Bayesian Models of Memory Retrieval

Richard M Shiffrin
IPAM
July, 2007

- ‘What is a Bayesian model’ is not a completely well formed question.
- Bayesian modeling is a matter of probabilistic induction, but most research in cognitive science is aimed at induction of the processes of cognition.

- In this sense, almost any probabilistic model – a model that assigns a probability to data, or to statistics based on data– is Bayesian, because we typically use those assigned probabilities to judge model fitness, and to make inferences about models and hypotheses.

- Although induction by assigning probabilities to observed data is one hallmark of Bayesian induction, it is not the only hallmark.
- From where do we get the models that are to be assessed in this framework?

- Creative thinking about the data-- One intuits a model (or hypothesis) to explain or predict the data or the data pattern.
- From introspection-- We have access to some degree to our own mental processes (we think),
- From history-- Previous hypotheses and models based on other data sets, sometimes from other fields.
- From considerations of elegance, simplicity, consistency, and coherence.

- Charles Kemp (and Josh Tenenbaum and others in their group) have used a hierarchical Bayesian approach in which the highest level involves a specification of model classes
  - using the data (and priors associated with the models), probabilities are assigned to the various classes as well as the particular model within each class.
- This approach still requires a careful delineation of the model types that are to be assessed, and in any given application the model types are a very small subset of those that are conceivable.

- Mark Steyvers and I developed our REM model for memory with a different sort of Bayesian induction.
- We first made some general assumptions about representation of memories and the processes and memory storage and retrieval.
- We then used Bayesian induction to specify the class of models that would optimize performance.

- That is, we did not specify a model class and then use the data to assign beliefs.
- Instead, we used Bayesian induction to choose a model class that would optimize performance on a given trial in a particular paradigm (recognition memory).
- Only after the class was so specified did we try to use the model to fit data.

- We started with some constraining assumptions about memory storage and retrieval:

STUDY:
A) Functionally separate traces are vectors of feature values
B) Episodic traces are incomplete. Each feature position in the vector is filled with a specified probability (u) each time unit
C) Episodic traces are stored with error: A given vector position is filled with the correct value with probability (c), or if not, with a random choice according to the environmental base rates (g)
TEST:
A) The probe vector is matched to each trace, to determine matching and mismatching feature values.
B) These are used to the likelihood ratio that the probe and trace j match.
C) The j likelihood ratios produce an odds that the test is OLD

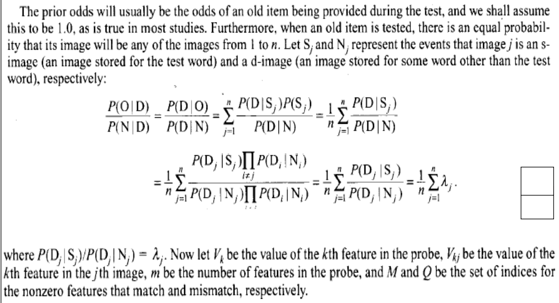
- Simplified recognition memory study:
- A list of study items is stored as a group of episodic traces.
- The tests have equal numbers of list items and new items, the participant guessing which is the case for each one.
- In theory, the retrieval probe vector consists of item plus list context features.

- However, to make the Bayesian derivations possible, we simplified by assuming that the context cues are used first, and restrict the episodic memory traces to those of the items from the studied list. Then the content features are used as a probe, and are compared to just this restricted set of traces.

- We asked: What is the best recognition performance that could be achieved on a given test? This could be thought of as an ***'ideal retriever'.***
- We answered this by using Bayes rule to calculate the probability (the odds) that the test item is ***old*** (on the list) vs ***new*** (not on the list). The optimal or ideal Bayesian retriever would respond old if the odds favored old (i.e. if the odds were greater than 1.0) and new otherwise.

- Here is how the induction goes:
- The odds for the test word being old (O) over new (N) equals the likelihood ratio of observing the data D for an old or new test times the prior odds for an old or new test (prior odds indicated by a subscript of o).
-

$$\frac{P(\mathrm{O}\mid \mathrm{D})}{P(\mathrm{N}\mid \mathrm{D})} = \frac{P(\mathrm{D}\mid \mathrm{O})}{P(\mathrm{D}\mid \mathrm{N})}\,\frac{P_{\mathrm{o}}(\mathrm{O})}{P_{\mathrm{o}}(\mathrm{N})}$$

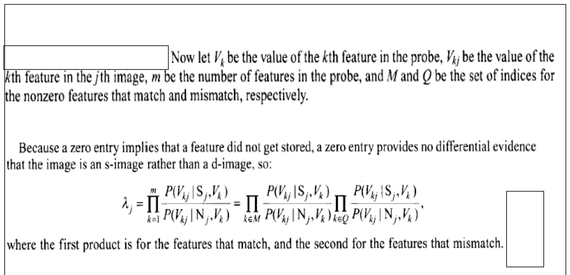


The prior odds will usually be the odds of an old item being provided during the test, and we shall assume this to be 1.0, as is true in most studies. Furthermore, when an old item is tested, there is an equal probability that its image will be any of the images from 1 to n. Let S_j and N_j represent the events that image j is an s-image (an image stored for the test word) and a d-image (an image stored for some word other than the test word), respectively:

$$\frac{P(\mathrm{O}|\mathrm{D})}{P(\mathrm{N}|\mathrm{D})} = \frac{P(\mathrm{D}|\mathrm{O})}{P(\mathrm{D}|\mathrm{N})} = \sum_{j=1}^{n}\frac{P(\mathrm{D}|\mathrm{S}_j)P(\mathrm{S}_j)}{P(\mathrm{D}|\mathrm{N})} = \frac{1}{n}\sum_{j=1}^{n}\frac{P(\mathrm{D}|\mathrm{S}_j)}{P(\mathrm{D}|\mathrm{N})}$$

$$= \frac{1}{n}\sum_{j=1}^{n}\frac{P(\mathrm{D}_j\,|\mathrm{S}_j)\prod_{i\neq j}P(\mathrm{D}_i\,|\mathrm{N}_i)}{P(\mathrm{D}_j\,|\mathrm{N}_j)\prod_{i\neq j}P(\mathrm{D}_i\,|\mathrm{N}_i)} = \frac{1}{n}\sum_{j=1}^{n}\frac{P(\mathrm{D}_j\,|\mathrm{S}_j)}{P(\mathrm{D}_j\,|\mathrm{N}_j)} = \frac{1}{n}\sum_{j=1}^{n}\lambda_j\,.$$

where $P(\mathrm{D}_j|\mathrm{S}_j)/P(\mathrm{D}_j|\mathrm{N}_j) = \lambda_j$. Now let V_k be the value of the kth feature in the probe, V_{kj} be the value of the kth feature in the jth image, m be the number of features in the probe, and M and Q be the set of indices for the nonzero features that match and mismatch, respectively.

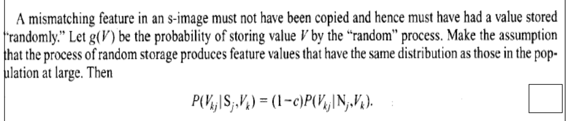


Now let V_k be the value of the kth feature in the probe, V_{kj} be the value of the kth feature in the jth image, m be the number of features in the probe, and M and Q be the set of indices for the nonzero features that match and mismatch, respectively.

Because a zero entry implies that a feature did not get stored, a zero entry provides no differential evidence that the image is an s-image rather than a d-image, so:

$$\lambda_j = \prod_{k=1}^{m}\frac{P(V_{kj}\,|\mathrm{S}_j,V_k)}{P(V_{kj}\,|\mathrm{N}_j,V_k)} = \prod_{k\in M}\frac{P(V_{kj}\,|\mathrm{S}_j,V_k)}{P(V_{kj}\,|\mathrm{N}_j,V_k)}\prod_{k\in Q}\frac{P(V_{kj}\,|\mathrm{S}_j,V_k)}{P(V_{kj}\,|\mathrm{N}_j,V_k)},$$

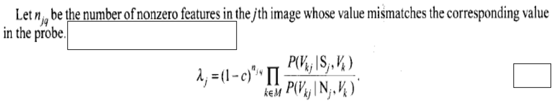
where the first product is for the features that match, and the second for the features that mismatch.



A mismatching feature in an s-image must not have been copied and hence must have had a value stored "randomly." Let $g(V)$ be the probability of storing value V by the "random" process. Make the assumption that the process of random storage produces feature values that have the same distribution as those in the population at large. Then

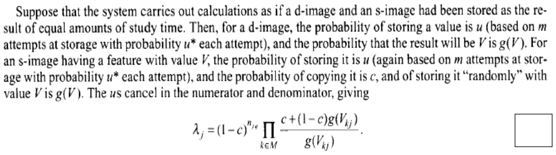
$$P(V_{kj}|\mathrm{S}_j,V_k) = (1-c)P(V_{kj}|\mathrm{N}_j,V_k).$$

This expression holds for the feature values that mismatch. Hence:



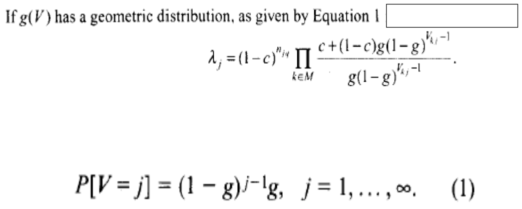
Let n_{jq} be the number of nonzero features in the jth image whose value mismatches the corresponding value in the probe.

$$\lambda_j = (1-c)^{n_{jq}} \prod_{k\in M} \frac{P(V_{kj} \mid S_j, V_k)}{P(V_{kj} \mid N_j, V_k)}.$$



Suppose that the system carries out calculations as if a d-image and an s-image had been stored as the result of equal amounts of study time. Then, for a d-image, the probability of storing a value is u (based on m attempts at storage with probability u^* each attempt), and the probability that the result will be V is $g(V)$. For an s-image having a feature with value V, the probability of storing it is u (again based on m attempts at storage with probability u^* each attempt), and the probability of copying it is c, and of storing it "randomly" with value V is $g(V)$. The us cancel in the numerator and denominator, giving

$$\lambda_j = (1-c)^{n_{jq}} \prod_{k\in M} \frac{c+(1-c)g(V_{kj})}{g(V_{kj})}.$$



If $g(V)$ has a geometric distribution, as given by Equation 1

$$\lambda_j = (1-c)^{n_{jq}} \prod_{k\in M} \frac{c+(1-c)g(1-g)^{V_{kj}-1}}{g(1-g)^{V_{kj}-1}}.$$

$$P[V=j] = (1-g)^{j-1}g, \quad j = 1, \ldots, \infty. \qquad (1)$$

- So we have now written the odds favoring the test item being old vs. new in terms of the two parameters c and g, and the data.

  The data consists of the matching feature values and the mismatching feature values for each trace in episodic memory.

  The optimal Bayesian decision on a trial is to respond OLD if the following odds is > 1.0:

[illegible]

- Putting aside the many simplifications and constraints in this derivation, we have a Bayesian derivation that uses the two system parameters to link the odds favoring an 'old' decision to the observed data, *for a given test trial*.

- If the decision uses the default optimal odds of 1.0, we can predict performance by summing across all possible memory configurations for a given condition, weighted by the probability of that configuration.

- We do this by simulation

- (But note that Max Montenegro in Jay Myung's lab recently derived analytic predictions with a Fourier transform approach).

- SIMULATION: For a given condition:

1) Specify values for c and g.
2) Simulate the traces that get stored in memory for the study list, producing a set of incomplete and error prone vectors.
3) For TEST, choose a random study item for an OLD test, and a random vector produced from base rates for a NEW test.
4) Give an OLD decision if the formula gives odds > 1.0, NEW otherwise.

- Do this simulation N times: The proportion of old decisions is the probability of an old response for that condition, for those values of c and g.

- In the original REM article we produced qualitative patterns of predictions for plausible parameter values for standard findings (see next).
- In many other articles we carried out quantitative fitting by searching the space of c, g values, obtaining the probability of the observed number of OLD responses in each condition of interest, for each c, g combination

- In most applications we did not use REM to carry out a full Bayesian analysis, instead finding the parameters producing the maximum likelihood of the data, and using the fit for inference.

- Before turning to REM's performance, a few remarks about its characteristics are useful. First, note that the expected value of the likelihood ratio (old/new) for a match of a new item to a random list trace is 1.0, and the expected value of the likelihood ratio (new/old) for the match of an old item to its own list trace is also 1.0, with both distributions being highly skewed.

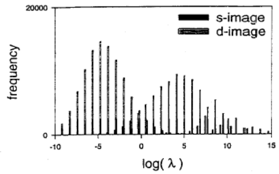
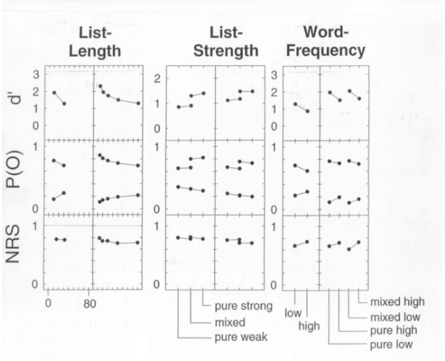


Fig. 4.4 Distribution of the natural logarithm of the likelihood ratio (λ) for an *s*-image (the image stored when the test word had previously been studied, illustrated with filled bars) and for *d*-images (images of other studied words; illustrated with open bars).

The odds used for decision is for a NEW item an average of N samples from the d-image distribution; for an OLD item, an average of one sample from the s-image distribution and N-1 samples from the d-image distribution.

Note that the overlap of these distributions is unchanged on the original non-log plots– i.e. sensitivity is unchanged by the change of scale.

- Among other things these properties of the likelihood ratios imply that recognition is generally 'centered' with respect to hits and false alarms.
- Thus REM using the default decision criterion of odds of 1.0 automatically produces mirror effects (as we shall see next).



- What we have so far is a vastly simplified model for episodic recognition memory. Although it makes a few nice predictions, it would be of little interest if it could not be generalized considerably, in two ways:
  - By making more reasonable assumptions about the boundary conditions
  - By proving applicable to many other types of tasks.

We have done this, both in the original article, and in many subsequent articles.

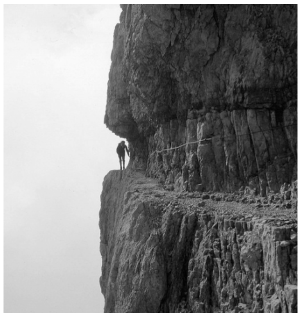
- One of the critical issues is the treatment of context (in particular list context). In the original REM, we bypassed the problem by restricting consideration to memory traces of items studied on the recent list (assuming that context somehow performed this restriction).
- We did look at ways to use a joint probe with both context and item cues, although exact Bayesian solutions were no longer possible.
- We also asked what would happen if we used the model as originally stated, but expanded the set of memory traces to which the probe is matched to include those due to pre-list experience (possibly including pre-list occurrences of the test item).
  - The system seemed robust to such manipulations.

- The REM model in its simplest form attributed performance decreases as list length increased to confusions with the greater number of memory traces in the longer lists.
- Simon Dennis and Mike Humphreys have argued that such interference due to similarity between list words has no effect, that true list-length effects do not exist, and that all memory interference in recognition studies with words are due to confusions with pre-list occurrences of the test word.

- The more general REM approach incorporates both kinds of interference. Amy Criss and I carried out a study to assess the relative strength of these factors: Our design placed many test words in lists before the current test list. In these circumstances the interference from the test word traces due to pre-list presentations proved even stronger than the interference due to within list similarity due to other words, but we found strong evidence for both factors.

- One nice feature of the REM model was the natural way that it connected to the earlier models for ***recall*** that I developed with Raaijmakers (SAM).
- In SAM, recall was conceived as a search process in which each cycle consisted of sampling a memory trace (in proportion to its strength), recovering information from the sampled trace, and then making decisions on the basis of that information (including a decision whether to respond, continue sampling, or give up).

- The REM model could use this recall system almost in its entirety, substituting the trace likelihood ratio for the ‘strength’ value of SAM.
- However, the SAM assumption of proportional sampling required modification in REM–
- I showed earlier how strongly skewed are the likelihood ratios. Proportional sampling would almost always select the strongest trace, which would in turn strongly distort the very successful predictions of SAM. However, an assumption that traces are sampled in proportion to something like a log of the likelihood ratios would reproduce essentially all the SAM recall predictions.



- Another key feature of the REM approach is the process it provided for knowledge development and retrieval of knowledge.
  - and the way the system predicts the relation of, and the interactions between, episodic storage and retrieval to knowledge development and retrieval.

- The linkage began with the list strength effect and differentiation.
  - The list strength effect is the finding that strengthening some list items benefits the recognition of other, non-strengthened, list items.
  - Using SAM, we proposed a process of differentiation to explain this: The more we know about an item the less noise in its representation, and the less similar it will be to a probe with a different item.
  - In REM this occurs naturally, as features are added to the strengthened item.

- However, to impose differentiation, it is necessary to assume that items strengthened by repetitions are represented by one trace that accumulates features at each repetition.
- We must assume that when an item re-occurs we often retrieve the previous trace (because it is sufficiently similar), and add some of the information in the new presentation to that already in the retrieved trace.

- This addition process can be used to explain the development of knowledge: As repetitions occur, a trace continues to accumulate information until it becomes knowledge, or in the case of words, part of our lexicon.
- We usually think of knowledge as independent of particular contexts. The process of accumulation just described causes the accumulation of many different context features, until the point is reached where a trace has features of so many different contexts that none emerges from the mixture, and the trace is effectively decontextualized.

- This process also explains long-term priming of knowledge retrieval:
- Presentation of, say, a word, causes current context to be added to the lexical trace. A later test in a similar context will use a probe cue with similar context features, producing better matching to the lexical trace.
- We have used this idea with considerably success to explain long term priming in a number of studies and settings.

- Of course priming must be appended to a model of knowledge retrieval.
- Retrieval of knowledge traces also fits nicely into the REM framework:
- The probe cue is compared to the lexical traces in parallel, each trace contributing a likelihood ratio for matching the probe, in a process that evolves over time.

- We have used variants of this idea to form models of lexical decision, naming, animacy judgment, forced choice perceptual identification, and other knowledge retrieval tasks, and have modeled priming effects in these various tasks.

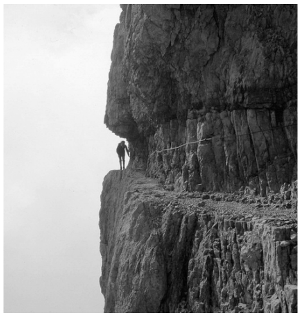
- These applications of variants and extensions of REM to a variety of episodic and knowledge retrieval tasks, and a host of interesting theoretical questions that arise during the applications, are potential subjects for addresses in their own right.
- Today, however, I wish to discuss an theoretical elaboration of REM, developed first with Shane Mueller, and more recently with Angela Nelson.

- It is rather remarkable how such a simple model as REM, containing almost no structure, and few parameters and assumptions, proved useful in predicting data from so many episodic and knowledge retrieval tasks.
- However, REM is far too simple, and is especially deficient if one tries to use it to explain the formation of real knowledge:
- Vectors of feature values, each encoded only as a base rate, are not rich enough a substrate on which to build knowledge.

- We therefore changed the representation:
- Each event, episodic or knowledge is represented as a matrix of feature (value) co-occurences.
- Episodic traces generally have one count per cell, but as knowledge develops the trace accumulates counts in each cell of the matrix.
- The co-occurrence counts are based on the features that are together at one time in short term memory, and especially on attention to features of those items.

- Thus each item representation picks up features of ‘nearby’ items, and items that co-occur a lot tend to have knowledge traces that grow more similar.
- These assumptions also imply that events and knowledge co-evolve in the following way: We encode and interpret new events in terms of our continuing evolving knowledge, and we use the features of new events to modify and add to our knowledge.

- We term this model REM-M (M for matrix)
- In the rest of this talk I will flesh out the ideas underlying REM-M, and discuss one simplified version applied to the study discussed next.
- The study is by Angela Nelson (in the audience) on perceptual learning and the effects of differential experience.



Chinese Characters

汛 完 沌 君

列 尬 囫 判

Design

- Our study uses Chinese characters as the novel stimuli.
- Subjects are trained using a visual search task modeled after that of Shiffrin and Lightfoot (1997)
- Search displays of 2 or 4 characters for a character presented just prior to the display; respond present or absent

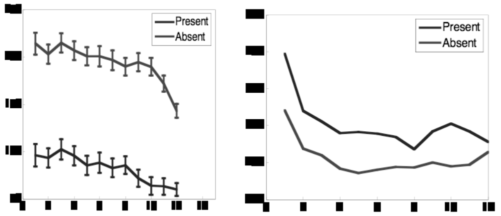
Design

- Frequency of occurrence (as both target and foil) of characters was varied in a ratio of 2::6::18::54
- For each subject, a set of 32 characters was taken from the pool of characters (which included approximately 200 characters, all with 7 strokes or less)
- From these 32 characters, 8 were assigned to each frequency category

Subject learning over training

- Response times measured, rate of search used to assess learning
  - Previous research (Shiffrin and Lightfoot,1997) showed a gradual but substantial decrease of search rate over training

Subject learning over training



Post-training tests

- Pseudo-lexical Decision – subjects identified whether a character was present in study sessions
- Episodic Recognition Memory – subjects viewed study list and test list
- Forced Choice Perceptual Identification – subjects viewed briefly flashed character and picked matching character from choice of two

Previous Models

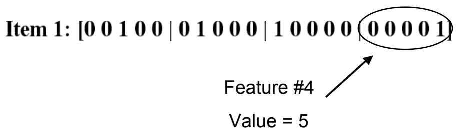
- REM (Shiffrin & Steyvers, 1997)
  - Items in memory represented as vectors of feature values
  - Assumes high frequency items are composed of high frequency features
  - As a result of this assumption, the high frequency items share more features with each other
  - This does not hold for our study!

Current Model – Contextual Diversity

- Simplification of the REM-M model proposed by Mueller and Shiffrin (e.g. 2006)
- Because the higher frequency items are seen in a larger variety of contexts than lower frequency items, the higher frequency items develop a more diverse representation in the lexicon

Item Representation

- Items are represented as a vector with a fixed number of features and a fixed number of possible values for each feature



Item 1: [0 0 1 0 0 | 0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1]

Feature #4

Value = 5

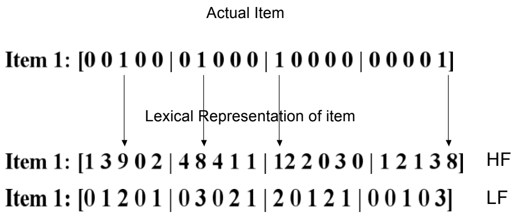
Training

- The lexicon is built through exposure to the items during training
- Each time an item is presented as a target in visual search, features are added to that item’s lexical representation from three possible sources:
  - Actual target item features
  - Distractor features
  - Previous target features

Surrounding Context

Training

- The lexicon builds up counts over training



Actual Item

Item 1: [0 0 1 0 0 | 0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1]

Lexical Representation of item

Item 1: [1 3 9 0 2 | 4 8 4 1 1 | 12 2 0 3 0 | 1 2 1 3 8] HF

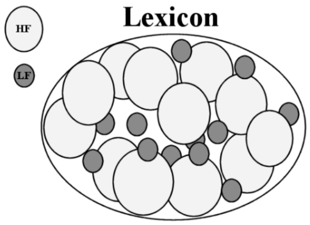
Item 1: [0 1 2 0 1 | 0 3 0 2 1 | 2 0 1 2 1 | 0 0 1 0 3] LF

Training

- HF items are presented more often and therefore are more likely to store features in their lexical entry from the surrounding context – they will have a more *diverse* lexical entry
- Because the surrounding context is also more likely to be HF items, HF items tend to overlap more with each other than LF items

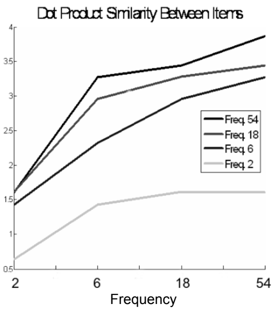
Training

- Since HF items have a larger “cloud” of features in the lexicon, they are more likely to overlap with other items



Training

- After training, HF items are more similar to each other than LF items
- Similarity between items measured by taking a dot product of the two item’s normalized lexical entries

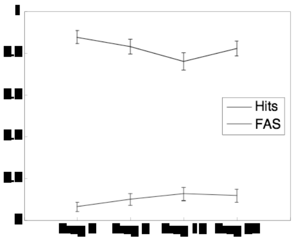


Post-training tasks

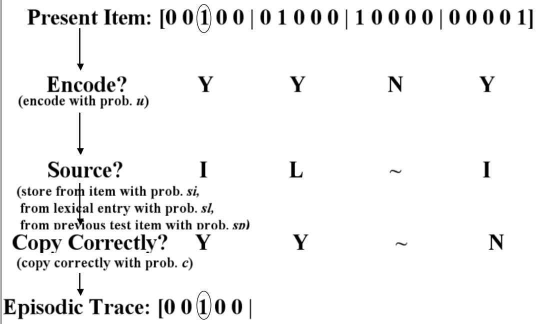
- The lexicon that has been established over training is used in the simulation of the post-training tasks
  - – Episodic Recognition
  - – Lexical Decision
  - – Forced Choice Perceptual Identification

Episodic Recognition

- Results of empirical study: LF trained items produced better performance than HF items in an episodic recognition task
- Showed mirror pattern



Episodic Recognition: Study Phase



Present Item: [0 0 1 0 0 | 0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1]

Encode? Y Y N Y

(encode with prob. *u*)

Source? I L ~ I

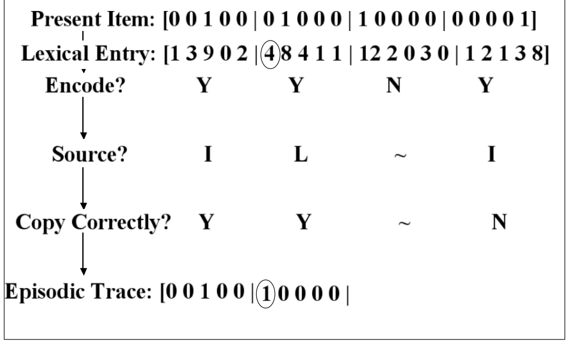
(store from item with prob. *si*, from lexical entry with prob. *sl*, from previous test item with prob. *sp)*

Copy Correctly? Y Y ~ N

(copy correctly with prob. *c*)

Episodic Trace: [0 0 1 0 0 |

Episodic Recognition: Study Phase



Present Item: [0 0 1 0 0 | 0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1]

Lexical Entry: [1 3 9 0 2 | 4 8 4 1 1 | 12 2 0 3 0 | 1 2 1 3 8]

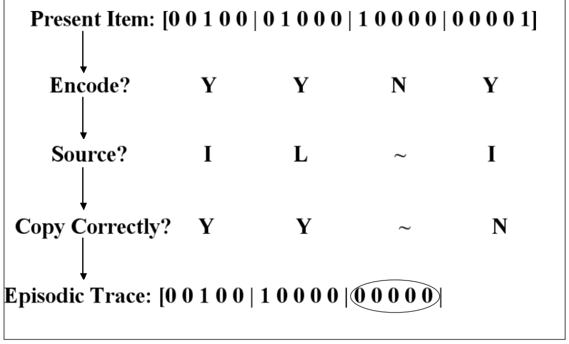
Encode? Y Y N Y

Source? I L ~ I

Copy Correctly? Y Y ~ N

Episodic Trace: [0 0 1 0 0 | 1 0 0 0 0 |

Episodic Recognition: Study Phase



Present Item: [0 0 1 0 0 | 0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1]

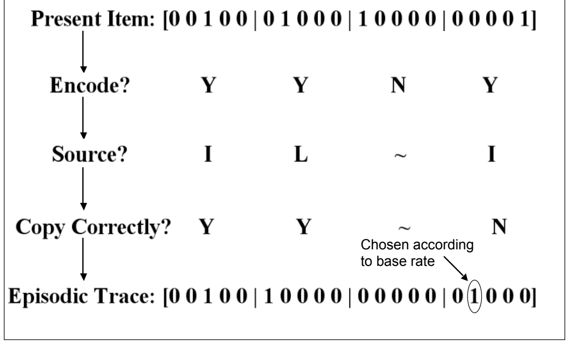
Encode? Y Y N Y

Source? I L ~ I

Copy Correctly? Y Y ~ N

Episodic Trace: [0 0 1 0 0 | 1 0 0 0 0 | 0 0 0 0 0 |

Episodic Recognition: Study Phase



Present Item: [0 0 1 0 0 | 0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1]

Encode? Y Y N Y

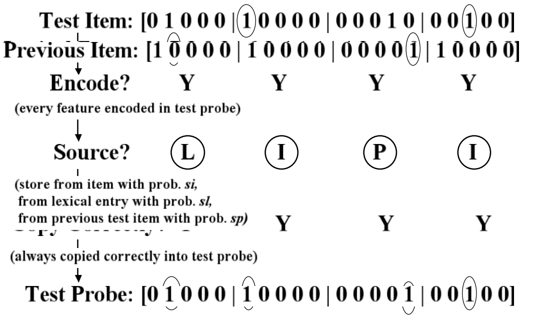
Source? I L ~ I

Copy Correctly? Y Y ~ N

Chosen according to base rate

Episodic Trace: [0 0 1 0 0 | 1 0 0 0 0 | 0 0 0 0 0 | 0 1 0 0 0]

Episodic Recognition: Test Phase



Test Item: [0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 1 0 | 0 0 1 0 0]

Previous Item: [1 0 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1 | 1 0 0 0 0]

Encode? Y Y Y Y

(every feature encoded in test probe)

Source? L I P I

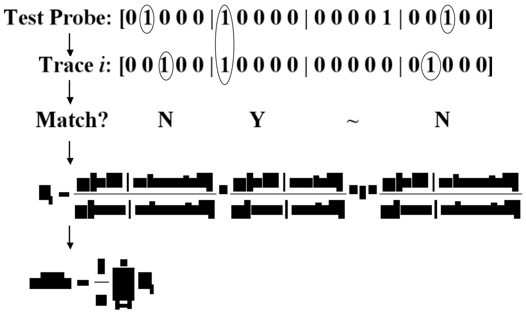
(store from item with prob. *si*, from lexical entry with prob. *sl*, from previous test item with prob. *sp)*

Y Y Y

(always copied correctly into test probe)

Test Probe: [0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1 | 0 0 1 0 0]

Episodic Recognition: Test Phase

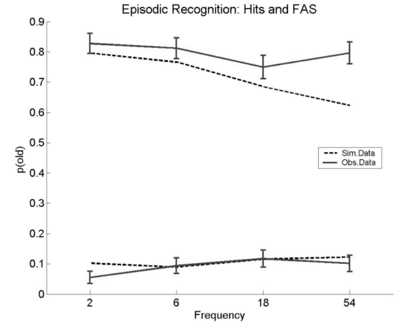


Test Probe: [0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 0 1 | 0 0 1 0 0]

Trace *i*: [0 0 1 0 0 | 1 0 0 0 0 | 0 0 0 0 0 | 0 1 0 0 0]

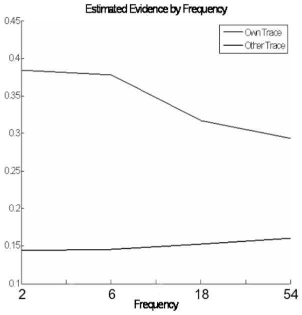
Match? N Y ~ N

Episodic Recognition



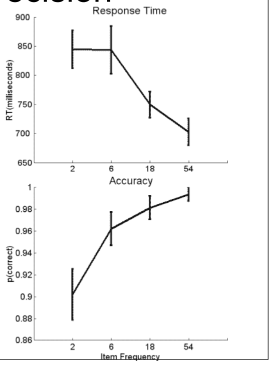
Diversity of Items

- LF items are more likely to match their own traces, and less likely to match different traces



Lexical Decision

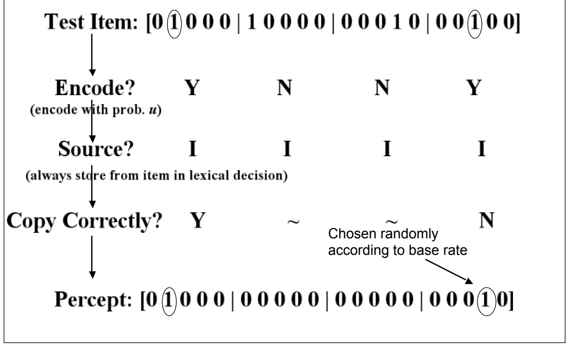
- Next, the model was used to fit the experimental data from the pseudo-lexical decision task



Modeling Process: Lexical Decision

- When presented with a test item, the percept accumulates *item* features as time passes
- At each time-step, each feature is encoded with probability *u*, and if encoded is copied correctly with probability *c*, otherwise is stored randomly according to lexical base rate

Lexical Decision: Create Percept



Test Item: [0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 1 0 | 0 0 1 0 0]

Encode? Y N N Y

(encode with prob. *u*)

Source? I I I I

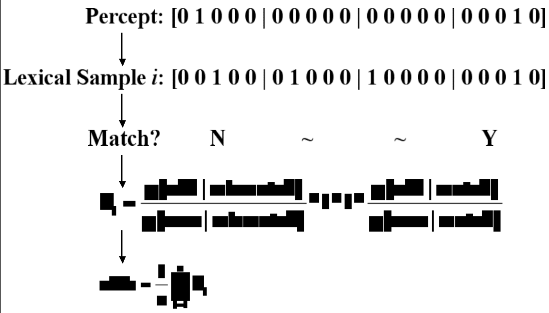
(always store from item in lexical decision)

Copy Correctly? Y ~ ~ N

Chosen randomly according to base rate

Percept: [0 1 0 0 0 | 0 0 0 0 0 | 0 0 0 0 0 | 0 0 0 1 0]

Lexical Decision: Compare Percept



Percept: [0 1 0 0 0 | 0 0 0 0 0 | 0 0 0 0 0 | 0 0 0 1 0]

Lexical Sample *i*: [0 0 1 0 0 | 0 1 0 0 0 | 1 0 0 0 0 | 0 0 0 1 0]

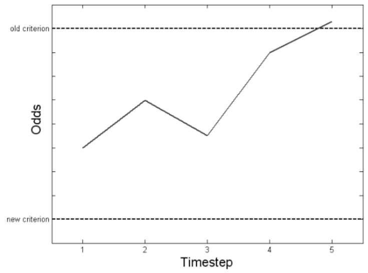
Match? N ~ ~ Y

Lexical Decision: Compare Percept

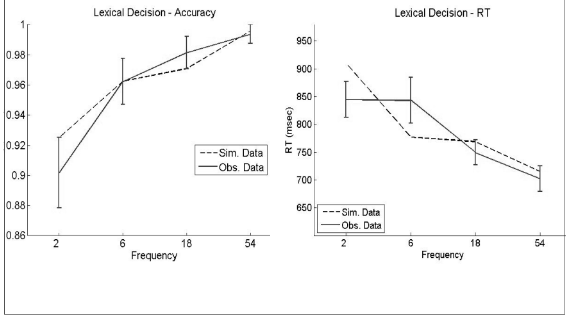
- If odds > old criterion, respond “old”
- If odds < new criterion, respond “new”

- If neither, then repeat the procedure, adding new features to the existing percept without replacing those already stored
- Re-calculate evidence

Modeling Process: Lexical Decision



Lexical Decision



Summary

- Differences in contextual diversity that develop over the training of novel items produce frequency effects in the model like those shown in the experimental results

- Now I'd like to return to the more general form of REM-M developed and simulated by Shane Mueller.

What Are the Features for General (Semantic) Knowledge?

- All the physical features
  - For treehouse: color, shape, size, etc.
  - For words: letter shapes, sound
- Inferred Properties
  - Physical features in percept enable access to vast amount of inferred information.
  - made of wood, in a tree, doors, windows
- Associated Properties
  - Dangerous, children, "Stand By Me"

What Are Properties of Features?

- Features Differ in importance
  - height from ground versus roofing material
- Features can have different conflicting values
  - Not all tree-houses are the same
  - yellow versus brown
- Features co-occur differentially
  - stilt-houses versus tree-houses

Co-occurrence representations

- The richness of the world and of concepts cannot possibly be captured as a list of simple, separate primitive, features, unless we extend the notion of feature to include large and integrative concepts such as word associations, paragraphs, books, etc., which would be effectively useless for modeling.
- As a first and critical step, therefore, we try to capture much of what is needed by encoding and storing which features are co-occurring with which other features.

Capturing Co-occurrence Information Knowledge and Episodic Representations

- Accumulate co-occurrence of features corresponding to a concept.
- Knowledge Matrix:
  - Set of multiple conditional representations

[<**5** 3 | 3 4 | 1 0 0>
<3 **6** | 0 1 | 1 3 0>
<3 0 | **4** 3 | 0 0 3>
<4 1 | 3 **5** | 4 2 1>
<1 1 | 0 4 | **6** 0 1>
<0 3 | 0 2 | 0 **4 1>**
<0 0 | 3 1 | 1 1 4>]

Forming LTM Co-occurrence matrix from Episodes

- Any episodic trace can produce a co-occurrence matrix.
- 110011 and 001100 form co-occurrence matrices at right
- Knowledge accrues by incorporating co-occurrence matrix from individual episodes into current knowledge structure.

110011:

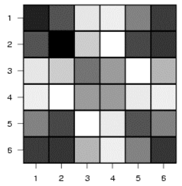
- 1 1 0 0 1 1
- 0 0 0 0 0 0
- 0 0 0 0 0 0
- 1 1 0 0 1 1
- 1 1 0 0 1 1

001100:

- 0 0 0 0 0 0
- 0 0 0 0 0 0
- 0 0 1 1 0 0
- 0 0 1 1 0 0
- 0 0 0 0 0 0
- 0 0 0 0 0 0

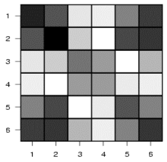
LTM Co-occurrence matrix

- Over time, complex matrix representation will form.
- Word has two primary concepts: 1-2-5-6 and 3-4.
- 1256 is stronger than 34
- Primary meanings could be extracted using factor analysis)



Encoding Episodes from Knowledge Matrix

- Generic Encoding: pick row, pick feature, pick new row based on what has been sampled, repeat.
- Biased Encoding: pick row from another trace, pick feature from that row of current matrix, repeat.

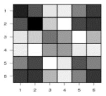


Biased Encoding

- 1-2 are features of months.
- Features 3-4 are properties of brass bands.

  "Late in winter, March blizzards are not uncommon"
- Nearby words create local semantic context of season and date.
- Sampling from rows 1-2.

MARCH



- Biased sampling will produce episodic trace for the month MARCH, rather than the parade MARCH.

How does knowledge form?

- We need to describe the process by which knowledge bootstraps itself into existence.
- E.g A word starts as a collection of physical features, without meaning.
- The idea is that the 'context' of the event provides features co-occurring with the word, and these tend to join the knowledge trace. Over many event occurrences in varying contexts, knowledge including meaning emerges.

Formation of Model Mental Lexicon

- We will produce a toy lexicon with such a bootstrapping process.
- Start off each word with a 'unique' representation
- During encoding, keep track of local feature context, and store co-occurrence counts in the lexical trace
- Words encoded in a way biased by context.
- Knowledge re-stored along with some features from local context.
- One result: Concepts that often appear together grow more similar (Hebbian-like principle).

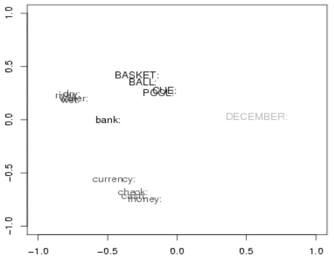
Semantic Spaces Demonstration: Bank

- Create lexicon for model by "reading" text.
- In text, BANK appears with MONEY words, RIVER words, or BALL words.
  - I deposit check in bank
  - money is withdrawn from bank

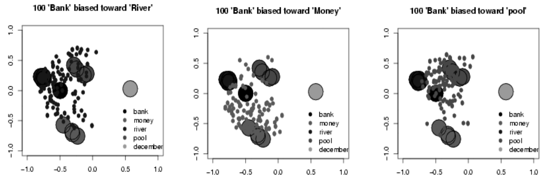
  - the river bank was dry
  - there was water on river bank
- In demo:
  - Form matrix representation for each word
  - Compute similarity between each obtained matrix
  - Perform 2-D MDS to visualize space.

Results: Bank Demonstration

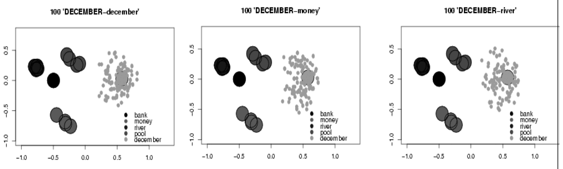
- After lexicon formed from text, perform MDS on feature representations



Biased Encodings of “Bank”



Biased Encodings of “December”

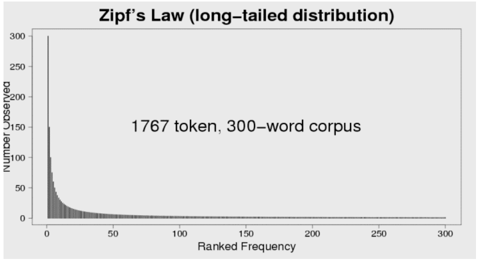


Model for Episodic Memory Tasks

- Previous simulation shows that semantic spaces can emerge within the matrix representation
- System must be extended to form model of episodic memory tasks (primarily list memory)
- Three main steps:
  - ‘Grow’ a Model Lexicon
  - Encode Episodes with the Lexicon, and Store Episodic Traces
  - Compute Likelihood Ratios = Trace Activations

Step 1: Form Model Mental Lexicon

- Real words occur in different frequencies.
- Typically, the few most common words happen a lot, and there are a lot of rare words that happen a little.



Step 1: Form Model Lexicon

- Each word initialized with a unique representation.
- Half of the features are “Physical”describing physical properties of word
- Half are relational, forming local semantic context.
- During lexicon formation:
  - observed tokens are encoded by consulting knowledge
  - features from local semantic context are added to relational features of trace.
  - New augmented trace is added to current semantic matrix for word.
- In this process, things that co-occur will gradually begin to share relational features.
- High-frequency words will grow more similar to one another than will low frequency words.

Step 2: Encode Episodic Traces

TASK: Suppose study of a list followed by single-item recognition for list items (targets) and non-list items (foils).

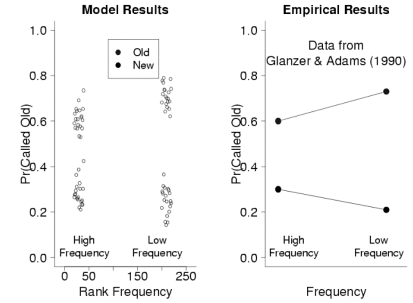
During study, episodic traces of each item are encoded from knowledge network (for now, independent of nearby items on the list). These are stored incompletely and with error.

- At test, a probe is encoded as another episodic trace.
- Response depends on likelihood calculation: probability trace being produced by that probe.

Recognition Memory Demo

- Task: present 40-item list for study comprised of high and low-frequency words. Later, present 20 old words and 20 new words; participant says “Old” or “New” for each.
- Typical finding: LF words remembered better; higher hits and lower false alarms.
- In the model, lexicon formation makes HF words more similar to one another.

Mirror Frequency Effect

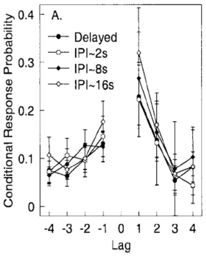


Encoding Biases from Nearby List Words

- The previous demo treated all words as independent units, unaffected by local study list context. This does not utilize the power of the new approach, that stores co-occurrences.
- We therefore provide a second demonstration that encodes list items in terms of other list items currently in rehearsal.

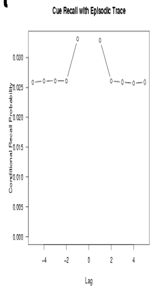
Biases of Encoding Demonstration

- Free Recall CRP
  - A list of words is presented
  - Participant recalls as many words as possible from the list, in any order
  - Words tend to be recalled from nearby input positions
  - Forward Bias: recall tends to go forward rather than backward.
  - Kahana et al., 2002



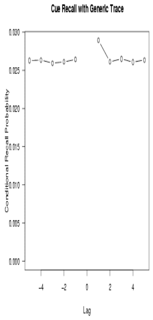
Encoding Bias Account

- Encoding of words is biased by the meaning of previous words.
- Produces “Biased” episodic traces.
  Presented: A B C D E
  Encoded: A aB bC cD dE
- During recall, new memory traces are identified by matching currently recalled word:
  - bC -> cD or bC->aB
  - But simplest model produces equal forward and backward biases



Encoding Asymmetry

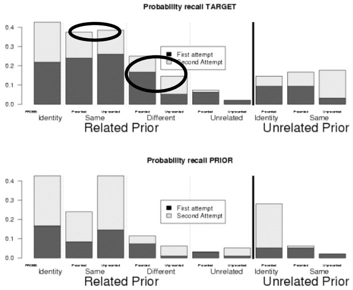
- If traces are reconstructed (which they must be for recall to happen), a more ‘generic’ trace may be generated during recovery, one less biased by nearby words.
- Presented: A B C D E
  Encoded: A aB bC cD dE
- During recall, new memory traces are identified by matching currently recalled word:
  - bC ->C-> cD; not bC->C->aB



Encoding Asymmetry Predictions

- If encoding biases are happening, then we can bias encoding just by placing a word nearby another word.
- Borne out in new experiment:
  - List containing words like “BANK” presented
  - Bank preceded by either “Money” or “River”
  - At recall, subject given a hint: “Check” versus “Water”
  - Cues consistent with prior word produced more recalls of the word “Bank”

Encoding Asymmetry Predictions



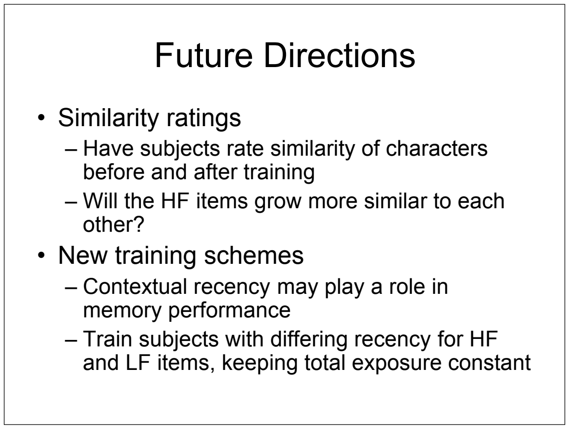
Model Summary

- Knowledge representation as a feature-based co-occurrence matrix.
- Episodes cause added counts to accumulate in lexicon
- Allows knowledge and new conceptual relations to develop from experience
- Concepts that co-occur grow more similar by sharing features.
- This approach:
  - Explains the development of semantic spaces
  - Captures multiple senses and meanings of a concept
  - Accounts for frequency effect in recognition memory
  - Allow encoding biases to be explored and explained.
- Unites previously independent approaches toward memory and knowledge, bringing new insights to both.

Applications

- **Semantic Spaces**
- **Frequency Effects in Episodic Memory**
- **Biases in encoding**

- *Priming*
- *Implicit Association Test*
  - *Consistent correlations in environment embed multiple connotations in concepts.*
- *Text comprehension/disambiguation*
- *Whorfian Hypothesis*
- *Corpus Analysis*



Future Directions

- Similarity ratings
  - Have subjects rate similarity of characters before and after training
  - Will the HF items grow more similar to each other?
- New training schemes
  - Contextual recency may play a role in memory performance
  - Train subjects with differing recency for HF and LF items, keeping total exposure constant