LS and LN values. LS values range from 0 to ∞, with high LSs corresponding to a feature F1 strongly predicting a second feature F2, (since high LS implies a high ratio of successes to errors of commission); and very low LSs corresponding to the case where F1 implies that F2 will *not* occur (low ratio of successes to commissions). Hence, for a high LS value, F1 is a positive predictor of F2; for low LS, F1 is a negative predictive cue,

[1]Although a non-prediction will *only* be considered to happen when either F1 or F2 has been predicted and then neither occurred. This is because all non-predictions would otherwise give rise to a huge, ongoing number of counts. Hence, in this algorithm, non-predictions are systematically 'undercounted'.

i.e., the presence of F1 predicts that F2 will not occur. An LN value near 1 indicates that the absence of a cue may be ignored, while a low LN value (near zero) indicates that a presence cue is quite necessary for prediction. When the value of LS is approximately 1, i.e., neither very high nor very low, then the cue F1 may be either a context cue or uncorrelated. In such a case, if there are more errors of commission than of omission, i.e., more failed predictions than unexpected shocks, then the cue is categorized as a context cue, since it is often present, but often fails to predict the shock; yet the shock doesn't often occur in its absence. If there are more errors of omission than commission, however, then the cue is categorized uncorrelated.

Positive cue	$ls >> 1$
Negative cue	$ls << 1$
Context	$ls \approx 1, omissions < commissions$
Uncorrelated	$ls \approx 1, omissions \geq commissions$

3.2 Gathering evidence

All counts in CAP-CEL's memory are initially 1. These counts are updated only when an index node (corresponding to a feature complex) is triggered by matching cues in the environment, at which point one of the relevant long-term memory traces organized below this index node is chosen for reconstruction; i.e., the trace contains predictions of what will happen and which behavior is associated with these predictions.

From this point on, that trace is continually matched against new events as they occur. When a prediction succeeds, the success scores of matched features in the environment are incremented. Cues failing to match receive incremented omission scores.

When a prediction fails, each cue feature that matched the environment scores a commission; each cue feature that was absent from the environment, a non-prediction. Novel features present in the environment are added with an initial score of 1 commission, 1 prediction, 1 omission, and 1 non-prediction.

3.3 A detailed example

Assume CAP-CEL is in a situation where tones, lights, noises, and shocks are occurring. CAP-CEL's task is to construct a memory record which will allow it to predict the occurrence of the shock accurately (presumably in order to avoid it). Specifically, given a situation where the shock is reliably preceded by a conjunction of features (e.g., tone and light), a *positive* contingency, a table representing a portion of CAP-CEL's memory about the shock will look similar to table 2. (Note that successes are indicated by '++', commissions by '+−', omissions by '−+', and non-predictions by '−−'. The figures in table 2 are taken from runs of an early version of our computer model.)

The LS (logical sufficiency) value indicates the degree to which a cue is sufficient to cause expectation of a result feature, with values greater than 1 indicating a positive contribution to expectation. The LN (logical necessity) value indicates the degree to which absence of a cue precludes expectation of a result feature. An LN value near one indicates that absence of a cue may be ignored, while an LN value near

	++	+−	−+	−−	LS	LN
Cage	52	11	1	1	1.65	0.35
Tone	52	7	1	5	5.29	0.14
Light	52	8	1	4	4.33	0.17
Buzz	19	4	34	8	1.02	0.91
Whrr	43	10	10	2	0.97	1.13
And[Tone,Light]	48	3	1	5	5.65	0.07

Table 2: Positive Contingency

zero indicates that a cue is quite necessary for expectation. The conjunction of light and tone has been proposed by the CAP-CEL program itself (see discussion in section 3.4).

This chart illustrates important differences between contingency learning and more intuitive notions of strengthening based on number of pairings. Cage and tone receive the same number of pairings with shock, but tone is a much better predictor of shock. Moreover, tone was involved in a greater number of mistaken predictions (errors of commission) than was buzz, but tone is still recognized as the better predictor.

3.4 Combining features

Learning in complex environments necessitates noting useful combinations of features. For instance, a conjunction of a tone and a light may indicate a shock while neither the tone nor the light alone do. CAP-CEL uses current associations between features to form combinations of features.

The introduction of new feature combinations is failure driven. When CAP-CEL makes an error of commission, a new clause[2] may be introduced from among the predictive clauses. A clause with a low LS value acts as a negative predictor; if satisfied in a negative instance, it is a candidate for negation. A pair of clauses with low LN values act as required positive predictors; if one is satisfied in a negative instance and the other is not, they are candidates for conjunction. We can think of LS and LN values as guidance for a plausible move generator searching through the space of conditions.[3]

Propose	When error of commission
Not[A]	$LS(A) << 1$, A satisfied
And[A,B]	$LN(A) << 1$, A satisfied $LN(B) << 1$, B unsatisfied

CAP-CEL does not expand its representation of clauses without bound. Two mechanisms serve to limit this growth. The first is simply that new clauses are only introduced following a failure. In an

[2]We will use the term *clause* to refer to both individual features and combinations of features in the discussion that follows.

[3]It is desirable to allow both discrimination (through conjunctions of clauses) and generalization (through disjunction). At this time, however, the CAP-CEL model proposes only conjunctions and negations. A weaker form of generalization is achieved by dropping clauses with low predictive value.

environment without erroneous training instances this alone can be quite effective.

Secondly, CAP-CEL utilizes competition between a newly introduced clause and its components. When a new clause outperforms its components, the latter are deactivated and are isolated from retrieval processes. CAP-CEL measures the performance of clauses by comparing LS and LN values. When a new clause is introduced it is assigned an LS *threshold*[4] set at the maximum of the LS values of the components. The competing component clauses are deactivated when the LS value of the new clause exceeds its threshold. Until this time the new clause cannot be combined with other clauses.

Utilizing this mechanism has the additional advantage that CAP-CEL can correct some erroneous clause formations. When a clause falls below its threshold, the clauses that led to its formation are reactivated and now compete with the ineffective clause. Each reactivated clause is assigned a threshold that is the counterpart of its rival; if the ineffective clause has an LS threshold, each reactivated clause will have an LN threshold set at the LN value of the ineffective clause. When either reactivated clause surpasses its threshold[5], the ineffective clause is deactivated and the clauses are free to form new combinations.

3.5 Organization of indexed memory

The algorithms described here for the incremental calculation of LS and LN and the use of those values to successfully categorize feature cues are grounded in the operation of an indexed network memory. Long-term memory in CAP-CEL is organized as a network of nodes and links accessed via parallel matching processes. Besides providing a degree of parallel processing our model provides a principled method of limiting the spread of activation during retrieval, by *diagnosticity* of tests applied during activation.

Links in our model are simple one-way transmission channels capable of communicating a single non-symbolic value (magnitude) between nodes. These are categorized according to the interpretation of the signal they carry as one of four types of signal: probe links, trigger links, expectation links, or confirmation links. Nodes are of two types: feature or feature combination nodes (henceforth nodes) and intermediate nodes (henceforth internodes) which record relationships between feature nodes.

If a feature F1 is indexed as a cue which may lead to an expectation of feature F2, then an internode will lie between node F1 and node F2, as illustrated in figure 1. Long term memory traces which record occurrences of both F1 and F2 are accessed via the internode between them. Feature F1 may be triggered by incoming sensory data and in turn send a signal to I1, which adjusts the signal strength based on the predictive value of F1 for F2, and passes it on.

Suppose that F1 matches an item in short-term memory but F3 does not. Further suppose that F1 is moderately suggestive of F2, F3 is quite necessary for an expectation of F2, but F4 is not highly correlated with F2. Activation from F1 will trigger internode I1 which will scale expectation by LS and pass it to F2 along the expectation link. F2 will then send signals along each of its outgoing probe links. I3 receives a probe and, finding the LN value relating F3 to F2 is quite small (i.e., the presence of F3 is quite necessary for any expectation of F2), I3 sends a probe signal to F3. Since F3 does not return a signal along a trigger link, I2 sends an inhibitory signal along the expectation link to F2, reducing the expectation of F2. (Values are multiplied together by the receiving node; hence, the LN value which has

[4]An LS threshold is chosen since a more restrictive clause is falseness preserving and thus LN is guaranteed to be as good.

[5]LN thresholds are satisfied if the LN value falls below the threshold.

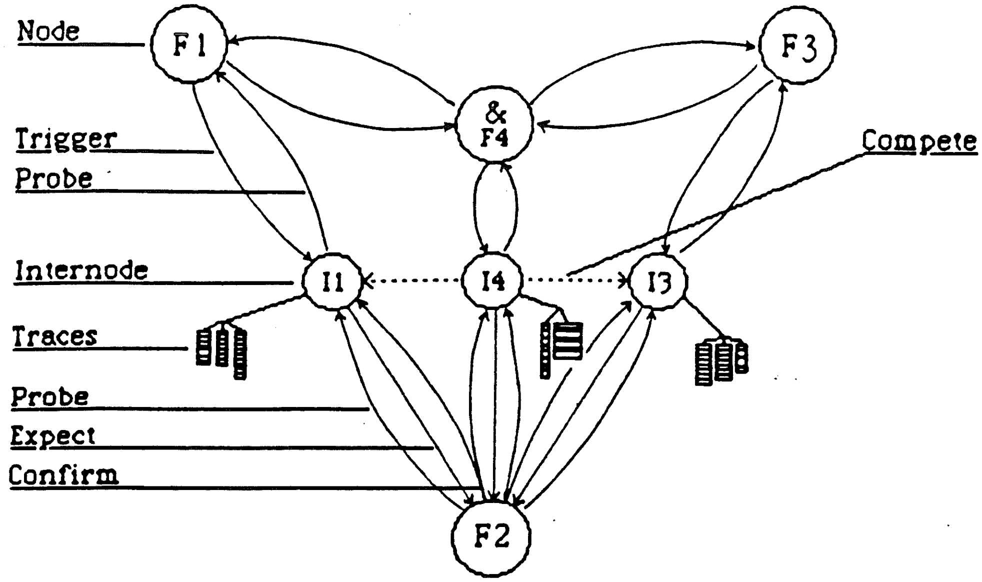

Figure 1: An example memory index structure.

a value less than one acts as an inhibitory signal). I4 also receives a probe signal from F2. Since the LN value relating F4 to F2 is about 1 (indicating that F4 is not strongly correlated to F2), I4 does not send a probe signal on to F4 and the spread of activation is attenuated.

Feature nodes may also represent combinations of features. Node F4 is triggered when F1 and F3 are both present, so F4 functions as the boolean AND operator. If F4 is triggered by either F1 or F3, it sends probe signals to both. When it is triggered by both, it triggers I4. Other boolean combinations can be similarly represented. The dashed lines in figure 1 indicate the competition between the recently introduced I4 internode and the component internodes I1 and I3.

The principle advantage of the index network as described above is that *test diagnosticity*, as expressed by LS and LN values stored in internodes, provides natural control for parallel retrieval processes; the spread of activation is limited in a principled way by these tests on links between nodes, rather than by a pursuit strength decay of links over time. The number of nodes needed to represent complex patterns is moderate, and the network can be modified incrementally as new relations between features are discovered.

3.6 Experience with the CAP-CEL system

3.6.1 Robustness

Real-world environments invariably entail some degree of noise, so a learning engine must be able to tolerate erroneous training instances. We have been pleasantly surprised by the performance of the CAP-CEL program in these circumstances. Figure 2 depicts the performance of CAP-CEL when trained with various rates of erroneous instances.

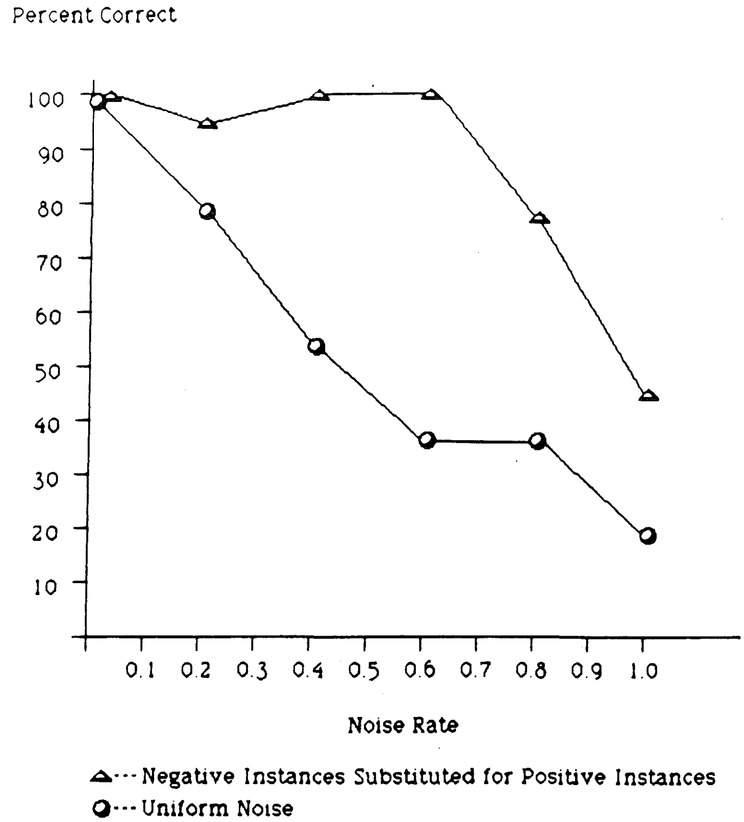

Figure 2: Performance of CAP-CEL as a function of noise.

The line plotted with circles in figure 2 shows performance under conditions of 'uniform' noise, that is, positive instances have been substituted for negative instances and vice versa, with equal likelihood. As the error rate approaches 0.3, the CAP-CEL's performance falls toward a chance level (50%). As one would expect, error rates in excess of .5 cause CAP-CEL to acquire the opposite of the training concept and to perform at less than chance level.

The triangles in figure 2 plot the performance of CAP-CEL when negative instances are substituted for positive instances, but there are no spurious positive instances. This is similar to partial reinforcement in conditioning. CAP-CEL's performance remains well above chance in this case even for levels of noise in excess of 50% because the tone and light are in positive contingency relation with shock even when the absolute probability of shock following the cues is low. CAP-CEL's level of performance is similar when the only noise is positive instances substituted for negative ones.

CAP-CEL's tolerance of erroneous training instance is partly due to the smooth weighting functions LS and LN. In addition, though, we found that robustness depends critically on the introduction of combined features (section 3.4). With no noise in the data, CAP-CEL can achieve perfect performance for simple conjunctive classifications even when the combination proposer is disabled, since the extreme values of LS and LN are sufficient to express logical necessity and logical sufficiency. But when even a small number of erroneous instances are introduced, performance falls off precipitously unless combinations are proposed. The predictive value of a combination of features is higher than that of any of its component features, and the influence of erroneous instances on that value is correspondingly less.

3.6.2 Annotated run-time output of the CAP-CEL program

We have implemented CAP-CEL in Franz Lisp on a VAX 11/750 running under Unix. In the following transcript, CAP-CEL learns that the conjunction of tone and light is a positively contingent cue for the onset of shock. Annotations are separated from actual program output by semicolons.

⋮

```
Detecting: cage, light, whrr, buzz      ; These are external cues.
    strongly expecting shock (odds = 7.81 >> 1).
Detecting: nothing                      ; CAP-CEL predicts the shock,
Updating expectations                   ;   but doesn't get one.
    marking commissions                 ; Satisfied cues get a commission.
    marking non-predictions             ; Unsatisfied cues get a
                                        ;   non-prediction.
Deactivating clause:                    ; This clause isn't proving
    and[whrr,light]                     ;   effective.
Introducing new clause:
    and[light,tone]

        ; The light cue is satisfied (present) in this instance
        ; and has a LN << 1.  The tone cue is not satisfied
        ; (missing) and also has a LN << 1.  Their conjunction
        ; is suggested as a new clause.
```

⋮

```
Detecting: cage, light, tone, whrr
    strongly expecting shock (odds = 31.49 >> 1).
Detecting: shock
Updating Expectations
    marking successes
    marking omissions
Establishing clause:                    ; This clause is now above its
    and[light,tone]                     ;   threshold and its rivals
Deactivating clause:                    ;   are deactivated.
    light
Deactivating clause:
    tone

shock predicted by:
    ----------------------------------
Pattern                               | ; Long-term memory looks like
    ++  +-  -+  --      ls   ||   ln  | ;   this now:
    ----------------------------------+
and[light,tone]              ||       |
```

```
     8   1   1   3      3.56  ||  0.15  |
    ----------------------------------+
buzz                              ||        |
     5   2   9   7      1.27  ||  0.65  |
    ----------------------------------+
whrr                              ||        |
    11   7   3   2      1.02  ||  0.97  |
    ----------------------------------+
cage                              ||        |
    13   8   1   1      1.24  ||  0.76  |
    ----------------------------------
```

3.6.3 Explanation of program behavior

Confidence in CAP-CEL is calculated by multiplying together the LS values of each satisfied feature and the LN values of each unsatisfied feature. This confidence measure is then interpreted in terms of odds: much less than 1 indicates that F2 is not expected; about 1 indicates uncertainty; much greater than 1 indicates that F2 is expected.

CAP-CEL introduces new clauses only on errors of commission and avoids endlessly making proposals when it reaches proficiency. Possible new clauses are proposed from the satisfied and unsatisfied features on the basis of LS and LN values. In the example above, light has an LN value below well below 1, as does tone, and an error of commission occurs where light is present and tone is absent; CAP-CEL hypothesizes that light *and* tone together might be a better predictor than either alone.

4 Related work

Each development and refinement of the CEL framework is driven by an attempt to accurately model experimental data. The model of contingency described in this paper were motivated by results from animal behavior experiments. Animal learning theorists, notably Rescorla and Wagner [1972], Wagner [1978], Mackintosh [1975], and Pearce and Hall [1979], have proposed models of associative learning which account for contingency. These theories predict the strength of associative learning as a function of experience, but do not describe the detailed processing necessary to form and modify associations. Our model is intended to supply this further level of detail.

Other researchers have also formulated systems for the purpose of modelling experimental data. One of the most comprehensive models of learning behavior is the ACT* family of programs were developed by John Anderson [Anderson 1983]. ACT* uses productions systems as a framework for describing the processing underlying complex behavior. ACT* creates new production rules through processes of composition, proceduralization, generalization, and discrimination. Of these, generalization and discrimination address the problem of discovering which features are relevant for determining when an operator should be applied. ACT* creates a generalized rule by omitting a condition from the antecedent part of another rule. Discrimination adds a new clause either to the antecedent or to the consequent part of a rule. Rules in ACT* are strengthened when they are reinvented and when they are activated through the spread of activation in memory. They are weakened by negative feedback.

The ACT* framework has been used to account for a wide variety of data from the psychology of human learning. The scheme for strengthening and weakening rules, however, does not appear to be consistent with the basic psychological data concerning contingency.

Another cognitively oriented Artificial Intelligence program employs learning while parsing English stories: IPP [Lebowitz 1983]. Stories read by IPP are used to form groups of story features that frequently occur together. These groups of features provide top-down direction for IPP's parsing mechanism. The number of features in common between two groups of features is used to determine when a new feature group should be introduced. These feature groups are strengthened and weakened by a unary amount given positive or negative evidence. Individual features within a group that increase expectation of other features in the same group are termed *predictive*; those that do not are termed *predictable.* The degree to which a feature is predictive is based on how infrequently it has occurred in feature groups.

IPP successfully improves its parsing of texts within the domain of newspaper stories. However, the mechanisms for determining predictiveness do not appear to be consistent with the findings of contingency experiments. Secondly, as Lebowitz notes, IPP has some difficulty when a set of features in a group are predictive while any single one isn't. Lacking a representation for explicit conjunctions prohibits IPP from assigning predictiveness only to a set of features.

Researchers in cognitive psychology have also outlined constraints for mechanisms that attempt to account for experimental results. Barsalou and Bower have expressed three concerns over the formulation of memory models: elimination of test contingency, parallel traversal, and storage requirements [1984]. The first is the characteristic of some memory models that allows or inhibits retrieval of items stored in a network based on the success of some distantly related test. Barsalou and Bower argue that partial matching should be allowed in memory retrieval and that there should be a meaning related ordering of tests if any. Secondly, experimental evidence that human memory is parallel have lead Barsalou and Bower to criticize inherently serial models of memory. Thirdly, Barsalou and Bower doubt the validity of memory schemes that require exponential memory space, for human memory systems do not appear to be limited in the ways predicted by such schemes.

CAP-CEL addresses each of these concerns in that it employs a memory scheme which allows partial matching (long-term memory traces are retrievable by any of the predictive cues in that trace), a parallel retrieval scheme, and a memory containment scheme which displays a growth bounded on the average by n^2 (where n is the number of novel stimuli in the environment) since there are at most n^2 pairwise associations. In the most pathological case, explicit representation of boolean combinations requires a bound of 2^n.

5 Conclusions

The importance of contingency is well known in animal learning theory, and the extensive experimental data concerning contingency provide a clear set of computational requirements for a process model of learning. We have expressed this set of requirements algorithmically as a *salience assignment* problem, and we have shown how this problem is solved within the CEL framework, via formulae based on Bayes' algorithm. The CAP-CEL program demonstrates that the account we have offered is in fact adequate to distinguish useful, predictive cues from context and uncorrelated cues. Moreover, the performance of

CAP-CEL comprises a set of detailed predictions of our hypotheses that may be confirmed or rejected on the basis of experiments.

We have had some limited success in integrating these findings into the complete framework of CEL. Specifically, scaling the memory model up in CAP-CEL requires that any clause can be the result of a given association as well as the predictor. This is likely to occur quite naturally given the current memory model. However, filtering memory traces so that they contain precisely the information represented in the memory clause nodes may involve some difficult computation. Knowing that a clause is satisfied by the set of features represented in a trace is easy; determining which features lead to satisfaction of that clause and which ones are spurious is a considerably more difficult problem.

There is a wealth of empirical behavioral data waiting to be accounted for. We intend to continue to concentrate on basic phenomena of animal learning rather than following the current Artificial Intelligence and cognitive science fashion of building computer models of complex human problem solving tasks; we believe these basic phenomena shed more light on the fundamental properties of learning in humans as well as animals.

References

1. Anderson, John R. 1983. *The Architecture of Cognition.* Cambridge: Harvard University Press.
2. Barsalou, L. and Bower, G. 1984. Discrimination nets as psychological models. *Cognitive Science,* **8**, 1–26.
3. Duda, R. O., Gaschnig, J. G., Hart, P. 1979. Model design in the Prospector consultant system for mineral exploration. In *Expert Systems in the Micro-electronic Age*, D. Michie (ed). Edinburgh: Edinburgh University Press, 1979.
4. Granger, R. H. 1983. An artificial intelligence model of learning and memory that provides a theoretical framework for the interpretation of experimental data in psychology and neurobiology. Department of Computer Science Technical Report 220, University of California, Irvine.
5. Granger, R. H. and McNulty, D. M. 1984. Learning and memory in machines and animals: An AI model that accounts for some neurobiological data. *Proceedings of the 6th Annual Conference of the Cognitive Science Society*, Boulder, Colorado.
6. Lebowitz, M. 1983. Generalization from natural language text. *Cognitive Science,* **7**, 1–40.
7. Mackintosh, N. L. 1975. A theory of attention: Variations in the associability of stimulus with reinforcement. *Psychological Review,* **82**, 276–298.
8. Pearce, J. M. and Hall, G. 1980. A model for Pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological Review,* **87**, 532–552.
9. Rescorla, R. 1966. Predictability and number of pairings in Pavlovian fear conditioning. *Psychonomic Science,* **4**, 383–384.
10. Rescorla, R. 1967. Pavlovian conditioning and its proper control procedures. *Psychological Review,* **74**, 71–80.
11. Rescorla, R. 1968. Probability of shock in the presence and absence of CS in fear conditioning. *J. Comparative and Physiological Psychology,* **66**, 1–5.
12. Rescorla, R. and Wagner, A. R. 1972. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. In *Classical Conditioning II: Current Research and Theory,* A. H. Black and W. F. Prokasy (eds). New York: Appleton-Century-Crofts, 1972.
13. Wagner, A. R. 1981. SOP: A model of automatic memory processing in animal behavior. In *Information processing in animals: memory mechanisms,* N. E. Spear and R. R. Miller (eds), Erlbaum, Hillsdale, NJ, 5–47.