Introduction to Cognitive Modeling

Q550: Models in Cognitive Science

Lecture 1

What are we doing here?

• An ill-defined art…

- ➢ no textbook on models or techniques
- ➢ no clear definition what a cognitive model is
- ➢ no accepted practice in designing and evaluating

BUT:
Over 80% of articles appearing in major journals of Cognitive Science involve cognitive modeling

Applications of cognitive modeling are being seen more often in applied fields: human factors, clinical, cognitive neuroscience, agent-based modeling in economics, etc.

Busemeyer (in prep)

Cognitive modeling is becoming an essential tool for cognitive science (in particular) and social sciences (in general)

Any student needs to be a competent reader and perhaps user of these tools

Problem: **model** is not clearly defined in this realm, and **cognitive science** itself is underspecified as a discipline

We know our models should have something to do with theory, but what differentiates a cognitive model from a conceptual framework, or purely statistical models?

We see errors in interpretation in our literature:

Defining cognitive models

We need to operationally define a **cognitive model**, our **goal**, and the **steps** involved in achieving that goal

Computational cognitive modeling has evolved from mathematical modeling of behavior in psychology (and AI)

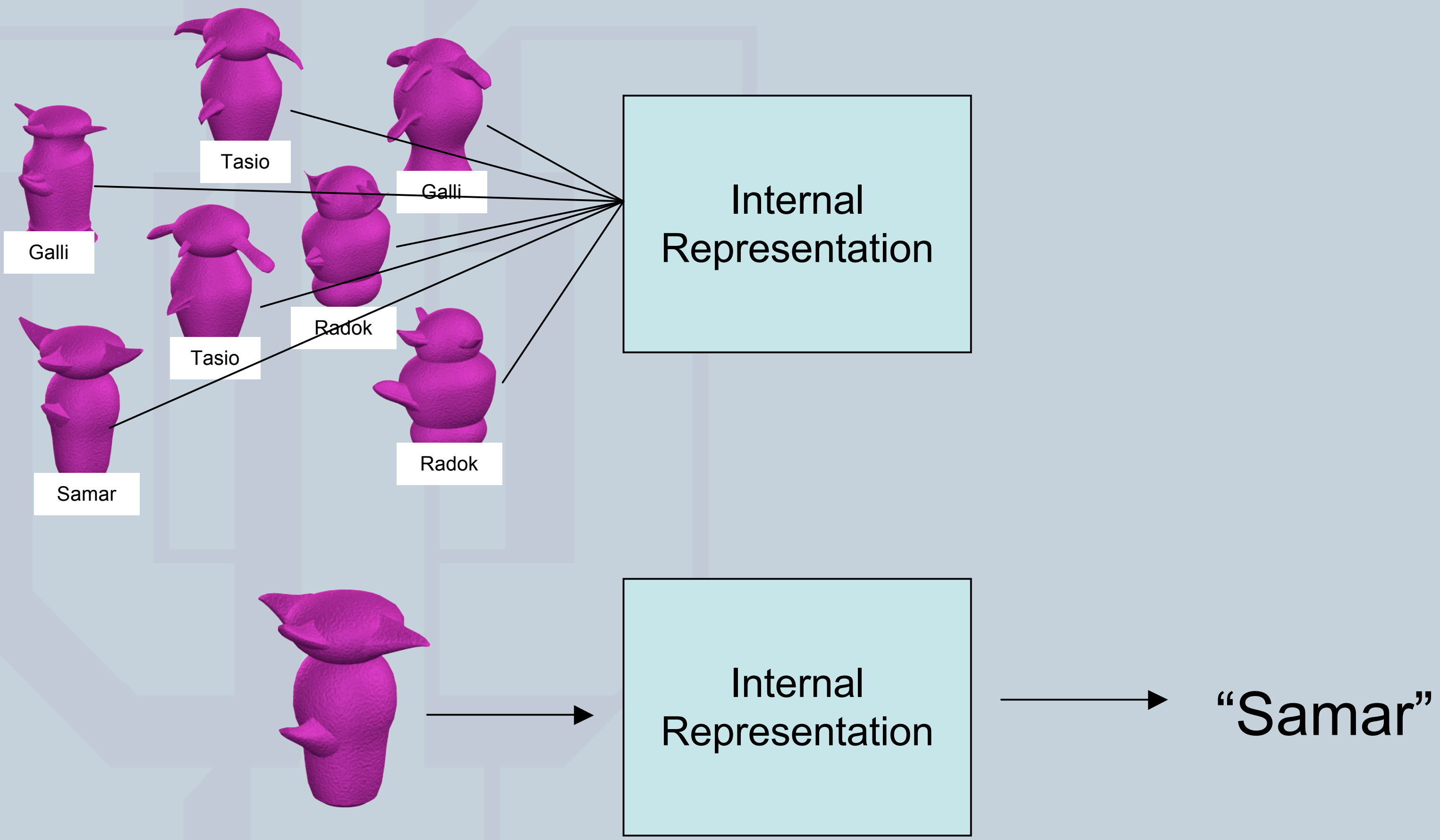
We hardly ever see the elegant closed-form solutions anymore. We have traded elegance for flexibility --> we can now test models that were not possible before modern computers, but the price we pay is a loss in ease of communication

A cognitive model

Input-Output black box example.

- sources of noise in cognitive system makes this more difficult

How do we categorize perceptual stimuli?
(Lets use novel stimuli so we have control)



Prototype model vs. exemplar model; ordinal/magnitude diffs

What is Cognitive Modeling?

How do we differentiate a cognitive model from a conceptual or statistical model?

"Cognitive science is concerned with understanding the processes that the brain uses to accomplish complex tasks including perceiving, learning, remembering, thinking, predicting, inference, problem solving, decision making, planning, and moving around the environment. The goal of a cognitive model is to scientifically explain one or more of these basic cognitive processes, or explain how these processes interact."

--Jerome Busemeyer

Features of a Cognitive Model:

1) Goal is to scientifically explain basic cognitive processes
2) Described in formal (mathematical or computer) languages

Conceptual frameworks are broadly stated theoretical assumptions

Cognitive models differ in that they are formally stated

E.g., Craik and Lockhart's LOP hypothesis provides a conceptual framework for memory, whereas Shiffrin's SAM model or Murdock's TODAM model, being mathematical, are examples of cognitive models (or, LSA vs. Indexical)

We can convert a conceptual framework into a cognitive model by formalizing it (recasting the verbal statements into mathematical or computer language)

Features of a Cognitive Model:

1) Goal is to scientifically explain basic cognitive processes
2) Described in formal (mathematical or computer) languages
3) Derived from basic principles of cognition

Statistical models are applicable to data from any field, as long as that data meet the statistical assumptions (normality, etc.)

These assumptions are not derived from any principles of cognition, and may be inconsistent with known facts of cognition

E.g., statistical assumptions are inconsistent w/ RT distributions

Cognitive models of response time accommodate this

Descriptive models vs. explanatory models (CSLR/SVM car)

Parameters must have a meaningful cognitive interpretation

Advantages of Cognitive Models:

1) Guaranteed to produce logically valid predictions

This is not true of conclusions based on intuitively-based verbal reasoning

E.g., early categorization research argued against exemplar models b/c transfer to a prototype was better than any of the experienced exemplars

But once the exemplar model was formalized, it naturally produces this effect

Reasoning from a conceptual framework can lead to incorrect conclusions

Goat game

Advantages of Cognitive Models

1) Guaranteed to produce logically valid predictions
2) Capable of making precise quantitative predictions

Even if a model predicts a correct qualitative (ordinal) effect, its predictions may be an order of magnitude off the mark

Formal models allow us to evaluate models both qualitatively and quantitatively in their correspondence to empirical data

Advantages of Cognitive Models

1) Guaranteed to produce logically valid predictions

2) Capable of making precise quantitative predictions

3) Capable of generalizing

Both cognitive models and statistical models (empirical curve fitting) are capable of generating qualitative predictions

But cognitive models should also *generalize* to new paradigms, and should describe the process that produces the output, rather than just describing the output itself

Newell's (1981) power law of practice vs. Logan's (1988) instance theory

Logan's theory produces and explains the power law, and can be used to make new predictions: how variance of RT dist changes w/ practice and how accurcy changes w/ practice (both inconsistent w/ a power law)

Practical Uses of Cognitive Models

- ***Clinical***: assessing individual differences between normals and clinical patients
- ***Cognitive neuroscience***: understanding the function of different brain regions
- *Aging research*: change in cognitive function with age
- ***Human factors***: improving human-machine interactions
- ***AI and Robotics***: automatic detection tools, handwriting recognition, face recognition, movement in robots, comprehension and document retrieval
- ***Social sciences***: cognitive and agent-based models of market behavior or social networking

Making intelligent systems more intelligent: Kasparov vs. Deep Blue

Major private sector modeling labs

Steps of Cognitive Modeling:

1) Conceptual theory → formal description

Steps of Cognitive Modeling:

1) Conceptual theory $\longrightarrow$ formal description
2) Ad hoc **assumptions** to complete formal description

Conceptual theory is often missing important details, so we have to make additional detailed assumptions in order to complete the model and to generate precise quantitative predictions

E.g., assumptions about features to represent stimuli

We try to minimize the number of ad hoc assumptions, but this step is often unavoidable

Steps of Cognitive Modeling:

1) Conceptual theory $\longrightarrow$ formal description
2) Ad hoc **assumptions** to complete formal description
3) Parameter estimation

Models almost always contain coefficients that are initially unknown, and these values need to be estimated from some of the observed data

E.g., the importance weight assigned to each feature is a free parameter that is estimated from the choice response data

We try to minimize the number of model parameters, but this is usually a necessary and important step

Steps of Cognitive Modeling:

1) Conceptual theory → formal description
2) Ad hoc **assumptions** to complete formal description
3) Parameter estimation
4) Compare predictions to empirical data

Qualitative: Ordinal and parameter free (models make these predictions for any value of the free parameters)

Quantitative: Magnitude of correspondence to data

Cognitive models can be compared to each other quantitatively, or we can use a base and saturated model

Steps of Cognitive Modeling:

1) Conceptual theory → formal description
2) Ad hoc **assumptions** to complete formal description
3) Parameter estimation
4) Compare predictions to empirical data
5) Iterate to constrain

Experiments are designed based on models, and models are constrained and modified/extended based on new data

This should produce an evolution of models that improve and become more powerful over time as the science in a field progresses

Generating model

Common Model Targets:

- ***Existence proof***: Multiple systems/processes/storage, e.g.: STM/LTM distinction, serial vs. parallel processing, implicit/explicit memory, semantic/episodic memory
- ***Model comparison:*** nested and non-nested models; qualitative vs. quantitative model comparisons
- ***Nature vs. nurture***: Could humans learn X, given only this information to learn from, and this learning algorithm
- ***Representation testing***
- ***Testing Learning algorithms***

Evolution of Math Modeling

Estes, W. K. (1975). Some targets for mathematical psychology. *Journal of Mathematical Psychology, 12,* 263-282.

In looking back on math modeling, Estes notes:

• We lack cumulative progress
(nobody builds on models, they just build new ones)

• We're modeling tasks, not general processes
(cf., Newell's call for unified theories of cognition)

• We're losing the good scientists b/c they can't understand what we're doing, and why it's important

• Math psyc isn't helpful to any applied problems

• We need to start archiving data

• We can't just fit a descriptive model, we need an explanatory and generalizable model

Evolution of Math Modeling

- We can't separate structure and function
- We need agreed upon methods for model comparison and fit
- We need to evaluate models based on usefulness, not just fit to data (explanatory vs. descriptive models)
- We fit models to highly controlled experiments, and we may be unknowingly hold constant factors which would have a major effect if allowed to vary (Ecological validity vs. control --> Large-scale models)
- What do computers hold for the future of modeling?

It doesn't appear that cognitive scientists paid much attention to Estes points, or we would be in a much better position today…most of these problems with math models are still problems with computer models

Computers and the future

Computers allow us to create theory *"…almost completely bypassing both the opportunities for ingenuity and the quantities of blood, sweat, and tears that used to go into this enterprise*." (p. 268)

Like a gun, a model requires a competent operator (Youth creating theory)

"*Is mathematical psychology in danger of becoming obsolescent before reaching maturity?" (p. 279)*

If you were to write a "Targets for cognitive models" paper today, what wishes would you come up with?

An Existence Proof

Hinitzman D. L. (1986). “Schema abstraction” in a multiple-trace memory model. *Psychological Review, 4,* 411-428.

Goal of Minerva was to explain memory for individual experiences (episodic memory) and memory for abstract concepts within a single system

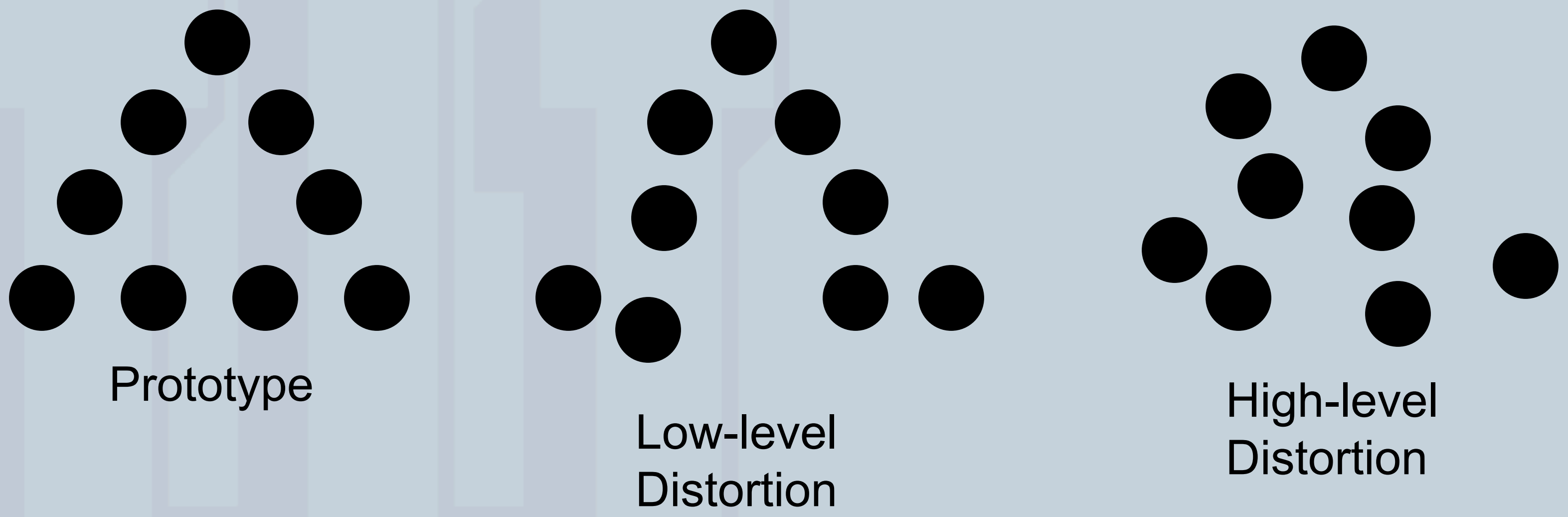
Tulving (e.g., 1983) has argued that we have two separate memory systems, episodic and semantic

Minerva assumes only episodes are stored, and abstract representations are simply a product of how these episodes are activated at memory retrieval

Each exemplar you experience is learned; if you see an exemplar twice, there are simply two of them in memory

Schema Abstraction Task

Posner & Keele (1968)



Exemplars are all distortions of their category prototype

Subjects learn to classify exemplars into A, B, C categories, but never see the prototype (manipulate # exemps per category)

After learning, subjects classify the protoype better than other exemplars, and better than exemplars used in training

Schema Abstraction

Findings:

1) Old exemplars are a bit better than the prototype on an immediate test, but classification for the prototype tends to be much better than the old exemplars with a greater delay
2) Old exemplars are classified better than new exemplars
3) New low-level distortions are classified better than high-level distortions on both immediate and delayed tests
4) Classification of new exemplars is better for large categories (categories w/ more exemplars) than small categories
5) Random patterns are more likely to be assigned to a large category than to a small category

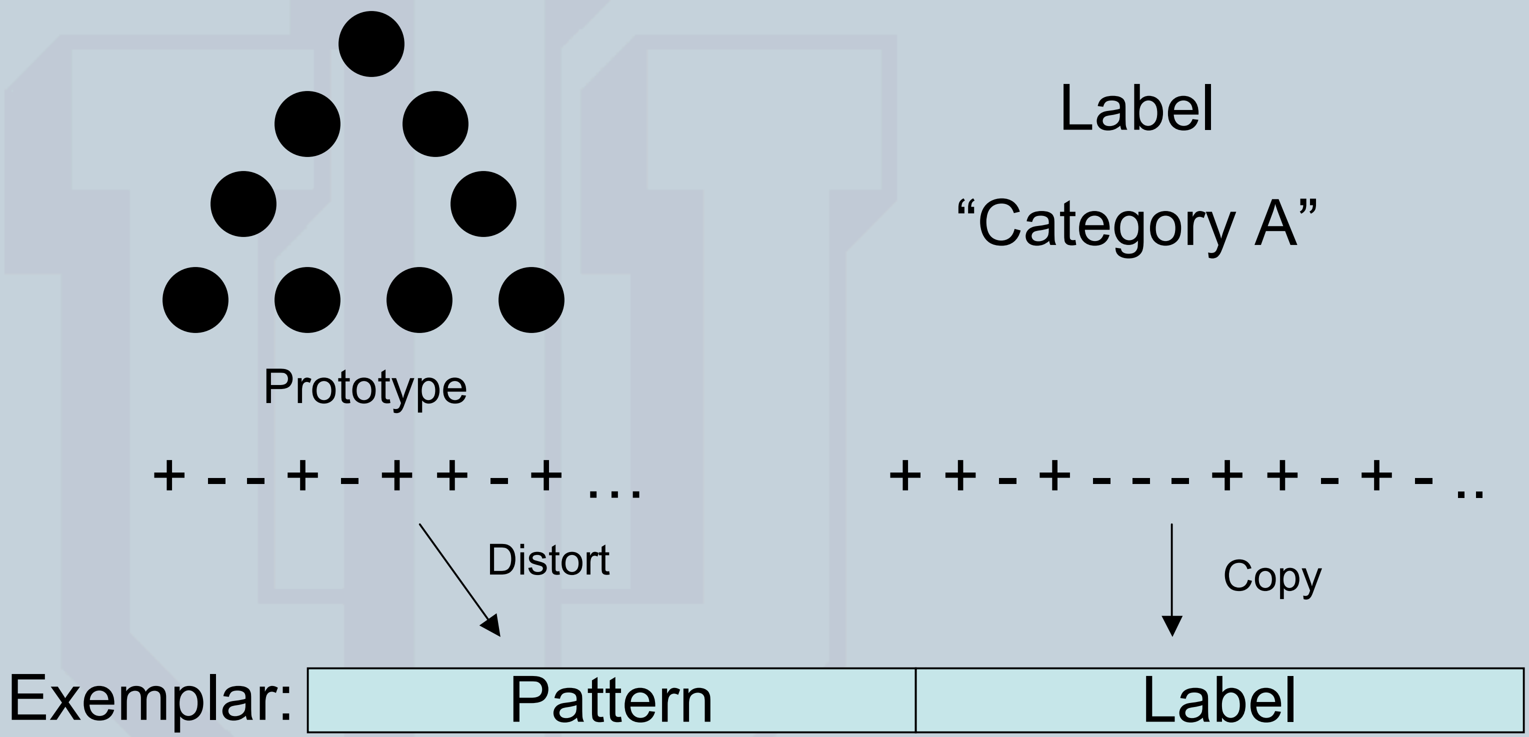
This (especially #1) has been taken as evidence that we create a central “abstract” representation rather than storing exemplars (think of how you learned about dogs or birds)

Minerva: Conceptual Framework

- We only store episodes in memory; there is no need to postulate a separate store for semantic memory
- Memory episodes are lists of descriptive binary features
- When a probe is presented (pattern or category label), it is put in STM, and each episode in memory is activated relative to its similarity to the probe
- A single representation is retrieved from memory and put in the STM buffer; this “Echo” is the aggregate response from all memory instances
- The longer something is in memory, the more it’s representation decays

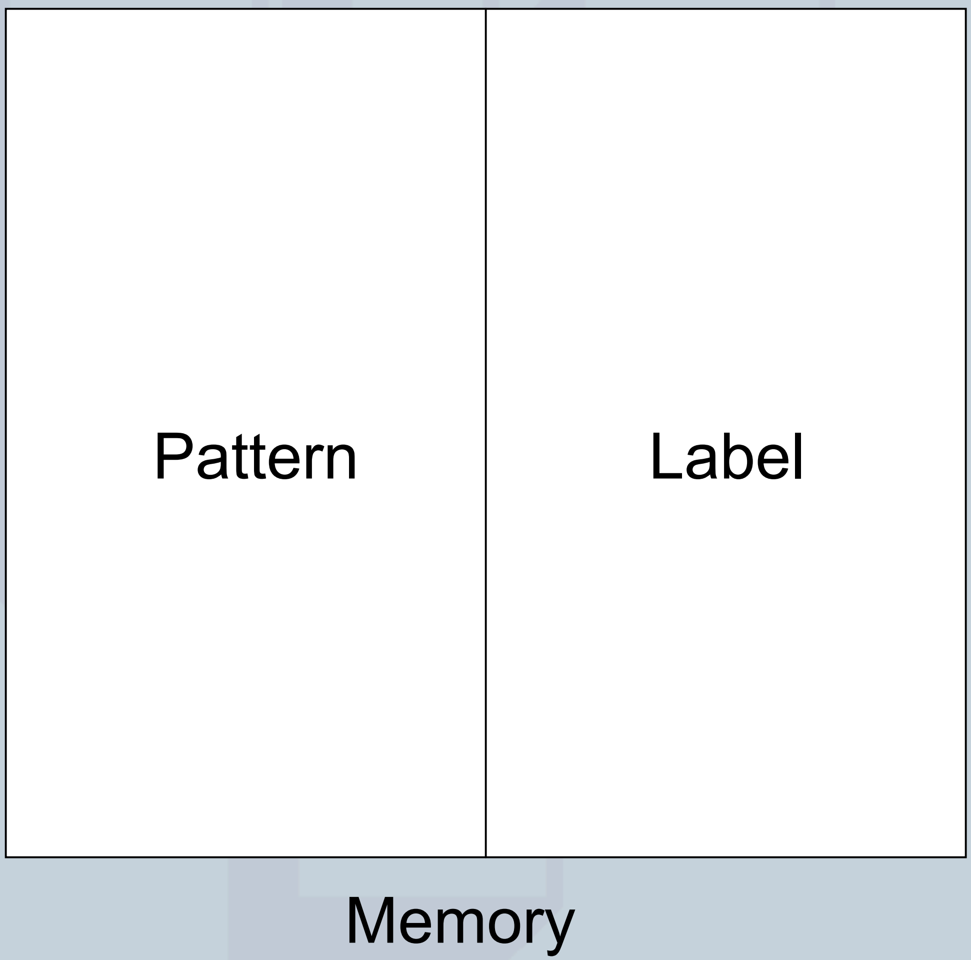
Let’s simulate Posner & Keele’s experiment, using 3, 6, and 9 exemplars for categories A, B, and C

Minerva: Conceptual Framework

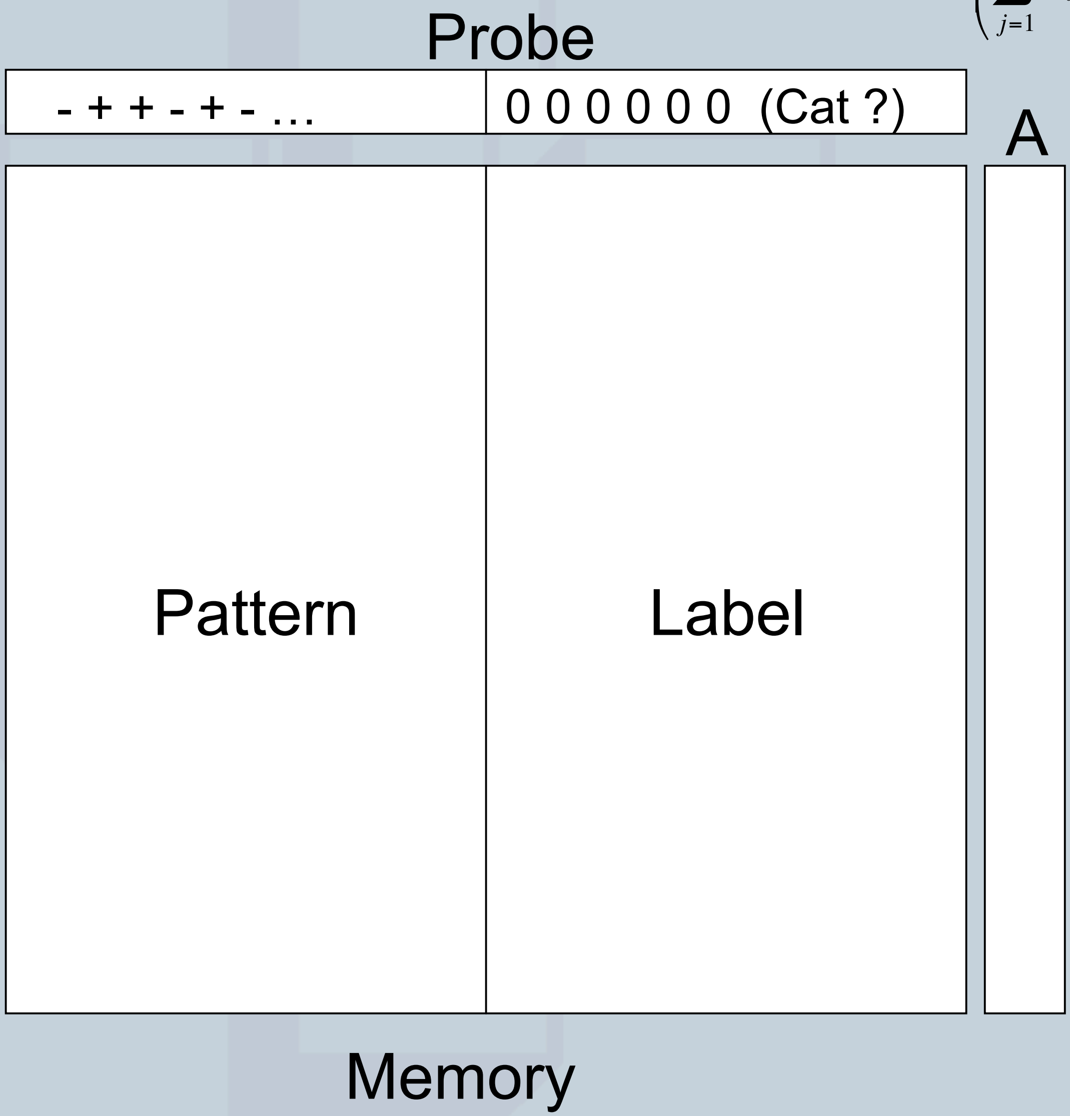


Let’s represent the image and label both w/ binary vectors (n=10)

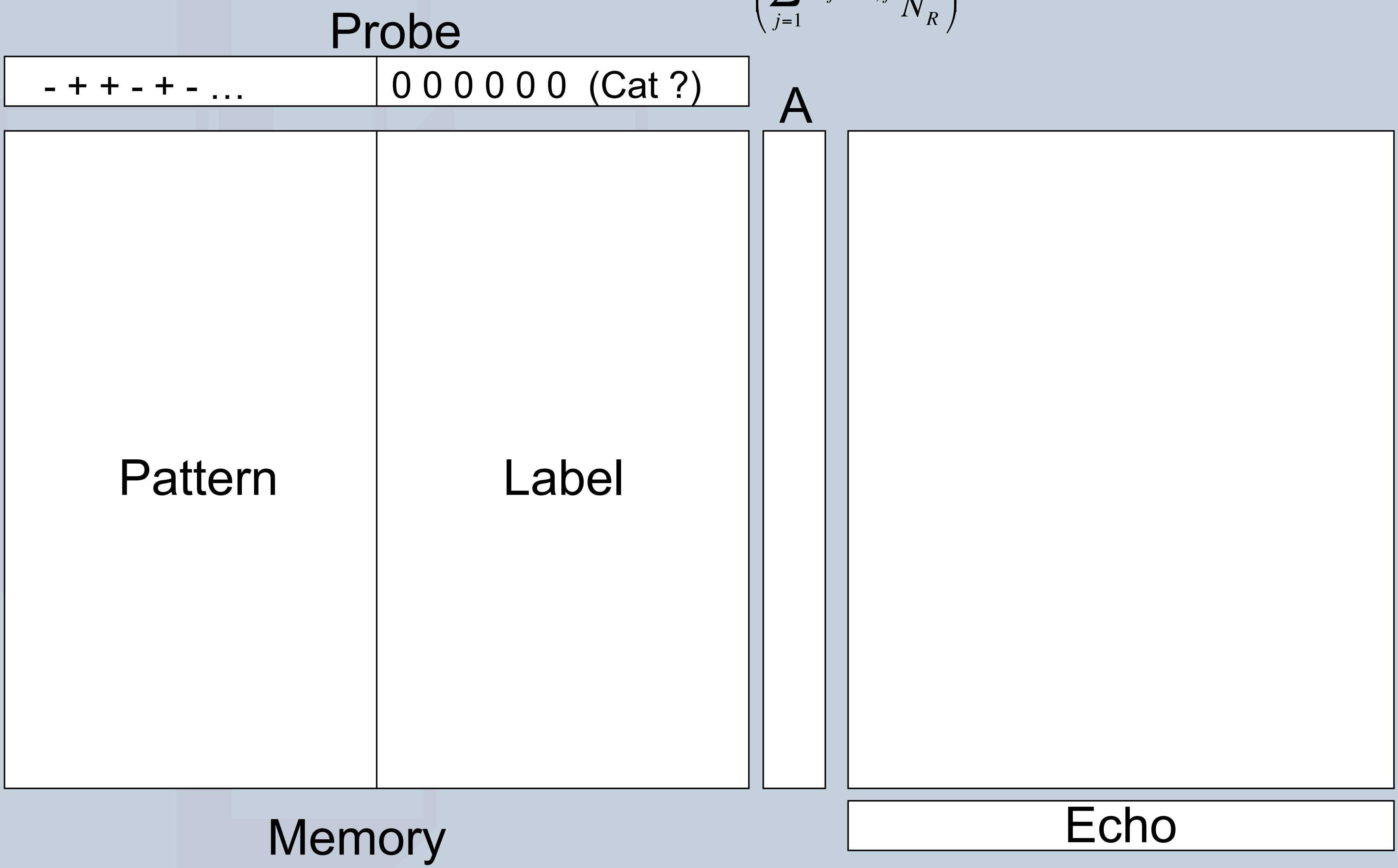
We do this 3, 6, and 9 times for Categories A, B, and C


Pattern
Label
Memory

$$A_i = \left(\sum_{j=1}^{N} P_j M_{i,j} \frac{1}{N_R}\right)^3$$



$$A_i = \left(\sum_{j=1}^{N} P_j M_{i,j} \frac{1}{N_R} \right)^3$$



Parameters:	N = 10	*# of features per vector*
	ITEMS = 18	*total # of items*
	N_CAT = 3	*# of categories*
	F = 0.0	*probability of forgetting a feature*

Data Structures:	**Memory**[2N, Items]	*complete memory matrix*
	Prototype[N, N_CAT]	*the 3 prototype patterns*
	Label[N, N_CAT]	*the 3 labels*
	Echo[2N]	*the echo from memory vector*

A few tools:

Random_Vector(N)	*make a random vector of size N*
Distort(Vector, N, n_flip)	*reverse a n_flip randomly selected features*
Cosine(Vector1, Vector2)	*dot-product of vectors (correlation)*

For each simulated subject:

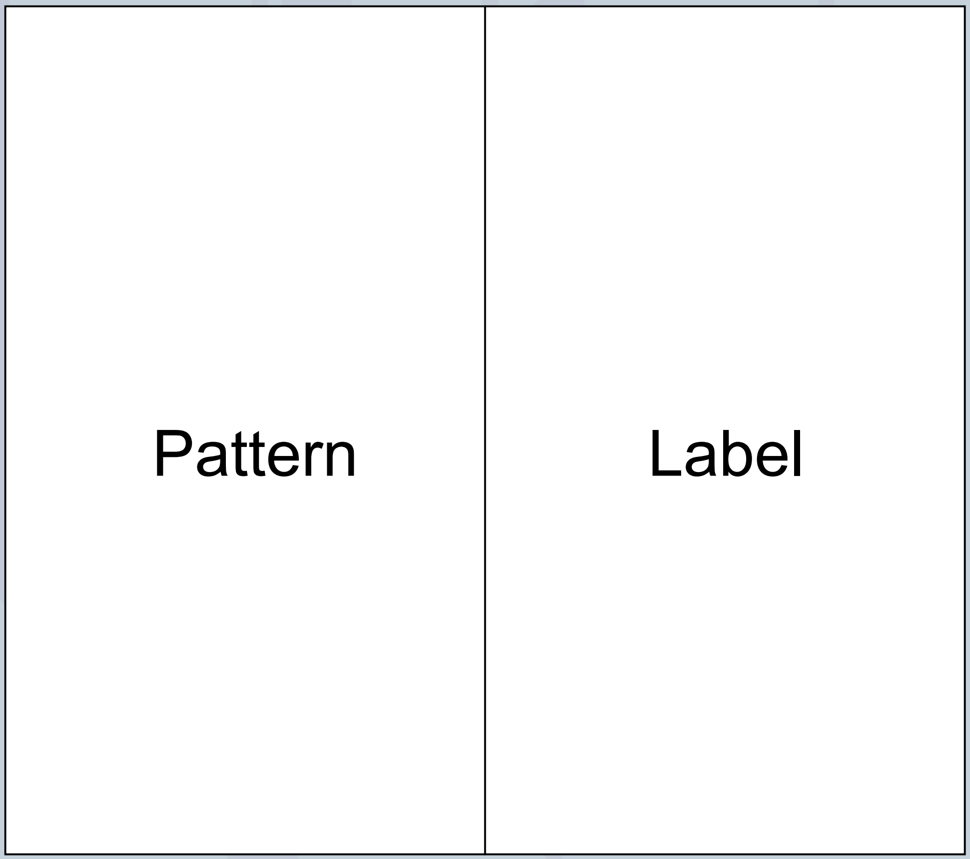
<u>Build Prototypes/Labels:</u>

```
do i = 1..N_CAT
   Prototype[:, i] = Random_Vector(N)
   Label[:, i] = Random_Vector(N)
enddo
```

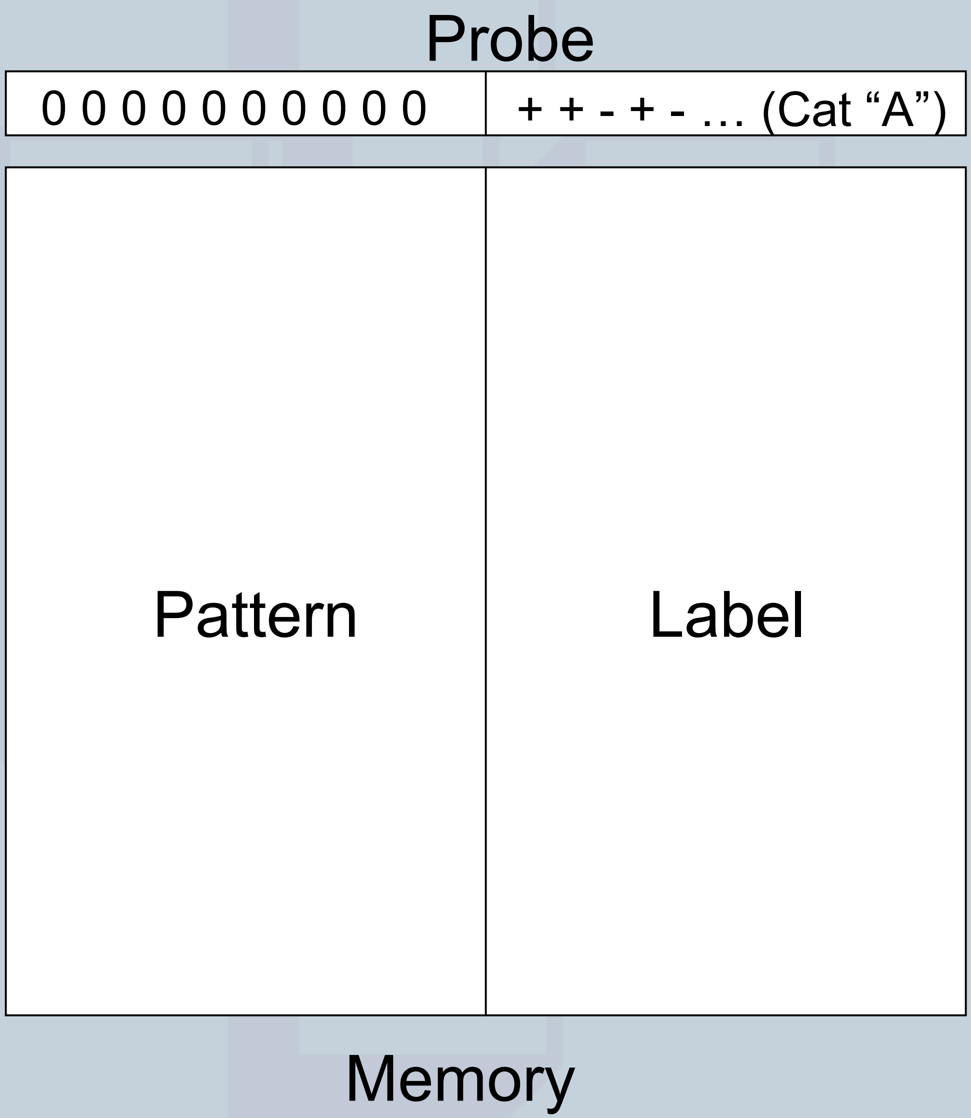
<u>Build Memory:</u>

```
{3 high-level distortions for Category A}
  do i = 1..3
    Memory[1:10, i] = Distort(Prototype[:, 1], N, 4)
    Memory[11:N, i] = Label[:, 1]
  enddo

{6 high-level distortions for Category B}
  do i = 4..9
    Memory[1:10, i] = Distort(Prototype[:, 2], N, 4)
    Memory[11:N, i] = Label[:, 2]
  enddo

{9 high-level distortions for Category C}
  do i = 10..ITEMS
    Memory[1:10, i] = Distort(Prototype[:, 3], N, 4)
    Memory[11:N, i] = Label[:, 3]
  enddo
```

We just built this: (a different memory for each subject)



Memory



Now let’s see what comes out when we probe with a category label

Probe[11:2N] = Label[:, 3]   *Label is "C" pattern is blank*

Create an echo using this category label:

do i = 1..ITEMS

  Activation = Cosine(Probe, Memory$_i$) ** 3   *this is basically*: $A_i = \left(\sum_{j=1}^{N} P_j M_{i,j} \frac{1}{N_R} \right)^3$

  Echo = Echo + (Memory$_i$ * Activation)

enddo

Now let's compare the pattern retrieved from memory (in the echo) when probed with "C" to the prototype pattern for Category C:

Cosine(Echo, Prototype[:, 3]) = 0.67

Versus all the 9 Category "C" exemplars learned:

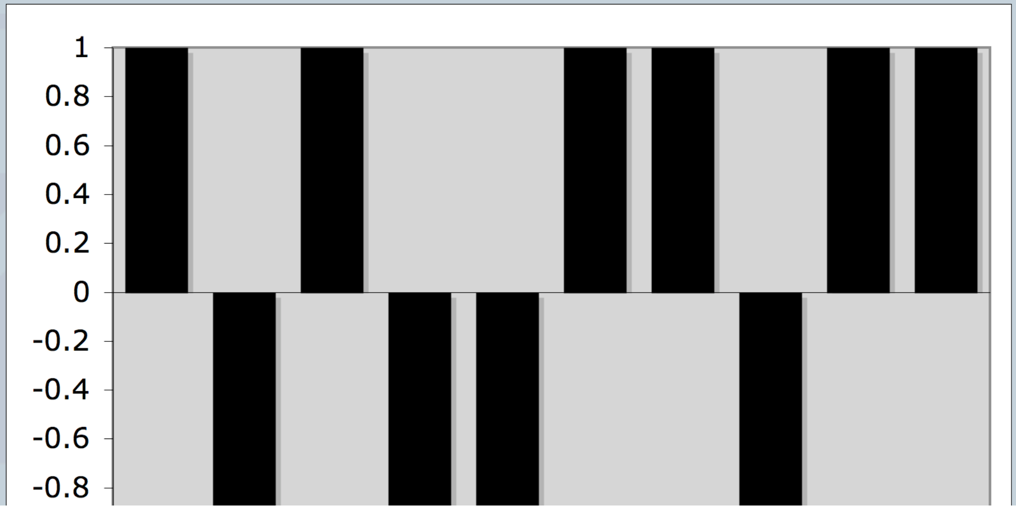
do i = 10, ITEMS

  Cosine(Echo, Memory[1:N])

enddo

0.43, 0.62, 0.48, 0.34, 0.60, 0.56, 0.45, 0.62, 0.48

“C” Prototype:



Minerva can retrieve an “abstract” idea, even though an abstraction was never stored

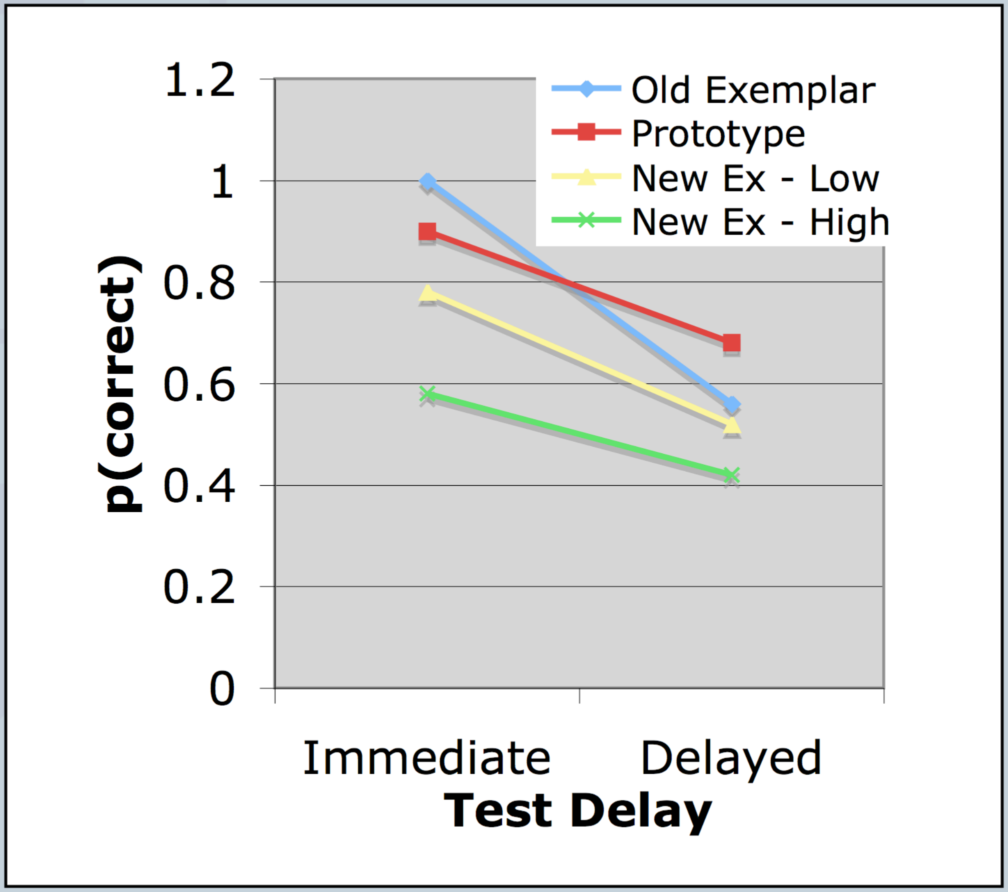
Echo:

The echo is more like the prototype and less like learned exemplars the larger the category (more exemps stored)

- The *F* parameter represents forgetting…a feature in memory remains with a p(1-F) and reverts back to 0 with a p(F)
- we left *F* = 0 in our simulation, but it makes sense for *F* to increase with time. Let's simulate the immediate vs. delayed test in Posner & Keele's experiment. (*F* = 0 or 0.75)

For Category B (6 exemplars)

Interacts w/ category size



In fact, Minerva can account for all of Posner & Keele's findings:

- Old exemplars > prototypes on immediate test, but prototype classification gets better than old exemplars w/ delay
- Old exemplars > than new exemplars on both immediate and delayed test
- New low-level distortions > new high-level distortions
- New exemplars are classified better for large categories than small categories
- Random patterns are more likely to be assigned to a large category than a small one
- Minerva also accounts for a variety of effects from recognition memory (recognition-failure of recallable words, etc.)

Note, however, that it is flawed as a model of categorization…I use it here as an existence proof of abstraction without the need for a separate memory system (we'll see better classification models shortly)

Let’s look at the code and manipulate the parameters:

Minerva.f95 will simulate retrieval of a pattern given a label, or classification of a pattern (retrieve a label given a pattern)

Forgetting.f95 simulates classification of a pattern (old, new-hi, new-lo, prototype, random) as a function of delay

Both use the module number_generators; to compile, simply type:

f95 Number_Generators.f90 Minerva.f95 -o Minerva

And type: “Minerva” to run

OR

f95 Number_Generators.f90 Forgetting.f95 -o Forget

And type: “Forget” to run