
Lecture 10: Computational Cognitive Modeling

Categorization, Classification, and Concepts

email address for instructors:
instructors-ccm-spring2020@nyucll.org

course website:
<https://brendenlake.github.io/CCM-site/>

categorization: where human and machine learning meet

object recognition (ImageNet)



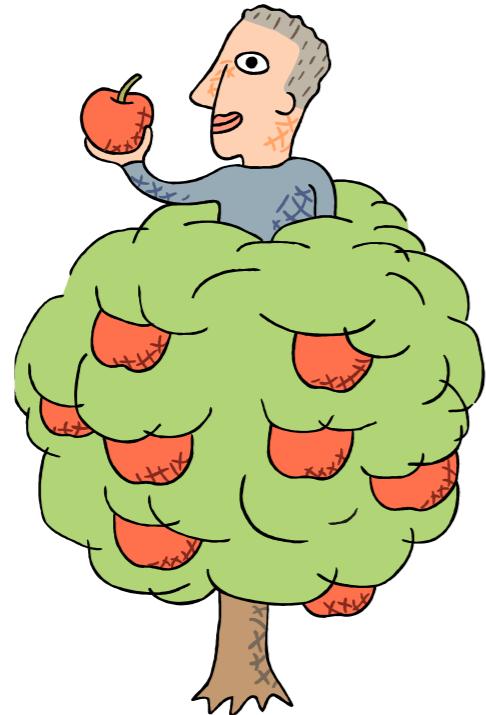
digit recognition (MNIST)



- **classification is a central problem in machine learning** (what category does this image show? what topic does this document best fit?)
- many important algorithms developed for this problems (e.g., decision trees, support vector machines, bayes classifiers, deep neural networks, hidden markov models, etc...)
- **what algorithms best characterize how people learn to categorize?**
- also, the goal of machine learning systems is to categorize things in ways that appear sensible to people: but what principles inform human categorization? what makes a good category from the perspective of a person?

what is the purpose of categorization (for humans)?

Categories have many functions:



- **Classification** - allows us to treat different things as the same
- **Communication** - we communicate using words that refer to more abstract ideas/concepts
- **Prediction and reasoning** - we can use categories to make predictions about unknown or unseen parts of the world

What you see:

Red
Shiny
In a tree → Apple

What you can then infer:

Has seeds
Sweet
Edible
Healthy

the machine learning framework

apply a prediction function to a feature representation of image to get the desired output:

$$f(\text{apple}) = \text{"apple"}$$

$$f(\text{tomato}) = \text{"tomato"}$$

$$f(\text{cow}) = \text{"cow"}$$

the machine learning framework

$$y = f(x)$$

Diagram illustrating the machine learning framework:

- output**: The final result, represented by the variable y .
- prediction function**: The function that maps input features to the output, represented by f .
- Image feature**: The input to the prediction function, represented by the variable x .

```
graph TD; y["y"] --> output["output"]; f["f(x)"] --> prediction["prediction function"]; x["x"] --> prediction; x --> image["Image feature"];
```

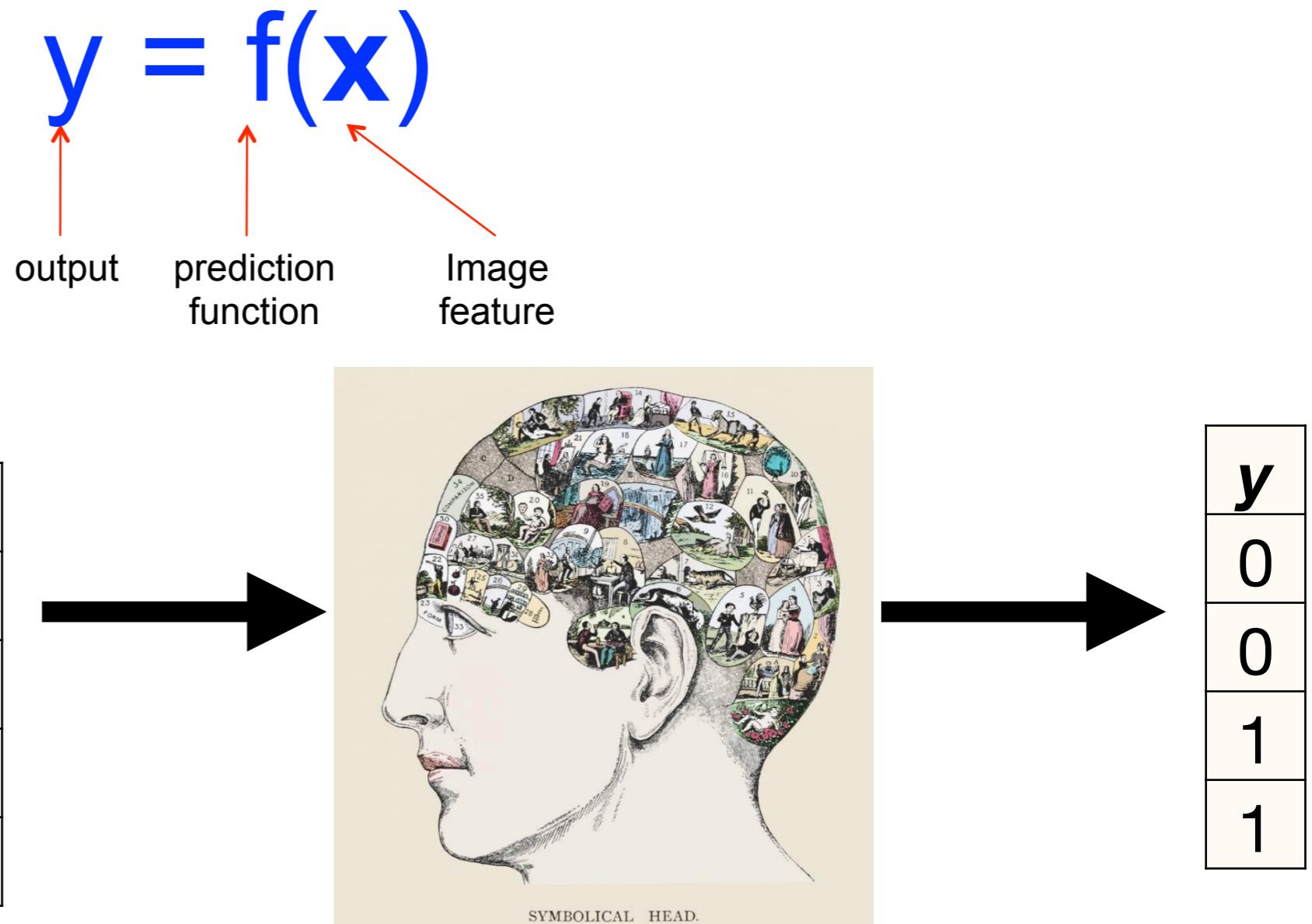
Training: given a *training set* of labeled examples $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set

Testing: apply f to a never before seen *test example* \mathbf{x} and output the predicted value $y = f(\mathbf{x})$

the human cognition framework

What is the function $y = f(x)$ that best characterizes how people make categorization decisions?

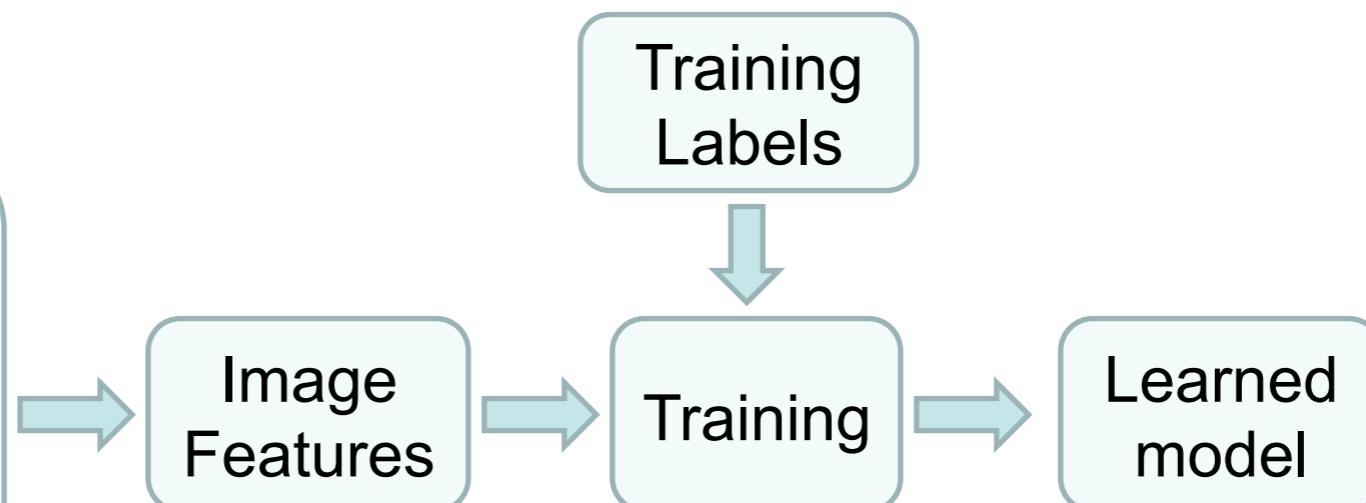
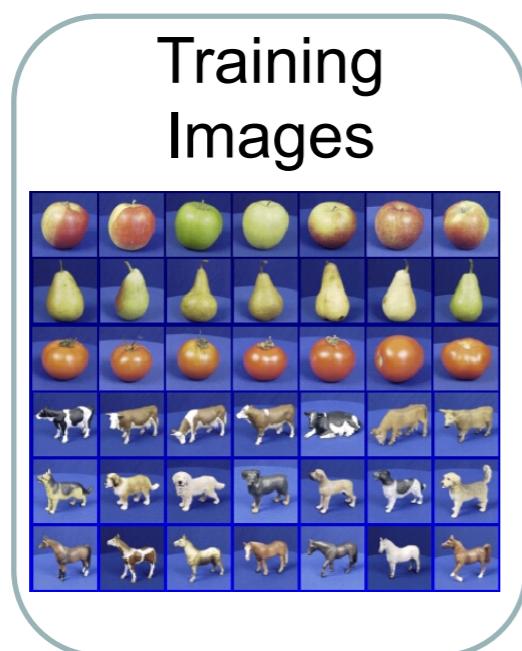
x_0	x_2	x_3	x_4	x_5	x_6	x_7
0	0	0	1	0	0	0
0	1	0	0	1	0	1
1	0	0	1	0	0	0
1	1	1	1	0	1	1



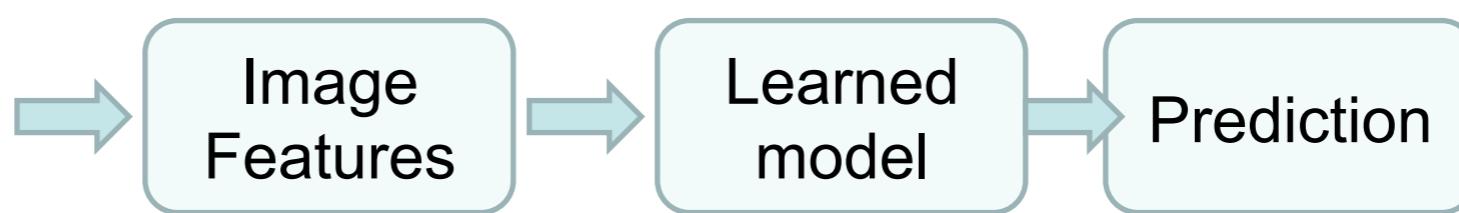
the machine learning framework

Steps

Training



Testing



Slide credit: D. Hoiem and L. Lazebnik

the human cognition framework

training stimuli

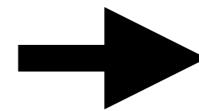
Name	Concept	Pack I	Pack II	Pack III	Pack IV	Pack V	Pack VI
oo	✓	建	拂	泳	泡	铁	漆
yer	古	俎	犮	珍	殆	砾	砾
li	力	助	勳	勑	勑	勑	勢
ta	弓	弦	弧	弔	弗	隨	罿
deg	石	石	青	磚	角	堅	石
ling	穴	空	審	寃	寃	寃	寃



Example of *Builder*

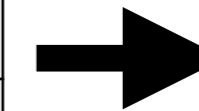


Example of *Digger*

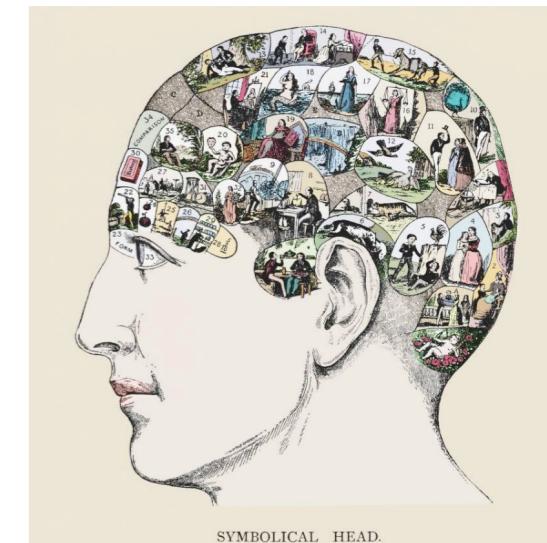


Some representation in terms of
psychologically meaningful features

X_0	X_2	X_3	X_4	X_5	X_6	X_7
0	0	0	1	0	0	0
0	1	0	0	1	0	1
1	0	0	1	0	0	0
1	1	1	1	0	1	1



some REPRESENTATION
of the category

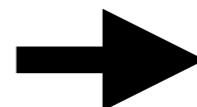


also examine aspect of the
learning (how mistakes are made,
learning rates, etc...)

probe the nature
of the representation
often by designing new
stimuli



Example of *Builder*



prediction

generalization is everything!



training set (labels known)

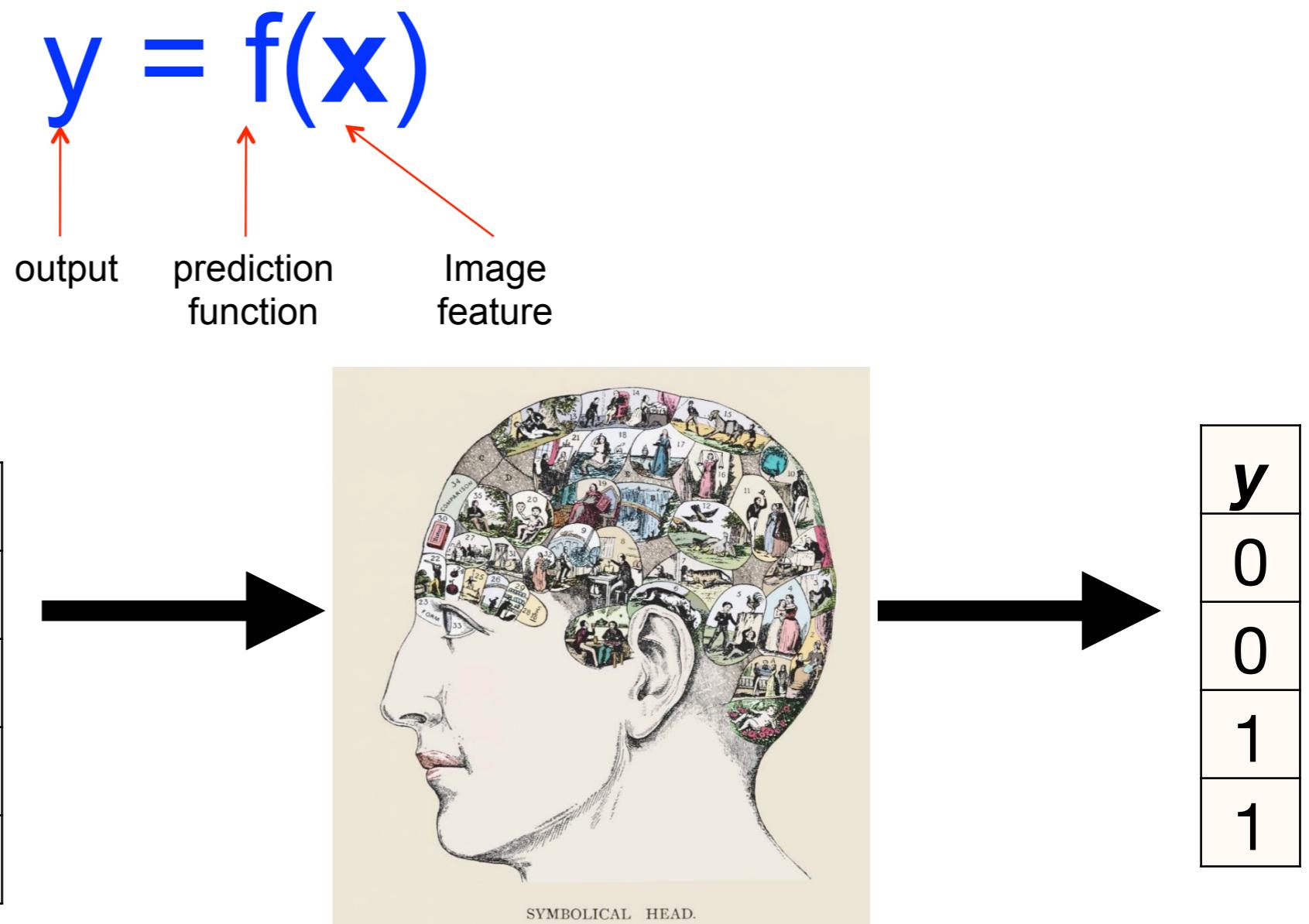


test set (labels unknown)

- Data science: How well does a learned model generalize from the data it was trained on to a new test set?
- Psychology: What types of generalizations do people make? What does that reveal about how they learn?

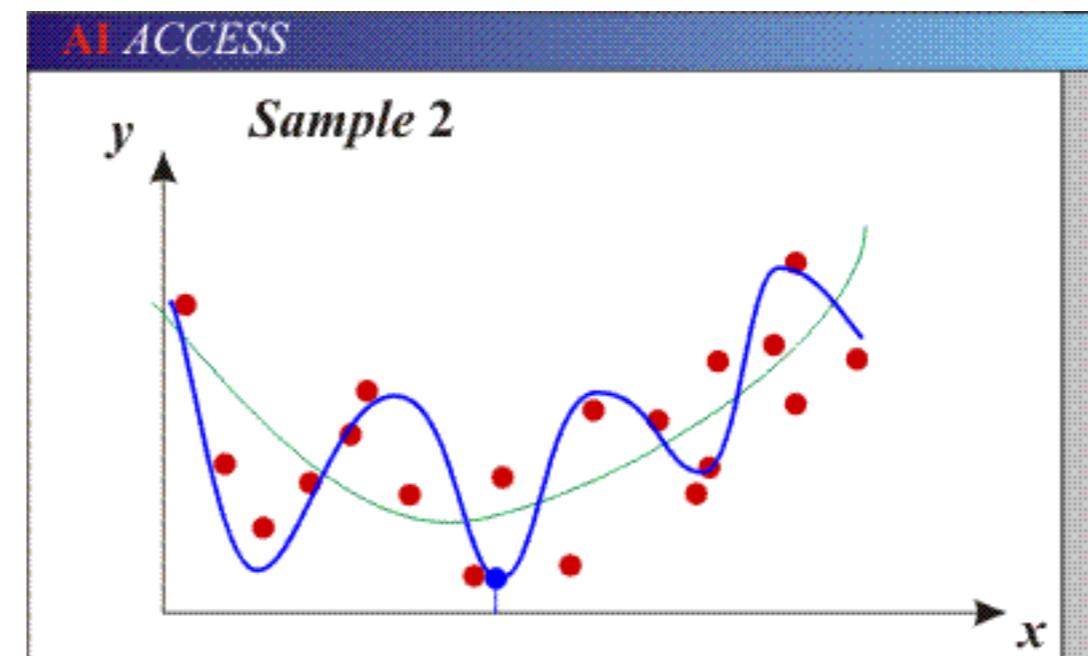
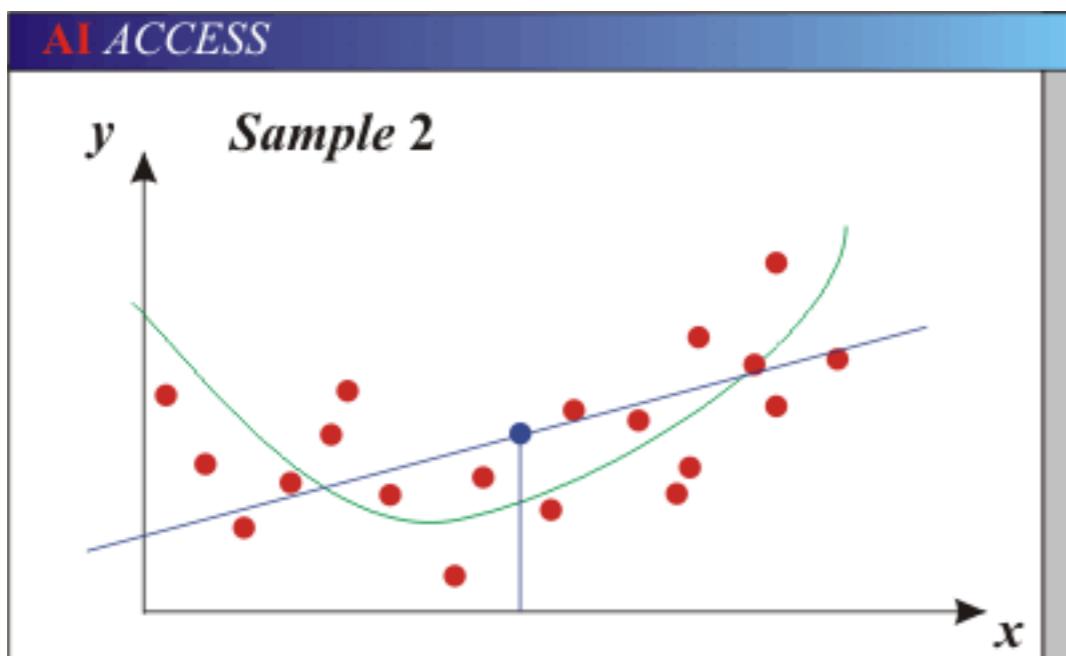
why not make function $y = f(x)$ include all possible generalizations and just pick the ones consistent with the evidence?

x_0	x_2	x_3	x_4	x_5	x_6	x_7
0	0	0	1	0	0	0
0	1	0	0	1	0	1
1	0	0	1	0	0	0
1	1	1	1	0	1	1



generalization is everything!

- Components of generalization error
 - **Bias**: how much on average does model over all training sets differ from the true model?
 - Error due to inaccurate assumptions/simplifications made by the model
 - **Variance**: how much models estimated from different training sets differ from each other
- **Underfitting**: model is too “simple” to represent all the relevant class characteristics
 - High bias and low variance
 - High training error and high test error
- **Overfitting**: model is too “complex” and fits irrelevant characteristics (noise) in the data
 - Low bias and high variance
 - Low training error and high test error



bias-variance tradeoff

$$E(\text{MSE}) = \text{noise}^2 + \text{bias}^2 + \text{variance}$$

Unavoidable error

Error due to
incorrect
assumptions

Error due to variance
of training samples

- If we predicted constant value on every trial the variance would be zero across different training sets. However, bias would be huge because model would never predict training data well.
- If we perfectly fit each training set (overfit), bias goes away completely. However, the variance term will be equal to the noise in the data which can be really big.
- Optimal balance to these issues is difficult but is addressed to some degree via model evaluation methods (see later lecture) such as cross validation and regularization.

See the following for explanations of bias-variance (also Bishop's "Neural Networks" book):

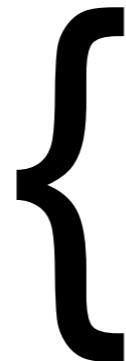
• <http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf>

an argument for the need for biases

If totally unbiased generalization systems are incapable of making the inductive leap to characterize the new instances, then **the power of a generalization system follows directly from its biases – from decisions based on criteria other than consistency with the training instances.** Therefore, progress toward understanding learning mechanisms depends upon understanding the sources of, and justification for, various biases.

Mitchell (1980)

possible biases



Domain knowledge

Intended use/goal of generalization (e.g., cost of being incorrect... i.e., risk sensitive)

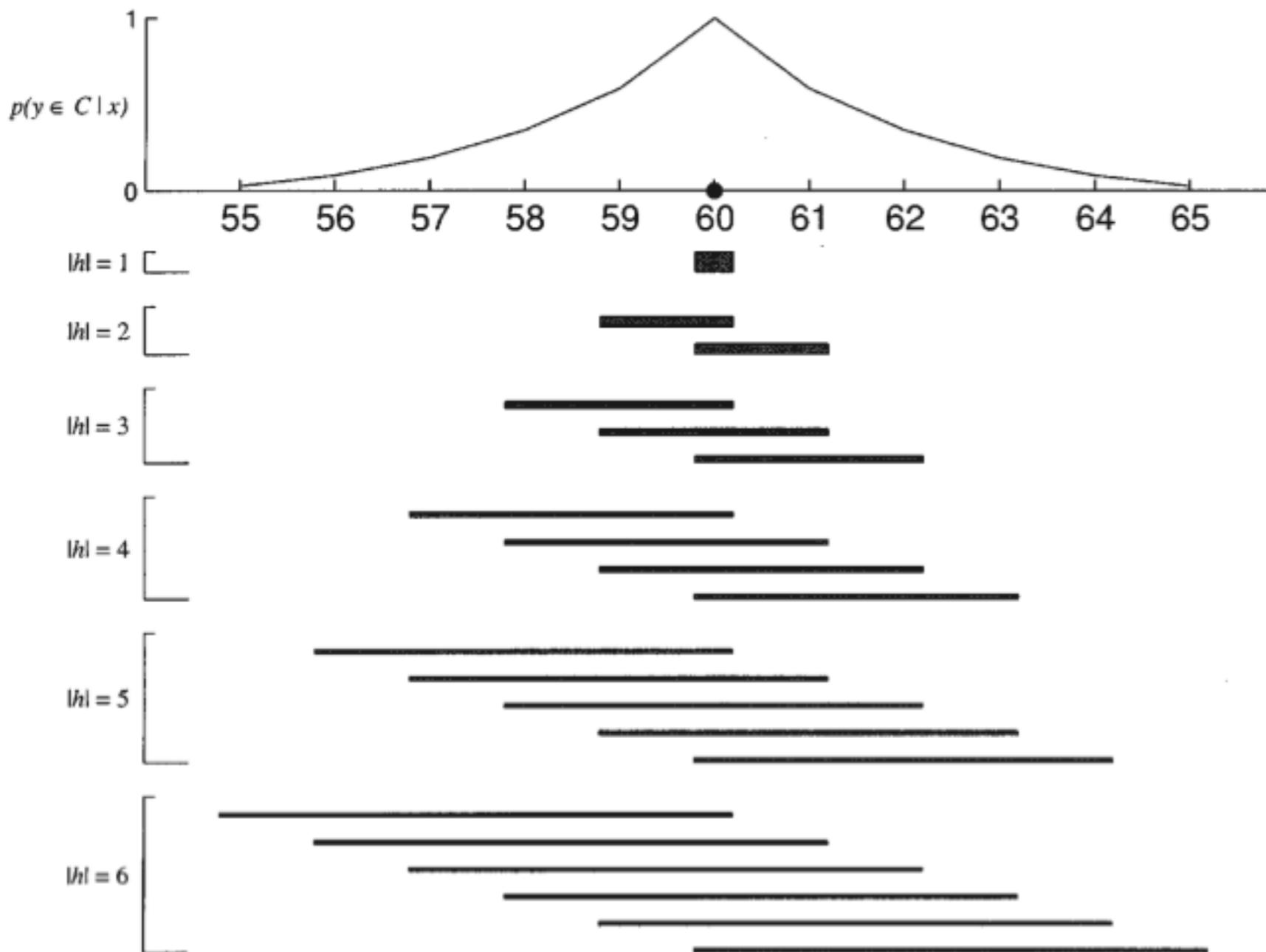
Knowledge about the source of training data

Biases towards simplicity/generality

Analogy with previous generalizations

an example in exploring inductive biases

Generalization, similarity, and Bayesian inference



Joshua B. Tenenbaum and Thomas L. Griffiths

Department of Psychology, Stanford University, Stanford, CA 94305-2130

jbt@psych.stanford.edu gruffydd@psych.stanford.edu

<http://www-psych.stanford.edu/~jbt>

<http://www-psych.stanford.edu/~gruffydd/>

$$p(h|x)$$

$$h \in H$$

$$p(y \in C | x) = \sum_{h: y \in h} p(h|x)$$

$$\begin{aligned} p(h|x) &= \frac{p(x|h)p(h)}{p(x)} \\ &= \frac{p(x|h)p(h)}{\sum_{h' \in \mathcal{H}} p(x|h')p(h')} \end{aligned}$$

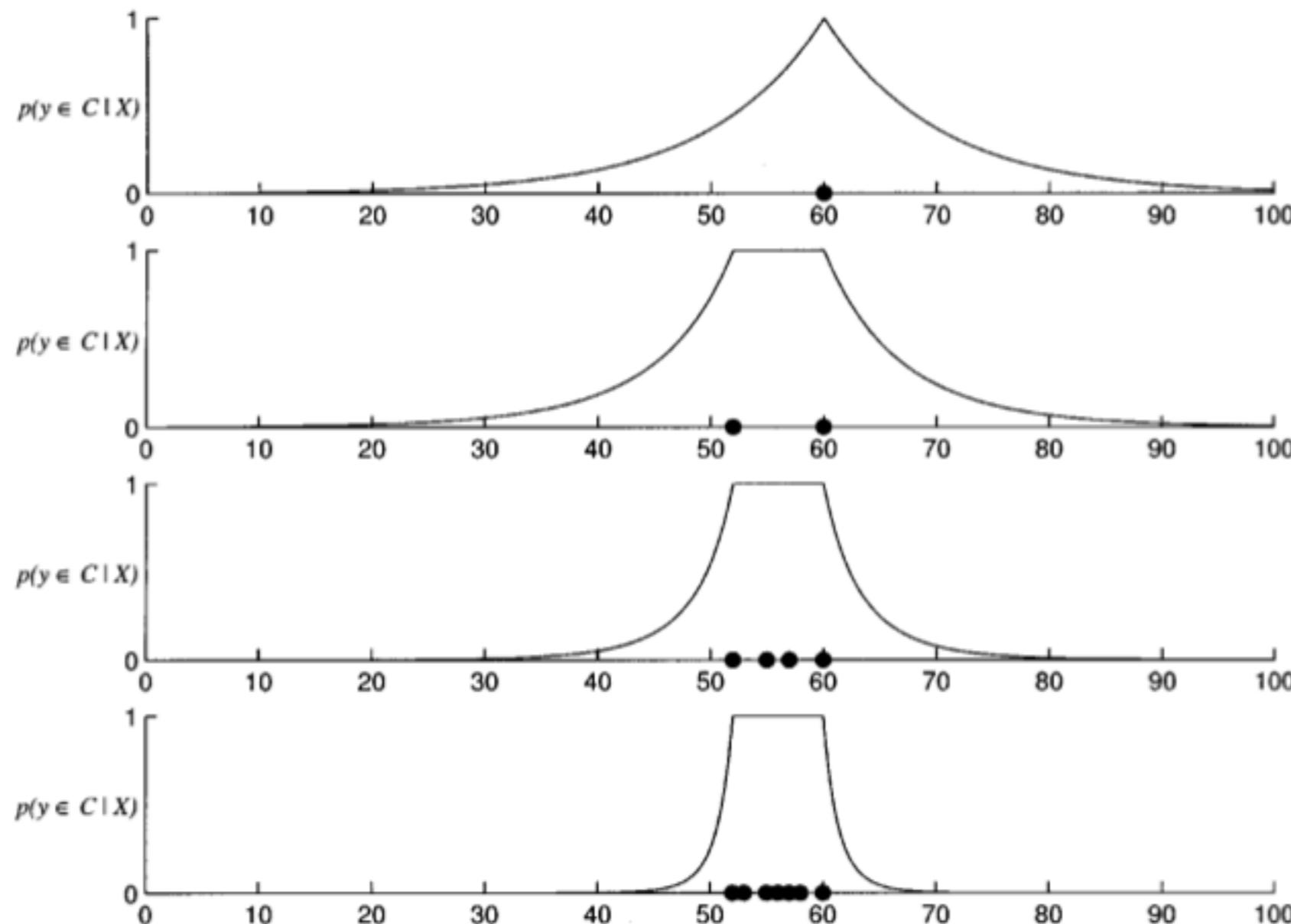
the need for biases

- What is learned will depend on the learners *assumptions* about the situation.
- Where the data generated from the true concept or were they generated at random (independent of the concept?)
- This map onto the notion of STRONG versus WEAK sampling

$$p(x|h) = \begin{cases} 1 & \text{if } x \in h \\ 0 & \text{otherwise} \end{cases} \quad [\text{weak sampling}].$$

$$p(x|h) = \begin{cases} \frac{1}{|h|} & \text{if } x \in h \\ 0 & \text{otherwise} \end{cases} \quad [\text{strong sampling}],$$

as data accumulates in one region of the space you start becoming more confident about the concept (sharper boundaries)



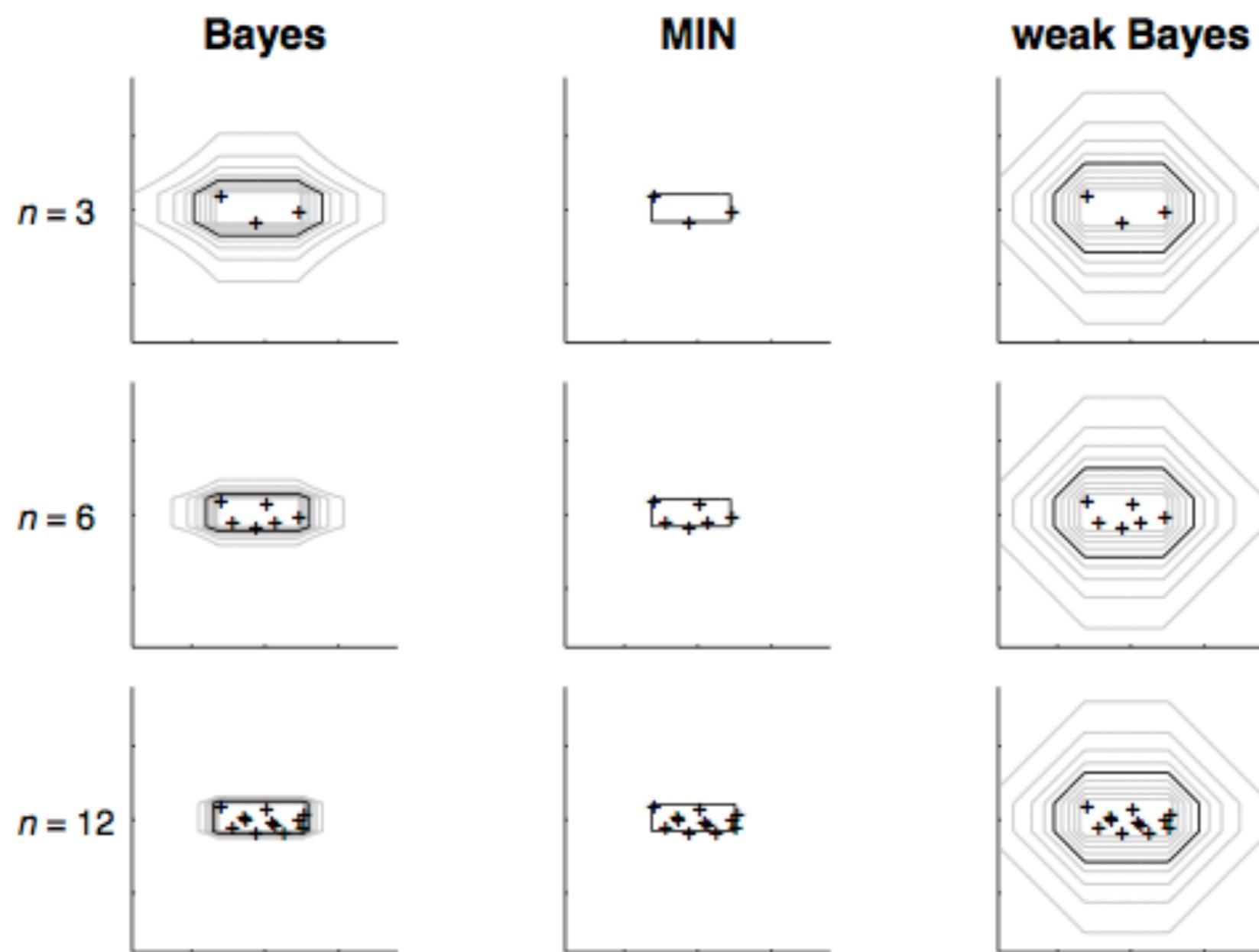
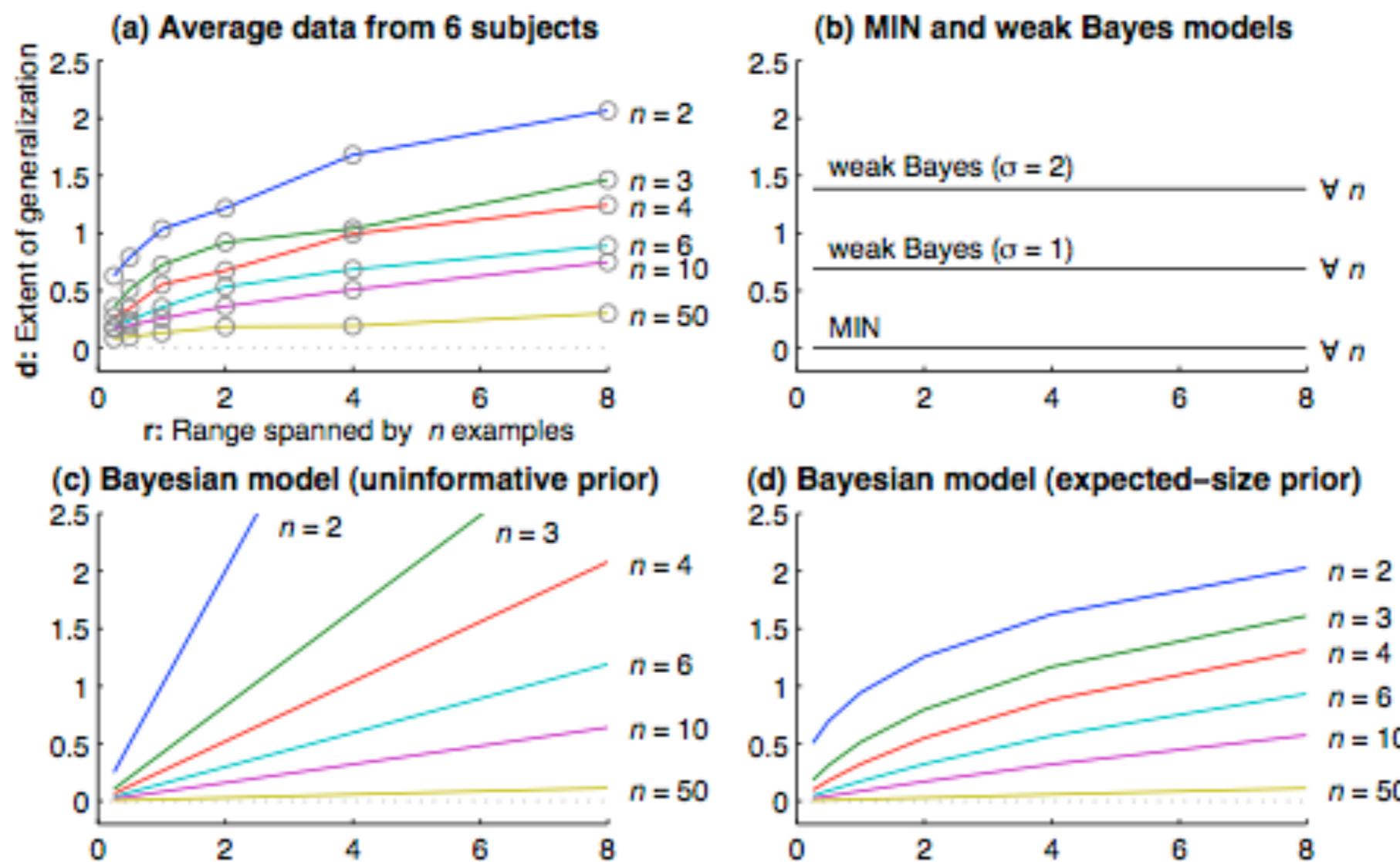


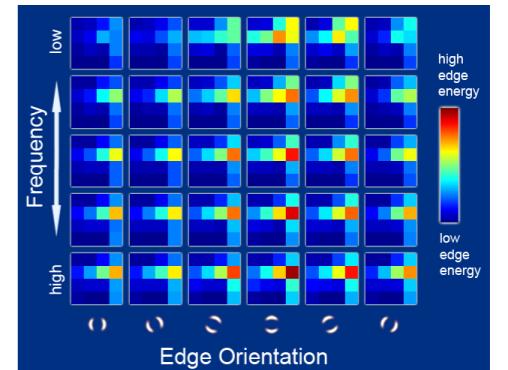
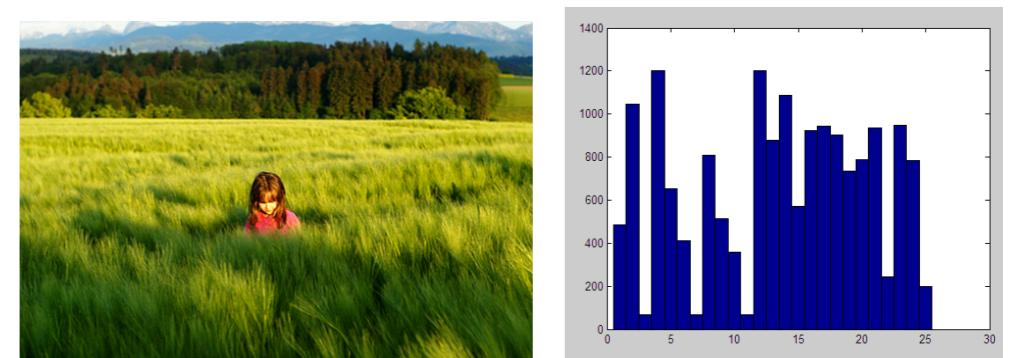
Figure 2: Performance of three concept learning algorithms on the rectangle task.

humans look like they expect generalizations to be more favorable according to the “expected size” prior... meaning they prefer some generalizations over others.... in other words a **inductive bias**.



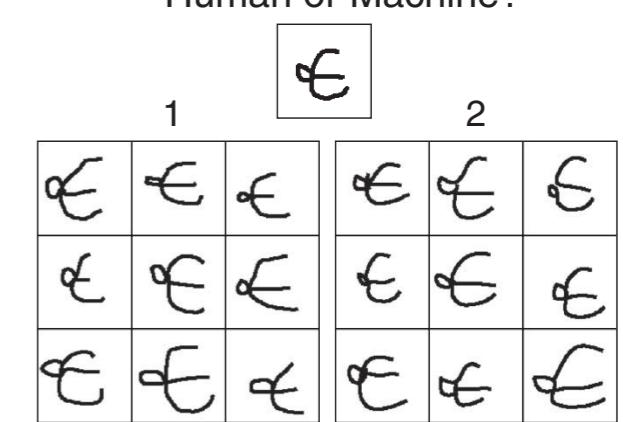
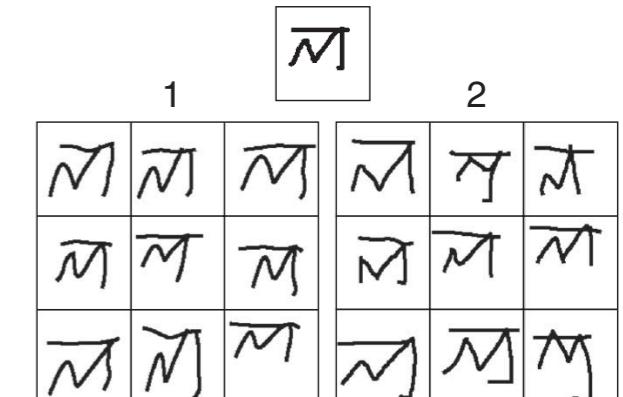
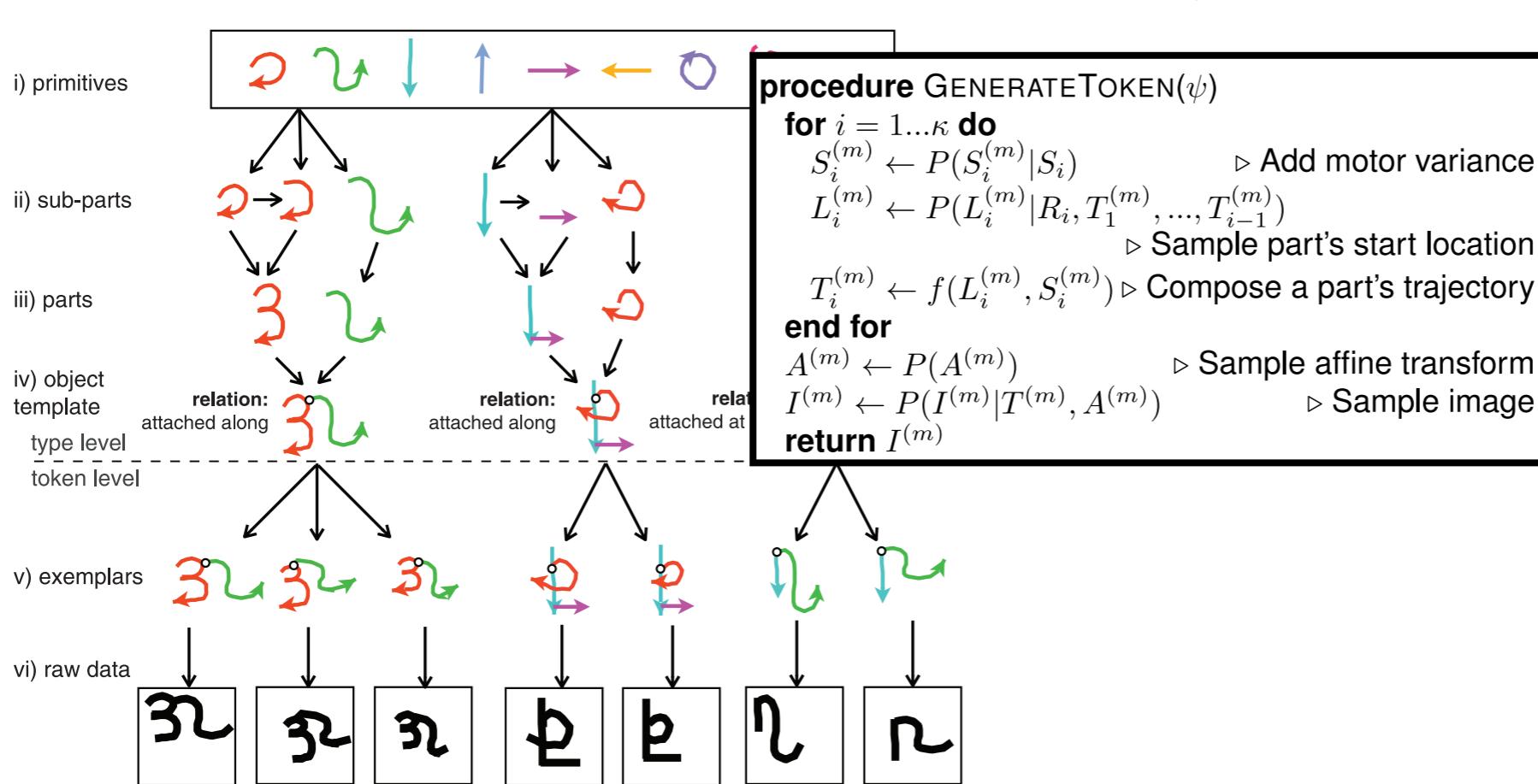
the machine learning framework - what are the features?

- performance often influenced by the nature of the input representation
- raw pixels of an image?
- histograms of intensities or other derived features (e.g., line orientations in local patches, etc...)
- GIST descriptors?
- discrete/symbolic features? has_wings, can_fly, is_a_mammal



the human cognition framework - what are the features?

new perceptually learned features



semantic features

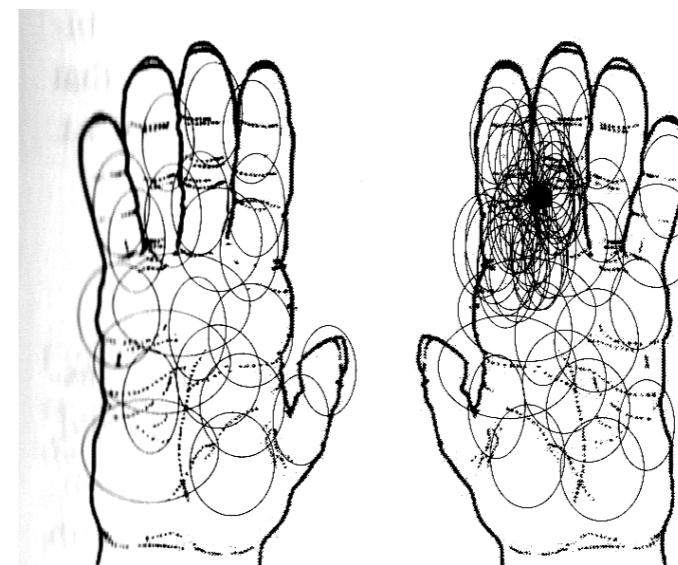
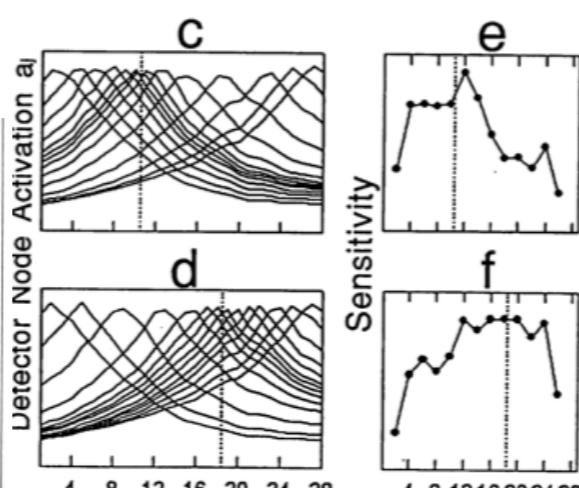
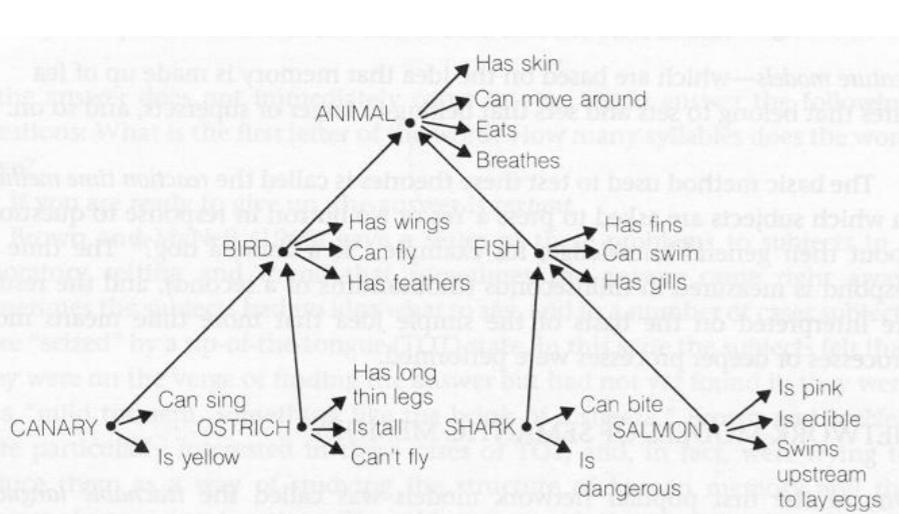
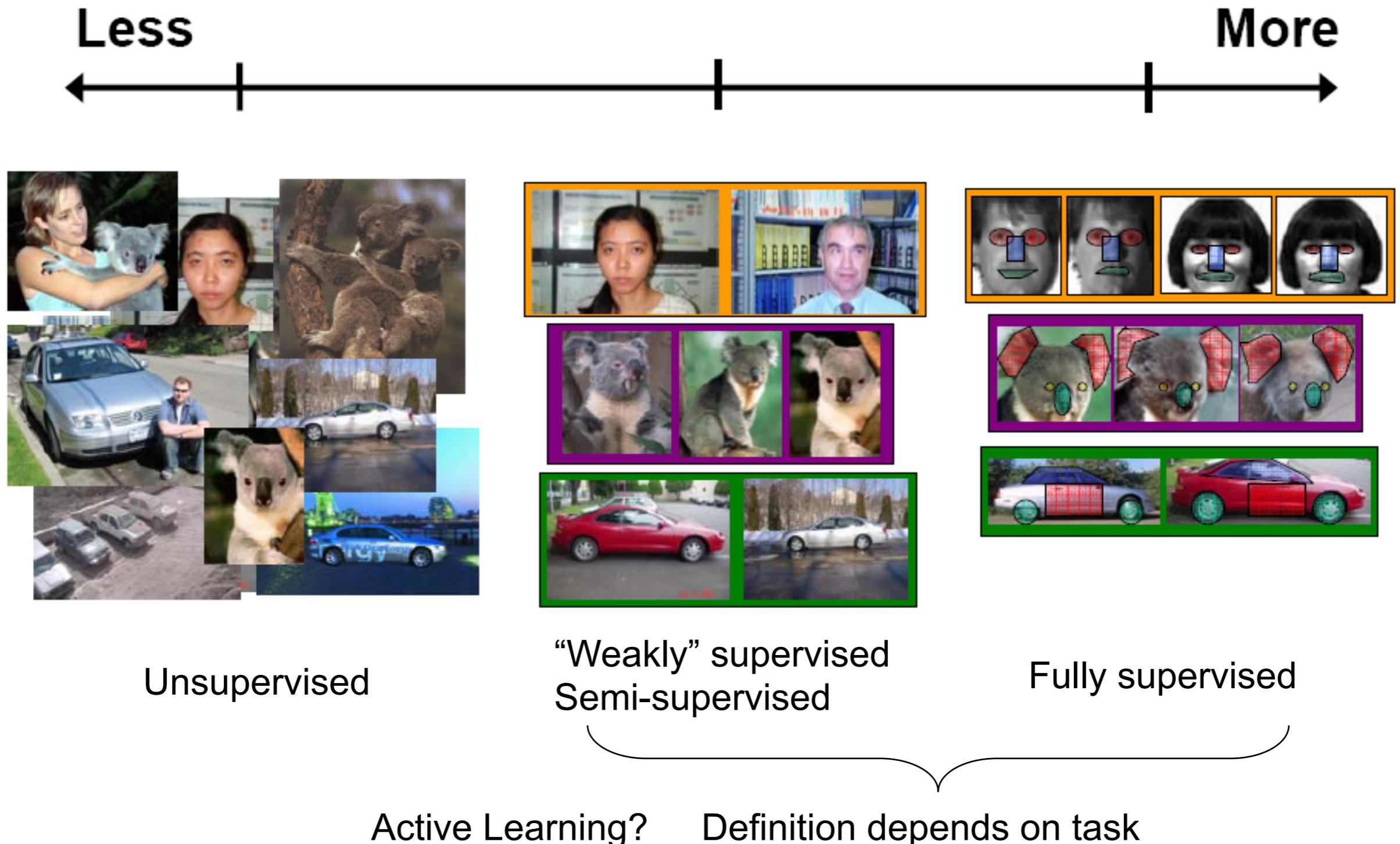


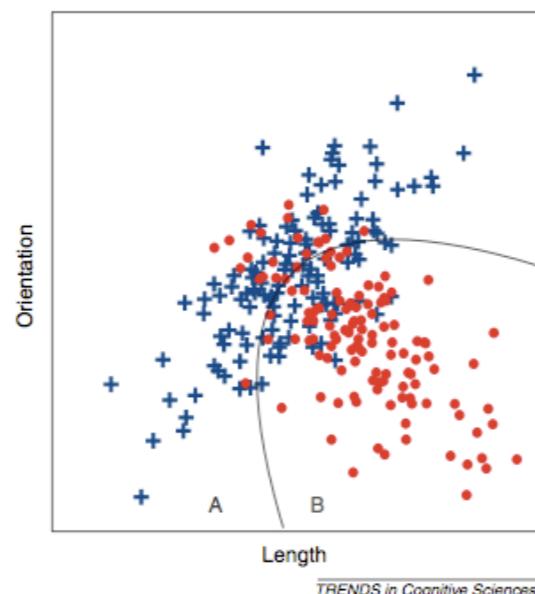
Figure 3.6 — COR fields of somatosensory palm surface, after the arrow indicates the stimulated hand is shown repeatedly at the depicted location.

Spectrum of supervision



Where do the training examples come from?

- In machine learning applications you want your training examples for estimating $y = f(x)$ to represent what you will ultimately be tested on. Also usually limited by data availability/
- Training sets that don't resemble what the system will be asked to do in engineering practice strongly contribute to expected error.
- **What should they be for human cognition studies?** Representative of the type of categories that exist in the world? How do we assess that? What is the statistics of natural concepts and categories and how can we ensure that these are reflected? Also somewhat limited by data availability.



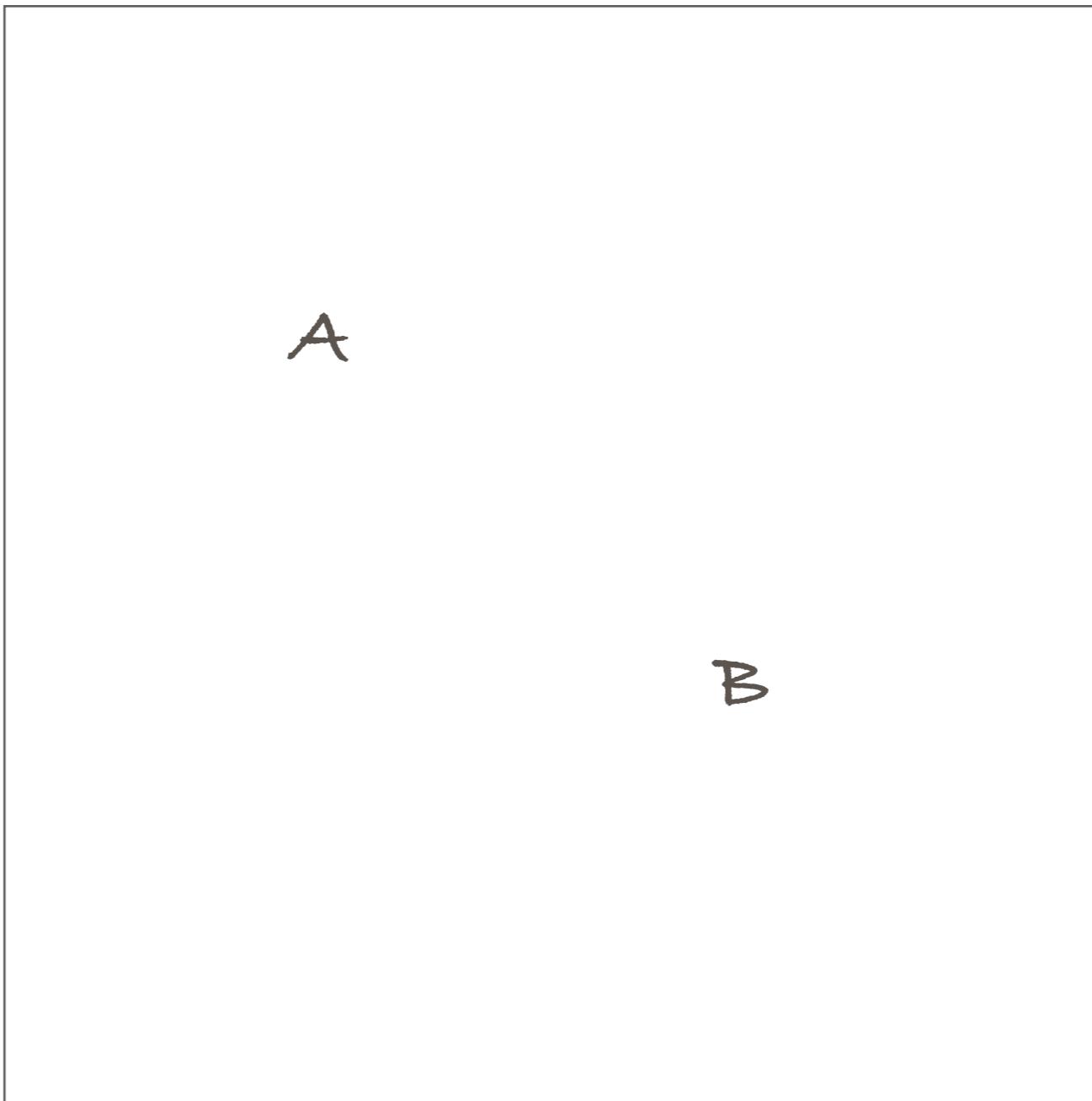
Active Learning

- The basic problem is this: you want to train a machine learning system to assign items to a category (for example diagnosing some biological samples as toxic or not). However, **getting corrective, supervised feedback is expensive** (usually involves humans)
- How can you choose samples to minimize the amount of feedback you need, while still having good categorization accuracy/generalization? In other words, is there a way to train on only a subset of the available data in order to optimize learning?
 - Classic work on this is Lindley (1956) “On the measure of the information provided by an experiment” - Optimal Experiment Design
 - David Mackay (1992) provides a Bayesian formulation for active sampling for function approximation, interpolation, classification, etc...
 - Settles (2010) provides an extensive review
 - see Gureckis & Markant (2013, Perspectives in Psych Sci) for an overview

Active Learning

A

Active Learning



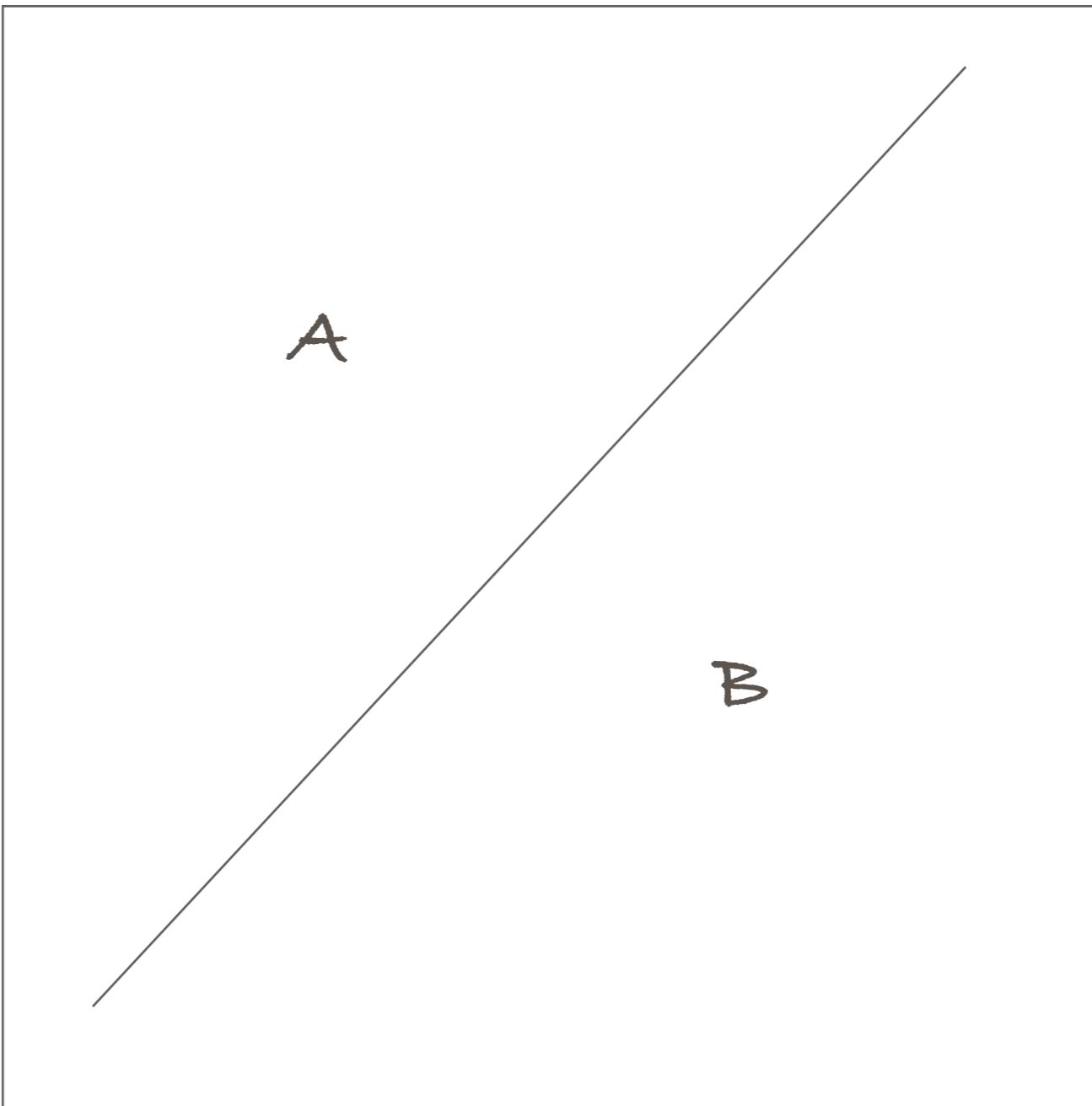
Active Learning

A

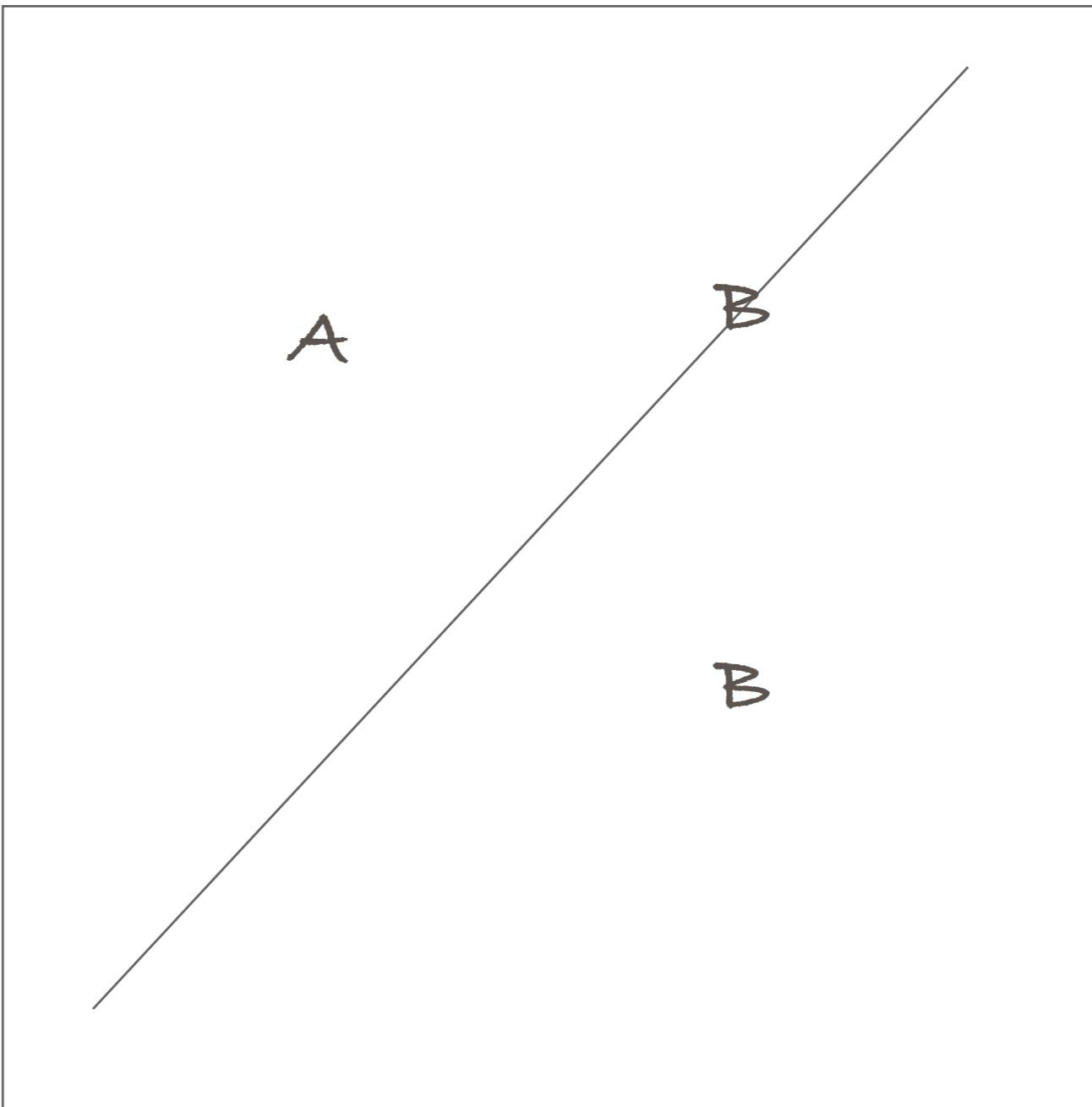
where should i sample next?

B

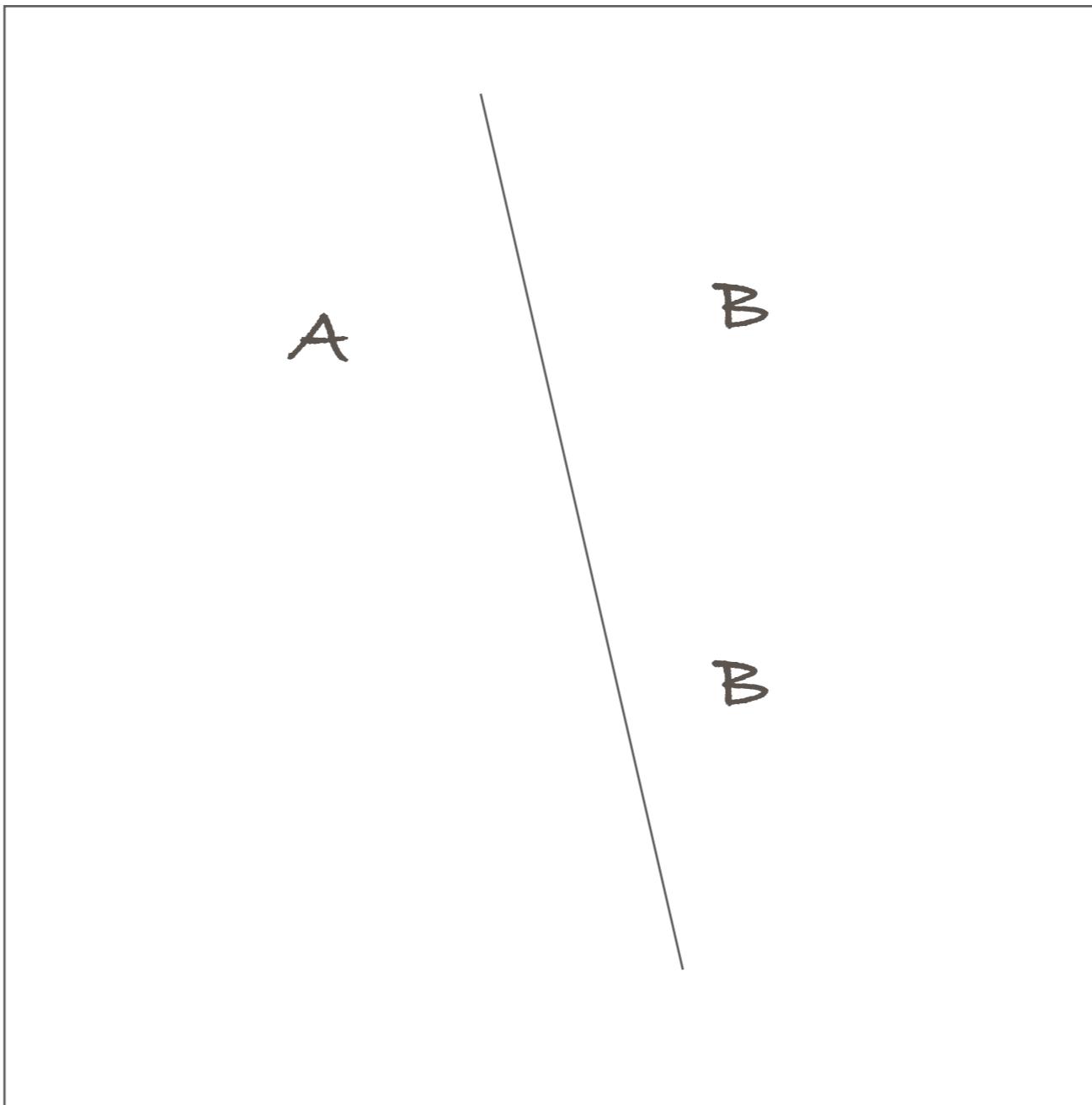
Active Learning



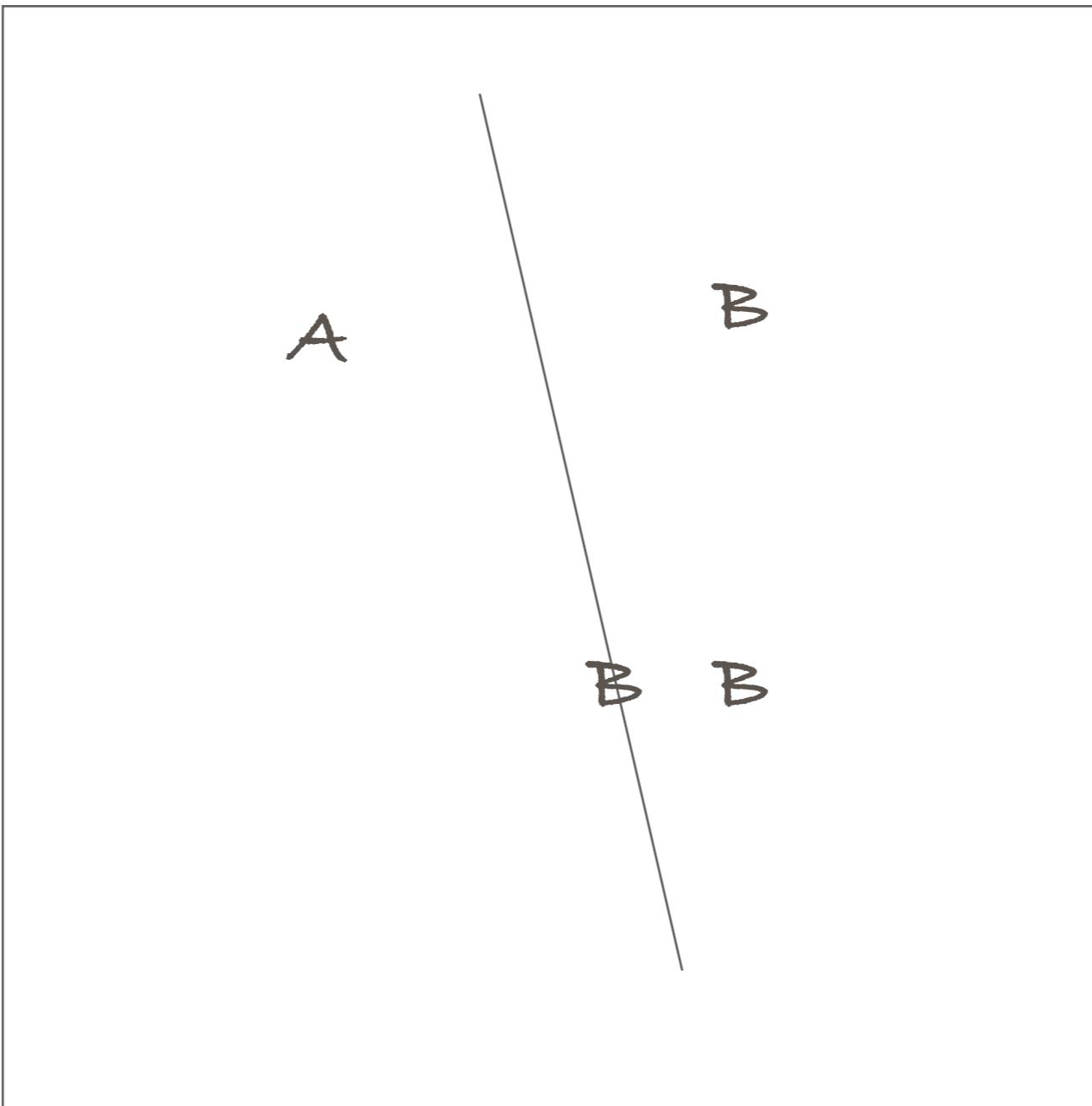
Active Learning



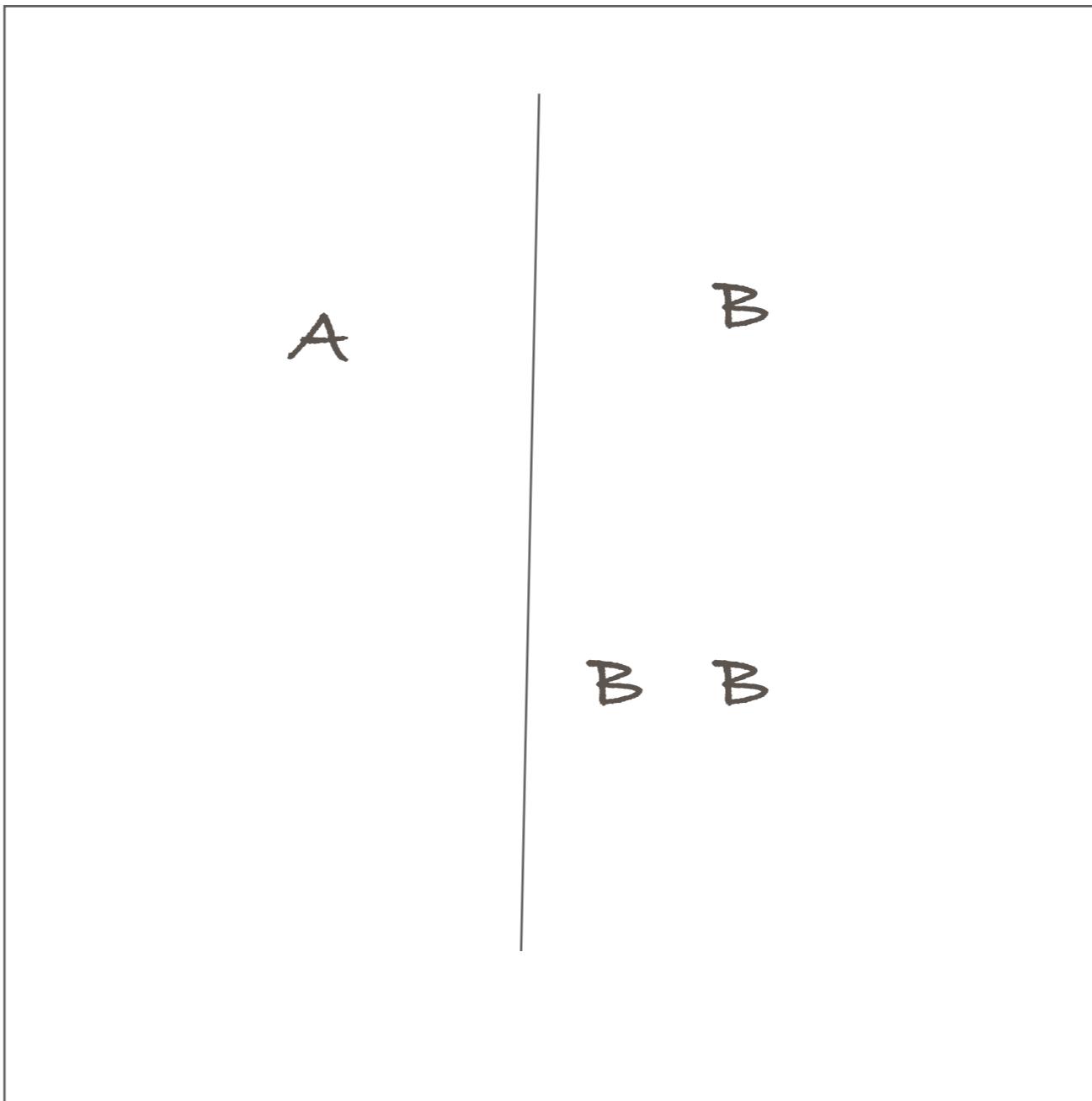
Active Learning



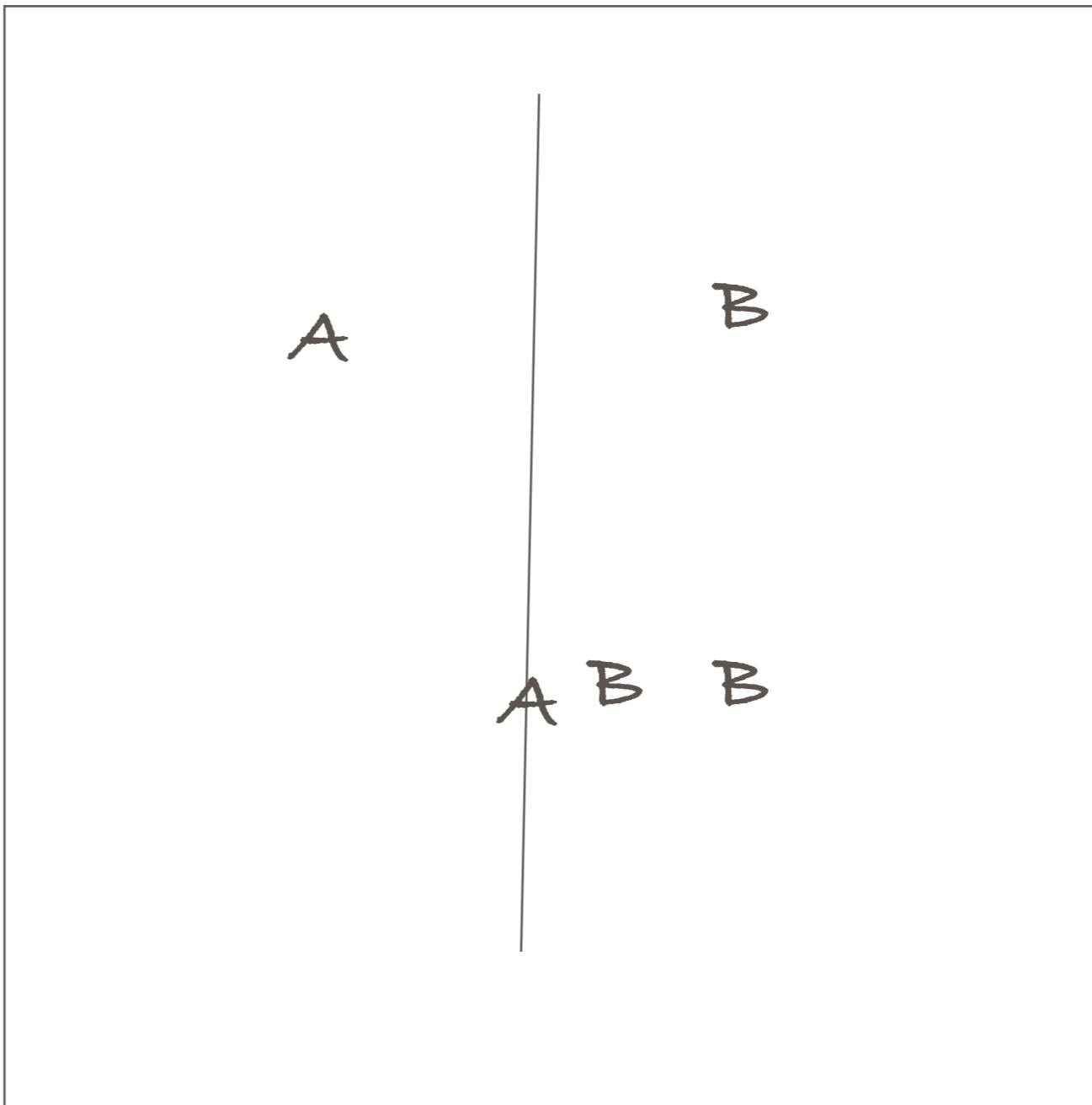
Active Learning



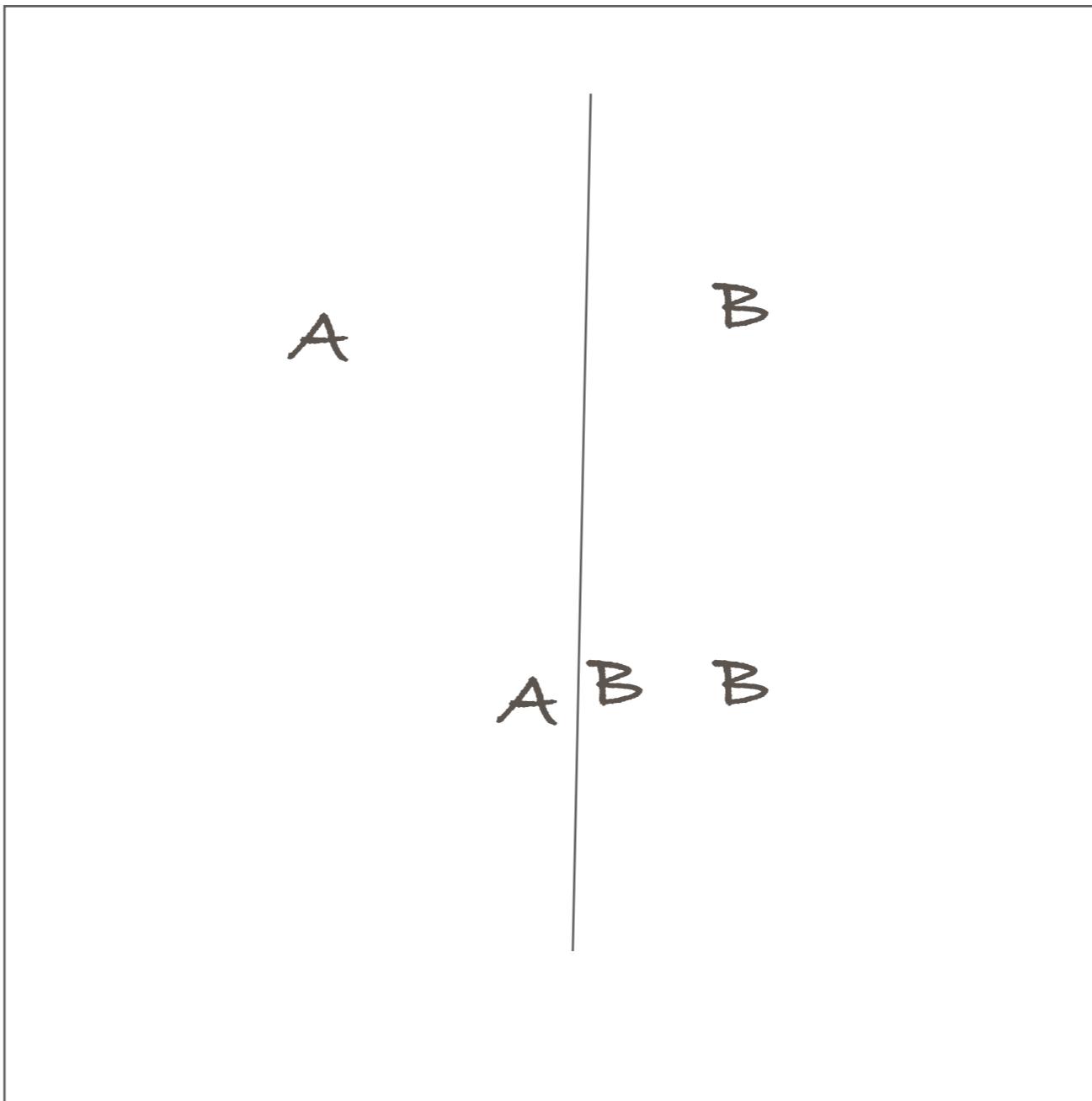
Active Learning



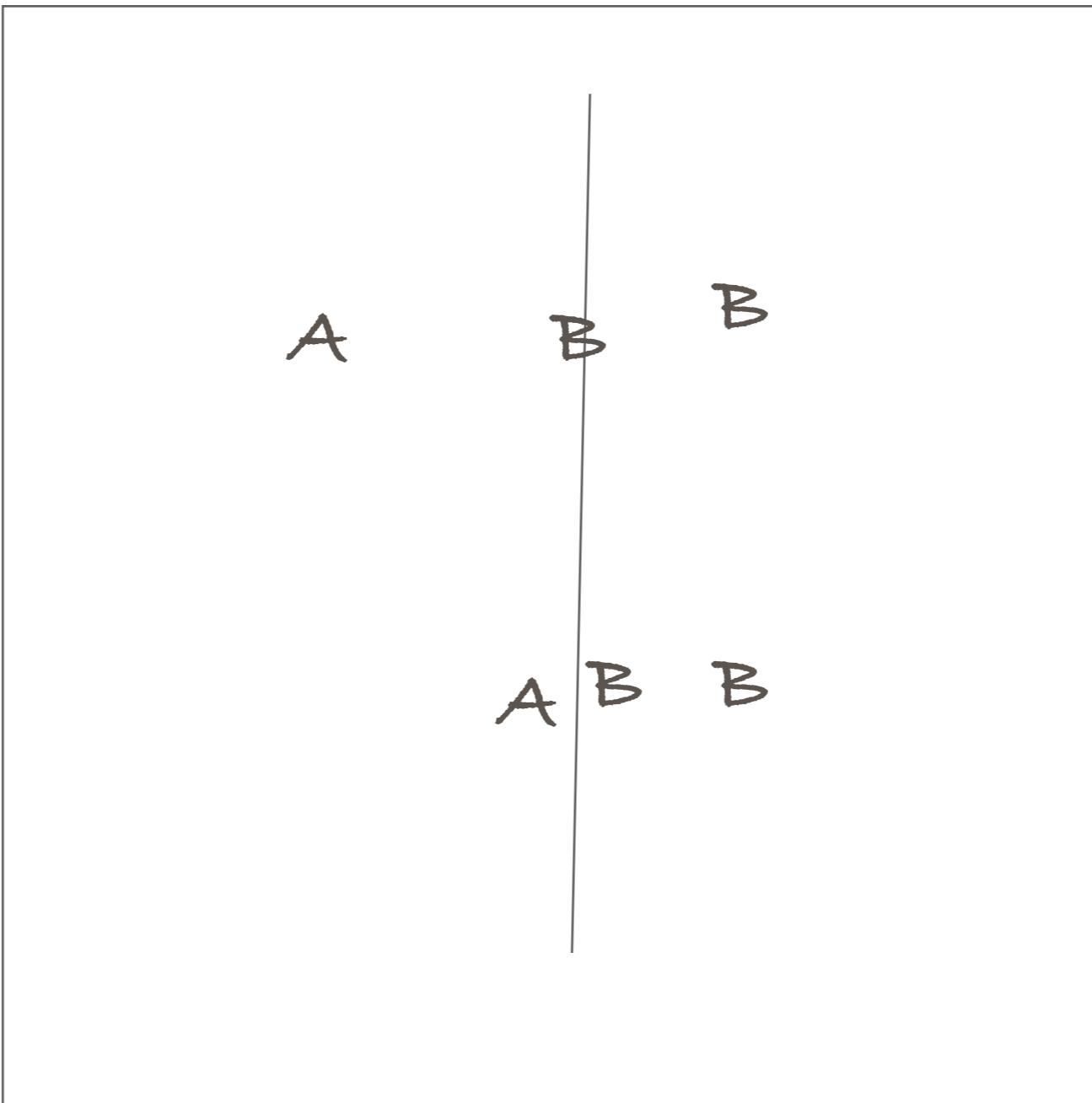
Active Learning



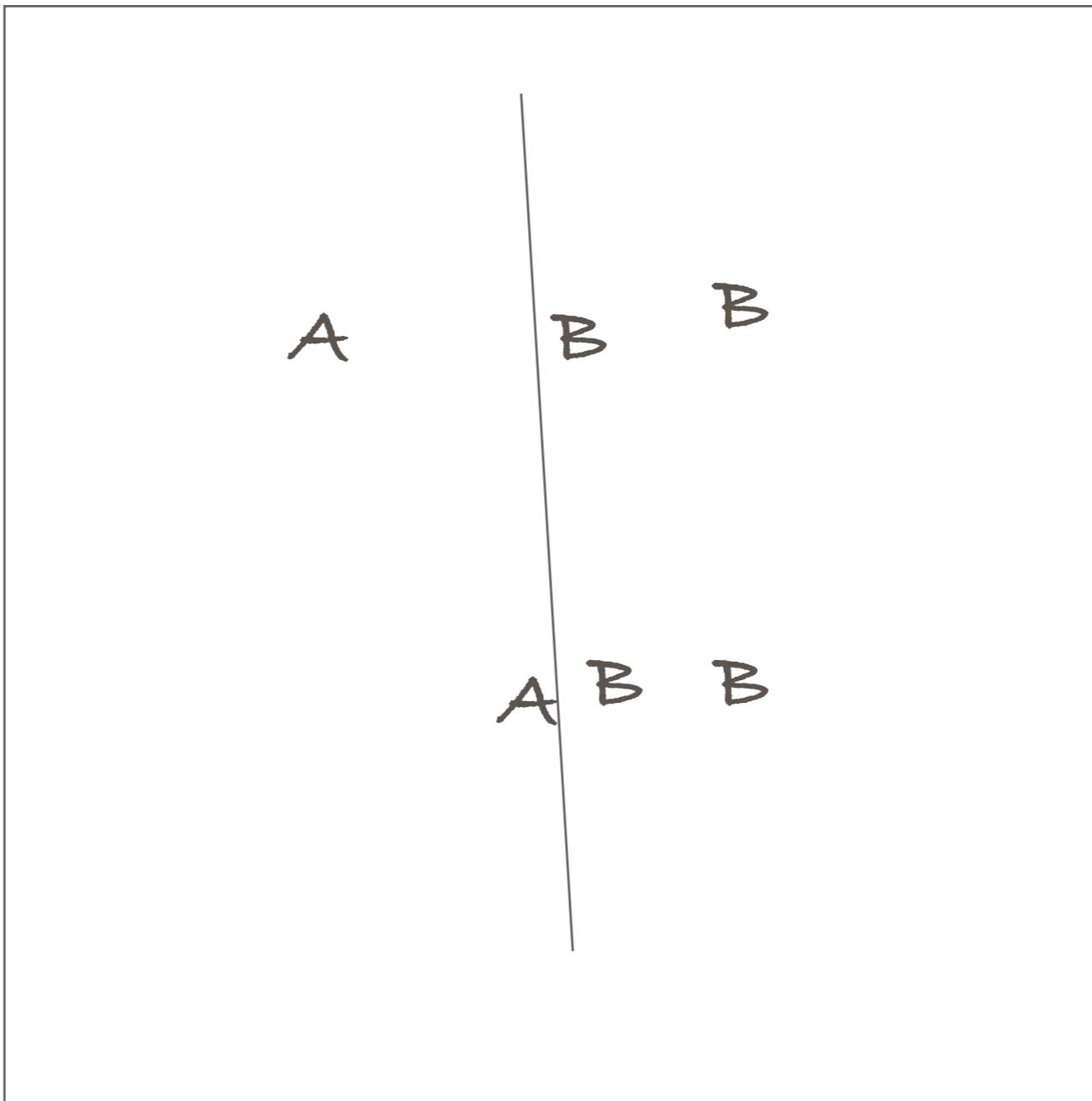
Active Learning



Active Learning

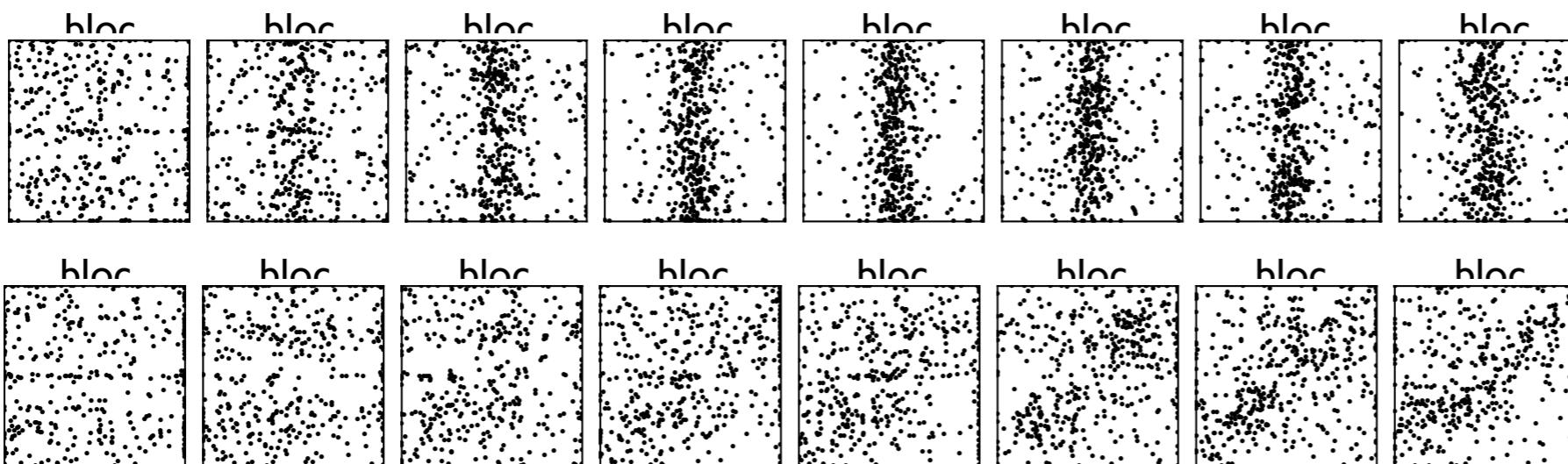
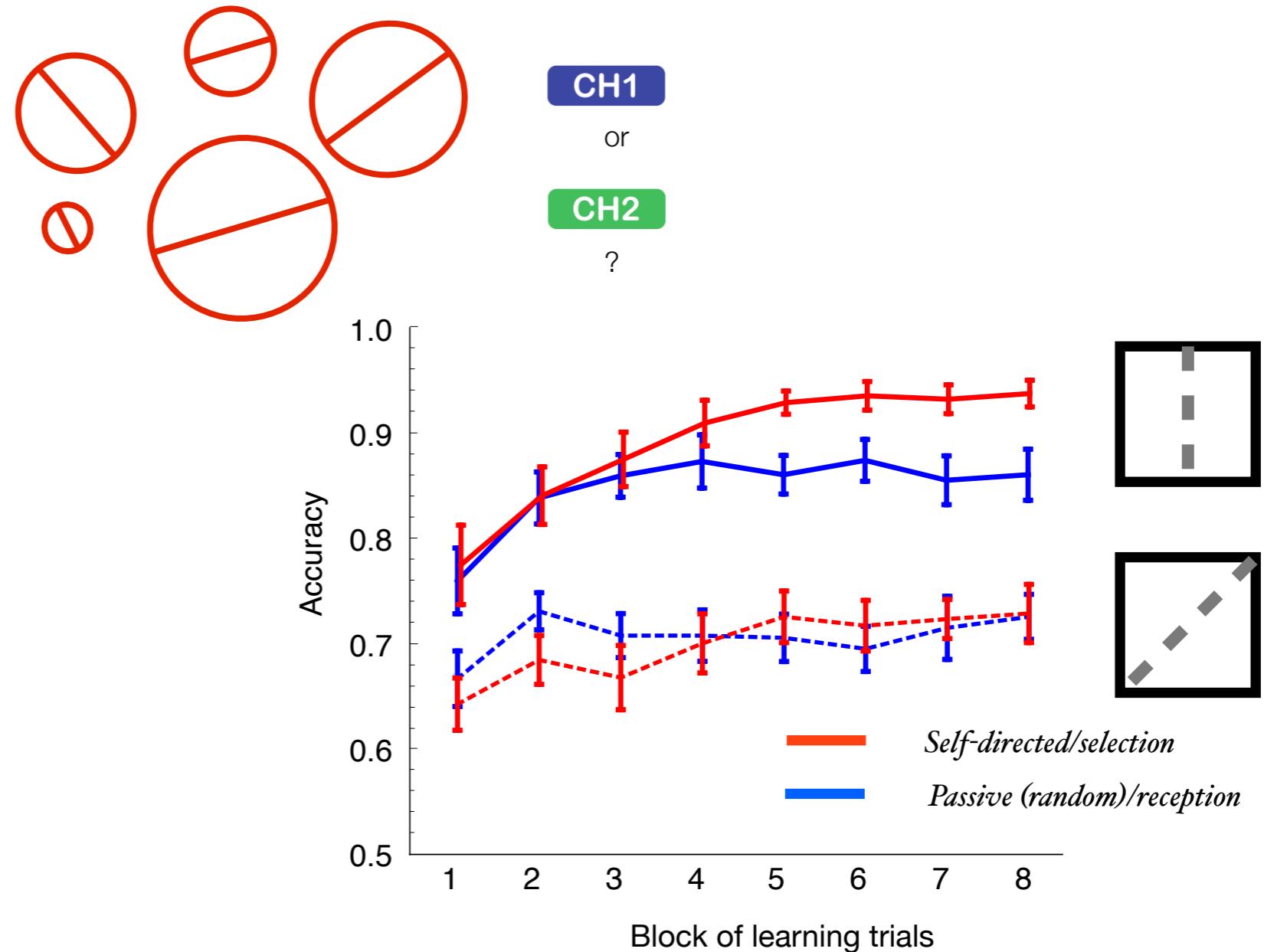
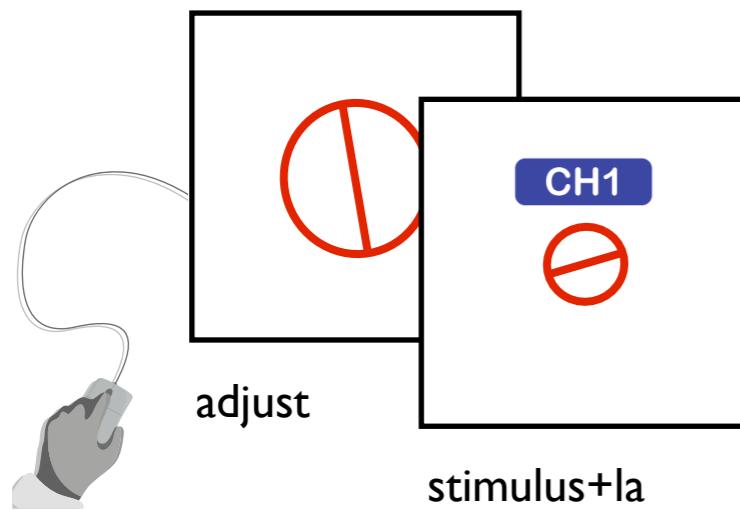


Active Learning



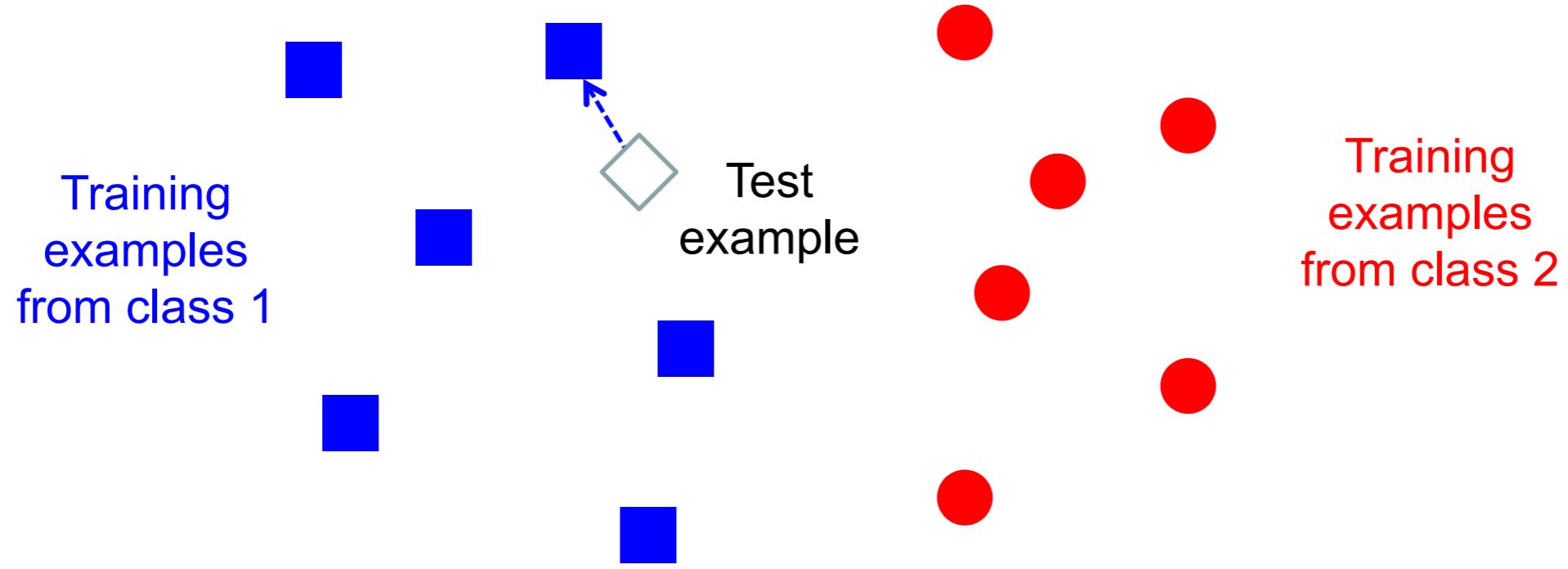
Active Learning by Humans

People choose examples near category boundaries and this can improve rate of learning on a held-out set.



What is the decision architecture of categorization decisions?

e.g., nearest neighbor

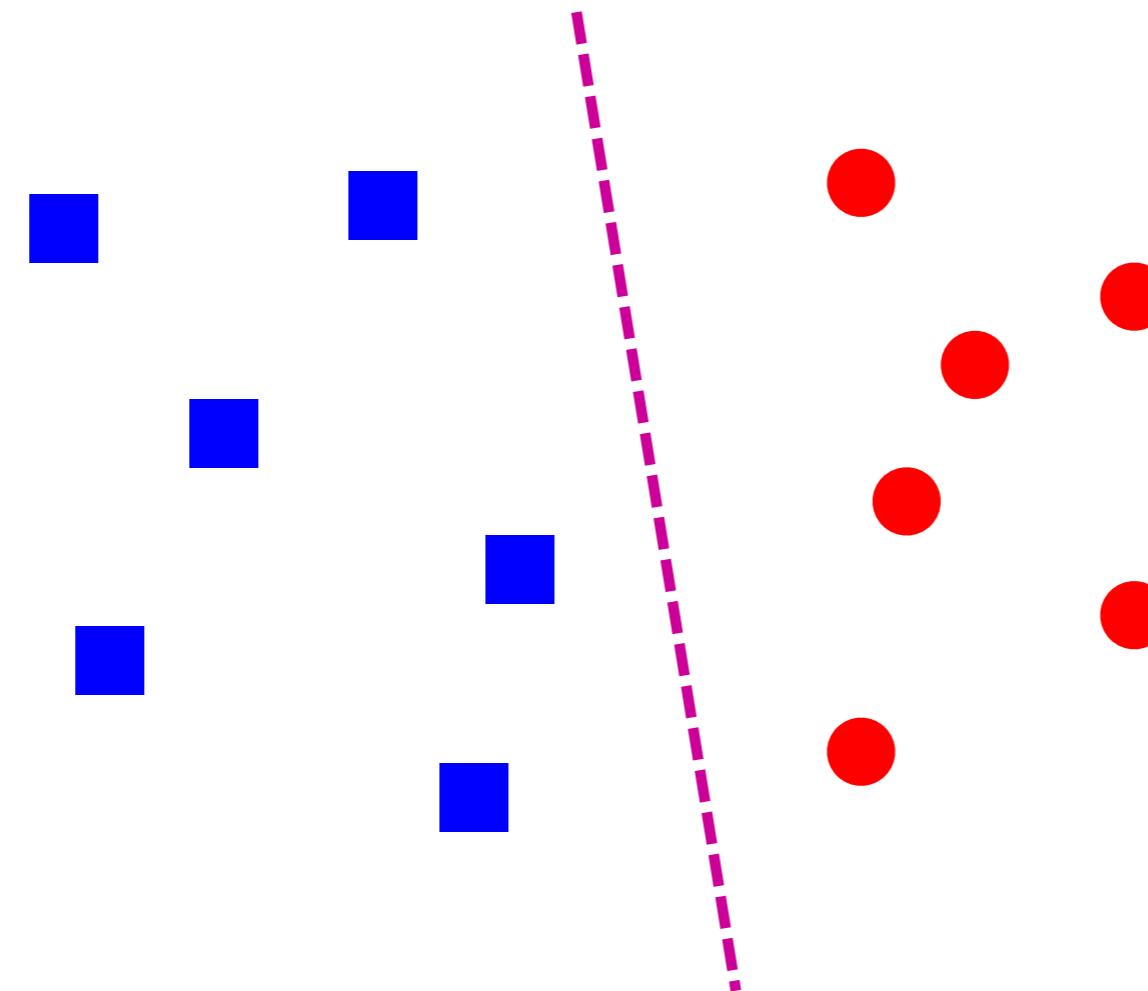


$f(\mathbf{x}) = \text{label of the training example nearest to } \mathbf{x}$

- All we need is a distance or similarity function for our input features
- No training required!
- Incidentally, similarity is a huge topic in human cognition (see Medin, Goldstone, Gentner, Tversky, etc...)!

What is the decision architecture of categorization decisions?

e.g., linear decision boundary



- Find a *linear function* to separate the classes:

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

In the machine learning toolkit many approaches to choose from, many are inter-related

- Support Vector Machines (SVM)
- Neural networks
- Naïve Bayes
- Bayesian network
- Non-parametric Bayesian models
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- Restricted Boltzman Machines (RBMs)
- Etc.

**Which is the best one?
Does that question even
make sense?**

Four case studies exploring relationships between machine learning approaches to categorization and influential ideas in psychology

- **Case 1:** Decision trees <=> Symbolic Rules/Definitions
- **Case 2:** Nearest neighbor <=> Exemplar/Prototype models
- **Case 3:** Mixture models <=> Clustering algorithms
- **Case 4:** Neural networks <=> Connectionist models of category learning

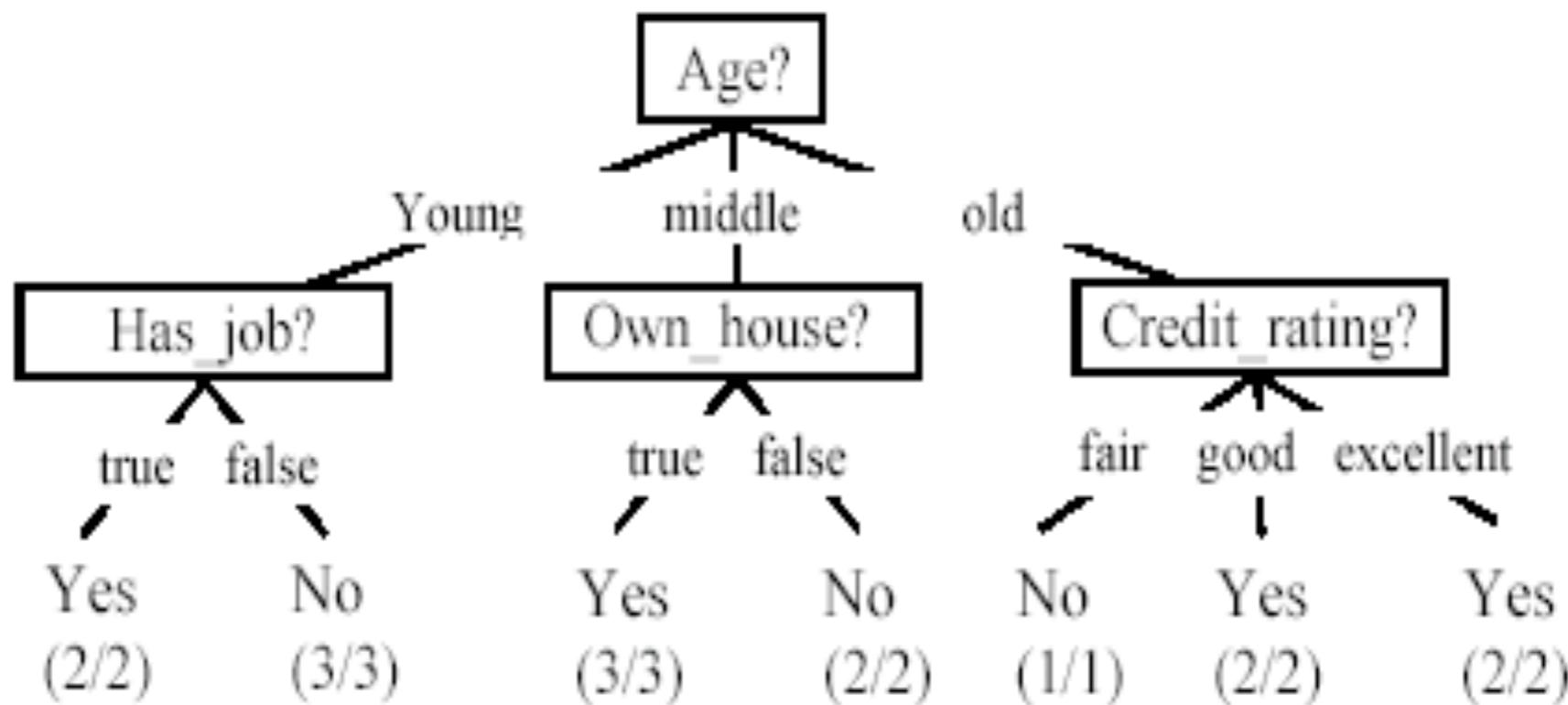
Case 1: Decision Tree Induction

- Decision tree learning is one of the most widely used techniques for classification.
- Discriminative** method
- Its classification accuracy is competitive with other methods, and it is very efficient.
- Assume for now discrete features (e.g., those in a database table)
- The classification model is a tree, called **decision tree**.

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Decision Tree Induction

Decision nodes and leaf nodes (classes)



Decision Tree Induction

Decision nodes and leaf nodes (classes)

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

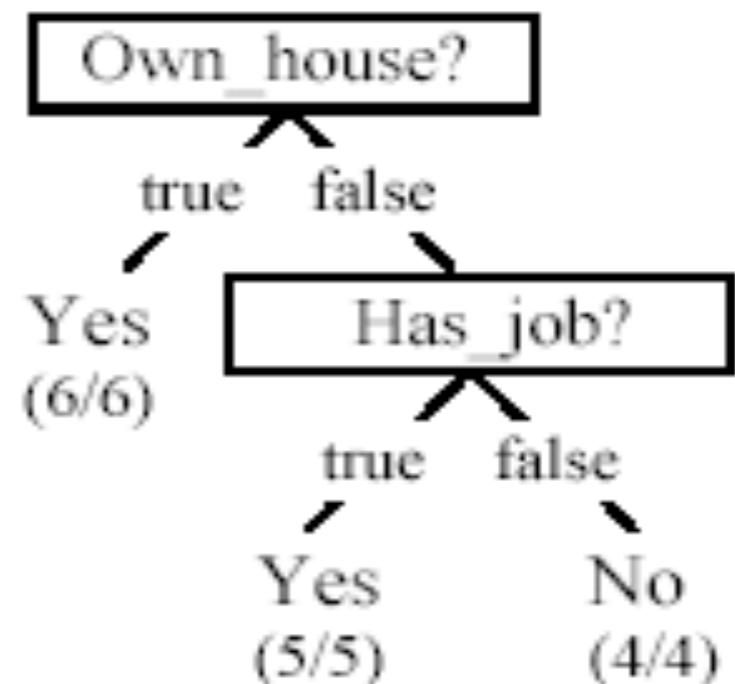
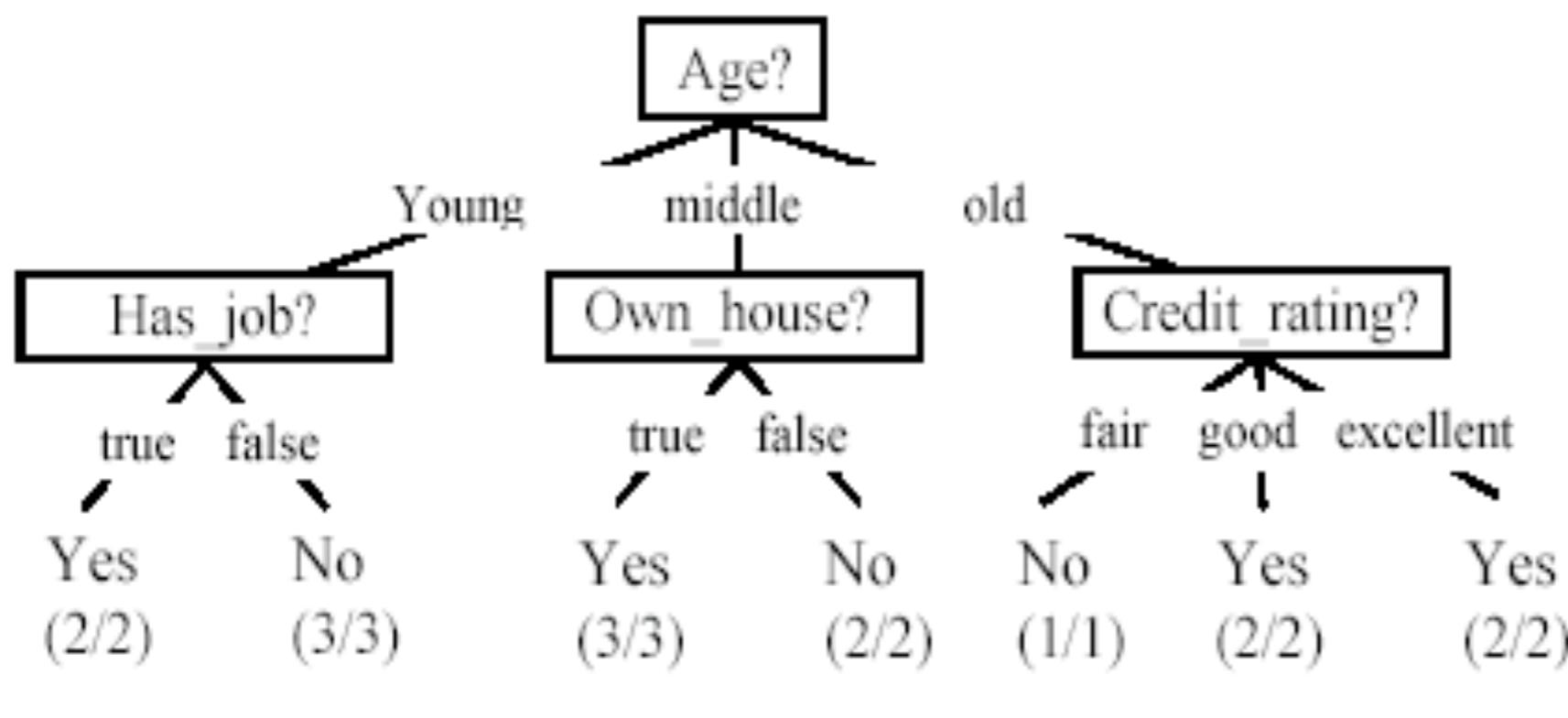

```
graph TD; A[Age?] --> B[Young]; A --> C[middle]; A --> D[old]; B --> E[Has_job?]; C --> F[Own_house?]; D --> G[Credit_rating?]; E --> H[true]; E --> I[false]; H --> J[Yes]; H --> K[No]; J --> L["(2/2)"]; K --> M["(3/3)"]; F --> N[true]; F --> O>false; N --> P[Yes]; N --> Q[No]; P --> R["(3/3)"]; Q --> S["(2/2)"]; G --> T[fair]; G --> U[good]; G --> V[excellent]; U --> W[Yes]; V --> X[Yes]; W --> Y["(1/1)"]; X --> Z["(2/2)"];
```

The decision tree starts with a root node labeled "Age?". It branches into three categories: "Young", "middle", and "old". The "Young" branch leads to a node "Has_job?". If "Has_job?" is "true", the outcome is "Yes" with a count of "(2/2)". If "Has_job?" is "false", the outcome is "No" with a count of "(3/3)". The "middle" branch leads to a node "Own_house?". If "Own_house?" is "true", the outcome is "Yes" with a count of "(3/3)". If "Own_house?" is "false", the outcome is "No" with a count of "(2/2)". The "old" branch leads to a node "Credit_rating?". If "Credit_rating?" is "fair", the outcome is "No" with a count of "(1/1)". If "Credit_rating?" is "good", the outcome is "Yes" with a count of "(2/2)". If "Credit_rating?" is "excellent", the outcome is "Yes" with a count of "(2/2)".

Decision Tree Induction

Decision nodes and leaf nodes (classes)

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?



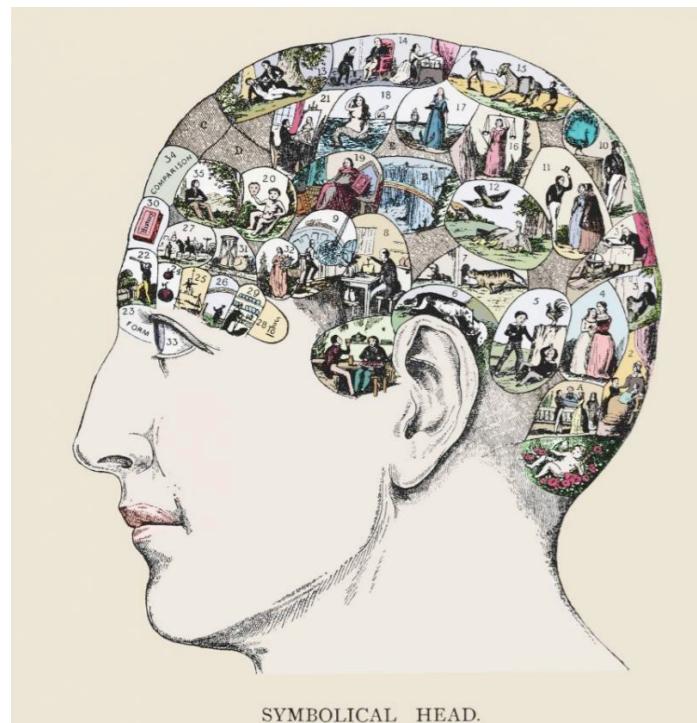
Goal is to find simpler, faster tree (NP-hard)

Decision Tree Induction

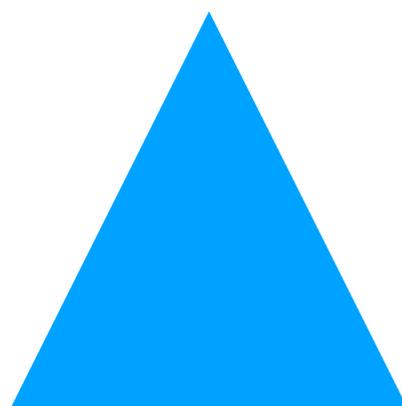
- Basic algorithm (a greedy **divide-and-conquer** algorithm)
 - Assume attributes are categorical (although continuous attributes can be handled too)
 - Tree is constructed in a top-down recursive manner
 - At start, all the training examples are at the root
 - Examples are partitioned recursively based on an impurity function (e.g., **information gain**)
- Conditions for stopping partitioning
 - All examples for a given node belong to the same class
 - There are no remaining attributes for further partitioning – majority class is the leaf
 - There are no examples left

$$\text{entropy}(D) = - \sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j)$$

Is $y = f(x)$ that people use a rule or decision tree?



- According to the **classical view in philosophy of concepts**, concepts are like definitions
- The defining features of are both necessary and sufficient
 - **Necessity:** If something is a category member, it has the defining features
 - **Sufficiency:** If something has the defining features, it is a category member
- **Defining features:** Closed figure, three sides, interior angles sum to 180 degrees
- **Sufficiency:** If something is a closed figure, has three sides and angles sum to 180 degrees it is a triangle
- **Necessity:** If something is a triangle, it is a closed figure, has three sides, and the angles sum to 180 degrees
- Under this view, recognizing something is akin to apply the rule that determines the class membership where the rules are hard and fast/brittle.



Is $y = f(x)$ that people use a rule?

- According to the classical view, category learning usually involves hypothesis testing or rule discovery:
 - A search for the defining features

Name	Concept	Pack I	Pack II	Pack III	Pack IV	Pack V	Pack VI
oo	✓	生 沛 涌 泡 熱 添					
yer	五	狃 狃 珍 狃 狃 狃	狃 狃 珍 狃 狃 狃	狃 狃 珍 狃 狃 狃	狃 狃 珍 狃 狃 狃	狃 狃 珍 狃 狃 狃	狃 狃 珍 狃 狃 狃
li	力	助 動 動 動 動 動	動 動 動 動 動 動	動 動 動 動 動 動	動 動 動 動 動 動	動 動 動 動 動 動	勢 勢 勢 勢 勢 勢
ta	弓	弦 弧 弣 弣 弣 弣	弣 弣 弣 弣 弣 弣	弣 弣 弣 弣 弣 弣	弣 弣 弣 弣 弣 弣	弣 弣 弣 弣 弣 弣	弣 弣 弣 弣 弣 弣
deg	石	善 砾 角 砾 砾 砾	砾 砾 角 砾 砾 砾	砾 砾 角 砾 砾 砾	砾 砾 角 砾 砾 砾	砾 砾 角 砾 砾 砾	砾 砾 角 砾 砾 砾
ling	穴	空 窟 窟 窟 窟 窟	窟 窟 窟 窟 窟 窟	窟 窟 窟 窟 窟 窟	窟 窟 窟 窟 窟 窟	窟 窟 窟 窟 窟 窟	窟 窟 窟 窟 窟 窟

Hull, 1920 - phd thesis

Studied learning concepts defined by simple features

Rules are one basis for complex forms of generalization

... but, there are empirical problems for the classical view

- Hampton (1979): Asked subjects for necessary and sufficient features of everyday categories (sofas, cars, dogs, chairs, birds, etc...). There was little agreement about what the defining features were.
- McClosky & Glucksberg (1978): Asked subjects to judge category membership of several everyday categories. Borderline cases flip from week to week.
 - Rosch (1973): Asked people to rate “how good” different items are as an example of a category (1-7 scale)

A shelf	76%
A rug	52%
A lampshade	63%
Bookends	57%
Candlestick	28%

Robin	1.4
Eagle	1.8
Wren	2.4
Chicken	2.8
Ostrich	3.2

Typical features appear in many category members. # of typical features determines the typicality of a category member.

Properties	Examples					Feature Score (a.k.a. "Weight")
	Robin	Cardinal	Eagle	Penguin	Bat	
Has wings	Yes	Yes	Yes	Yes	Yes	5
Flies	Yes	Yes	Yes	No	Yes	4
Has feathers	Yes	Yes	Yes	Yes	No	4
Sings	Yes	Yes	No	No	No	2
Builds nests in trees	Yes	Yes	Yes	No	No	3
Eats worms/insects	Yes	Yes	No	No	Yes	3
Family Resemblance Score	$5+4+4+2+3+3=$ 21	$5+4+4+2+3+3=$ 21	$5+4+4+3=$ 16	$5+4=$ 9	$5+4+3=$ 12	

Rule induction models in cognitive science

- Nosofsky, Palmeri, McKinley (1994): RULEX “Rule-plus-exception model of classification learning” http://www.cogs.indiana.edu/nosofsky/pubs/1994_rmn-tjp-scm_pr_rule.pdf
- RULEX starts by trying to form perfect single-dimensional rules with some tolerance, and then tries to store exceptions. If evidence demands increase in complexity considered more complex rules (*inductive bias towards simpler rules*)

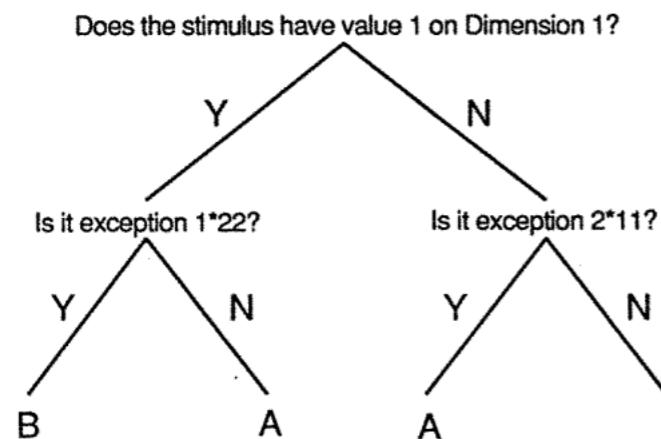


Figure 1. Schematic illustration of one possible decision tree for discriminating the members of Categories A and B in Medin and Schaffer's (1978) experimental paradigm (see Table 1). Y = yes; N = no. The terminal nodes of the decision tree indicate the category to which an item is assigned (A or B). Note that the tests for the exceptions (1*22 and 2*11) can themselves be broken down into a sequence of tests of values on the individual dimensions, thereby extending the decision tree. The simplified structure shown here is provided for conceptual clarity.

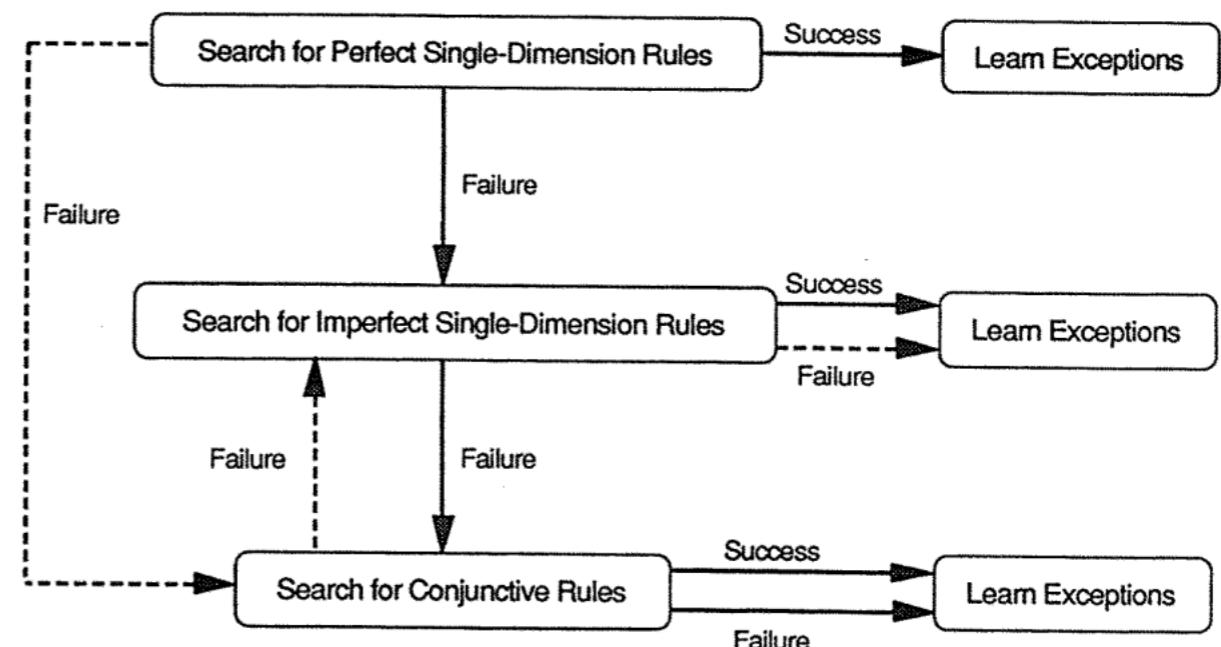


Figure 2. Schematic flow diagram of the sequence of hypothesis-testing stages in rule-plus-exception model of classification learning. The solid lines show the sequence that occurs with high probability, and the dotted lines show the sequence that occurs with lower probability.

Rule induction models in cognitive science

- Nosofsky, Palmeri, McKinley (1994): RULEX “Rule-plus-exception model of classification learning” http://www.cogs.indiana.edu/nosofsky/pubs/1994_rmn-tjp-scm_pr_rule.pdf
- RULEX starts by trying to form perfect single-dimensional rules with some tolerance, and then tries to store exceptions. If evidence demands increase in complexity considered more complex rules (*inductive bias towards simpler rules*)

Table 1
Example Category Structure Tested in Some of Medin and Schaffer's (1978) Experiments

Category A	Category B	Transfer stimuli
A1 1112	B1 1122	T1 1221
A2 1212	B2 2112	T2 1222
A3 1211	B3 2221	T3 1111
A4 1121	B4 2222	T4 2212
A5 2111		T5 2121
		T6 2211
		T7 2122

“training data with labels” “held-out test data”

Table 3
Fit of RULEX Model to Medin and Schaffer's (1978) Experiment 3

Stimulus	Predicted p	Observed p
	Category A	Category B
A1 1112	.950	.970
A2 1212	.974	.970
A3 1211	.997	.920
A4 1121	.867	.810
A5 2111	.734	.720
Category B		
B1 1122	.391	.330
B2 2112	.210	.280
B3 2221	.026	.030
B4 2222	.001	.050
Transfer		
T1 1221	.726	.720
T2 1222	.486	.560
T3 1111	.991	.980
T4 2212	.251	.230
T5 2121	.299	.270
T6 2211	.477	.390
T7 2122	.045	.090

Note. Entries are the predicted and observed probabilities with which each stimulus was classified in Category A during the test phase. RULEX = rule-plus-exception model of classification learning.

Bayesian rule induction methods



h : hypothesis
 d : data

$$P(h | d) = \frac{P(d | h)P(h)}{\sum_{h'} P(d | h')P(h')}$$

Posterior probability Likelihood Prior probability
↓ ↓ ↓
Sum over space of hypotheses



Bayesian rule induction methods

- Goodman, Griffiths, Feldman, Tenenbaum (2007/2008) “A rational analysis of rule-based concept learning”
- The rational rules model assumes a hypothesis space of rules and uses Bayes rules to infer from training examples which set of rules are likely descriptions of the data set.

$$P(F|\mathbf{E}, \ell(\mathbf{E})) \propto P(F)P(\mathbf{E}, \ell(\mathbf{E})|F)$$

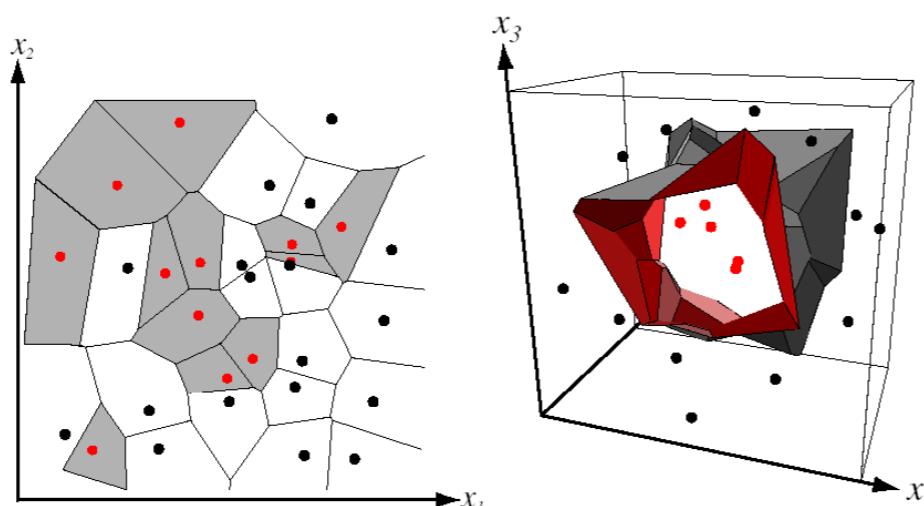
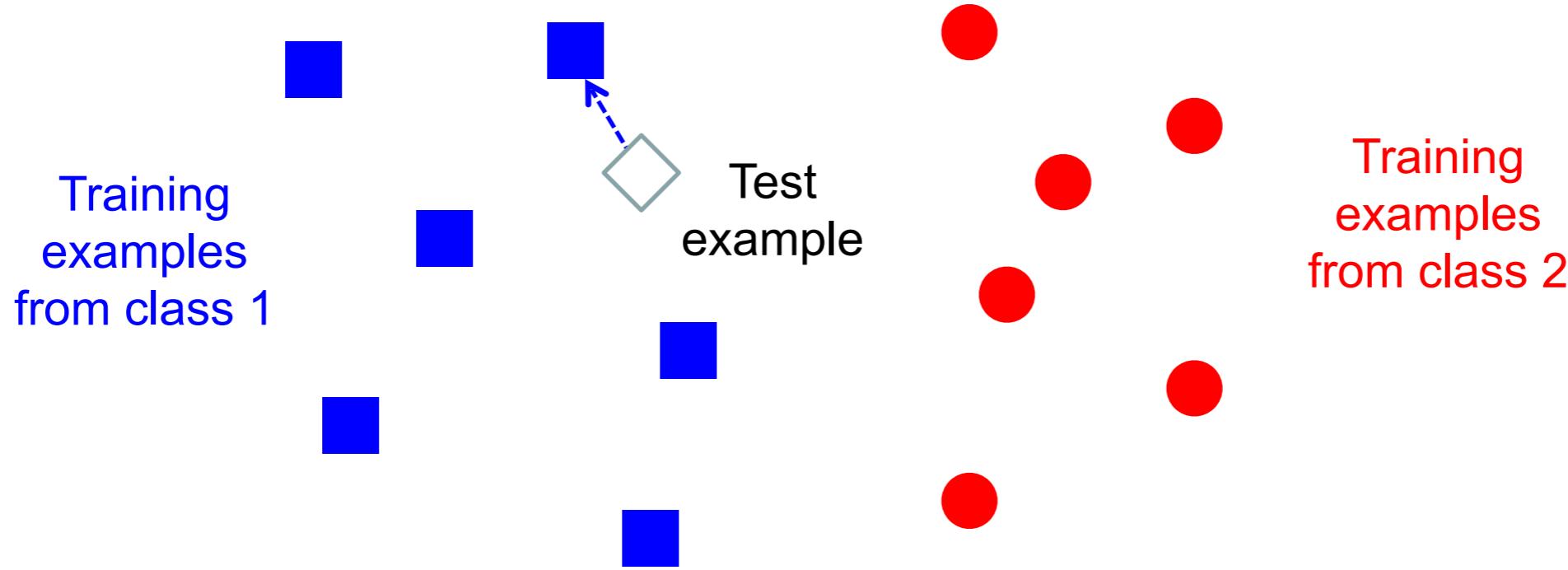
- F is the space of rules,
- E is the set of examples
- $\ell(E)$ is the labels for the example

$$\begin{array}{lcl} S & \rightarrow & (B) \vee S \\ S & \rightarrow & (B) \\ B & \rightarrow & B \wedge P \\ B & \rightarrow & P \\ P & \rightarrow & D_1 \\ & \vdots & \\ P & \rightarrow & D_N \\ D_1 & \rightarrow & f_1(x) = 1 \\ D_1 & \rightarrow & f_1(x) = 0 \\ & \vdots & \\ D_N & \rightarrow & f_N(x) = 1 \\ D_N & \rightarrow & f_N(x) = 0 \end{array}$$

More on this in next lecture!

Figure 1: The DNF Grammar. S is the start symbol, and B, P, D_i the other non-terminals. $f_i(x)$ is the value of the i^{th} feature.

Case 2: Nearest Neighbor Methods (instance based learning)

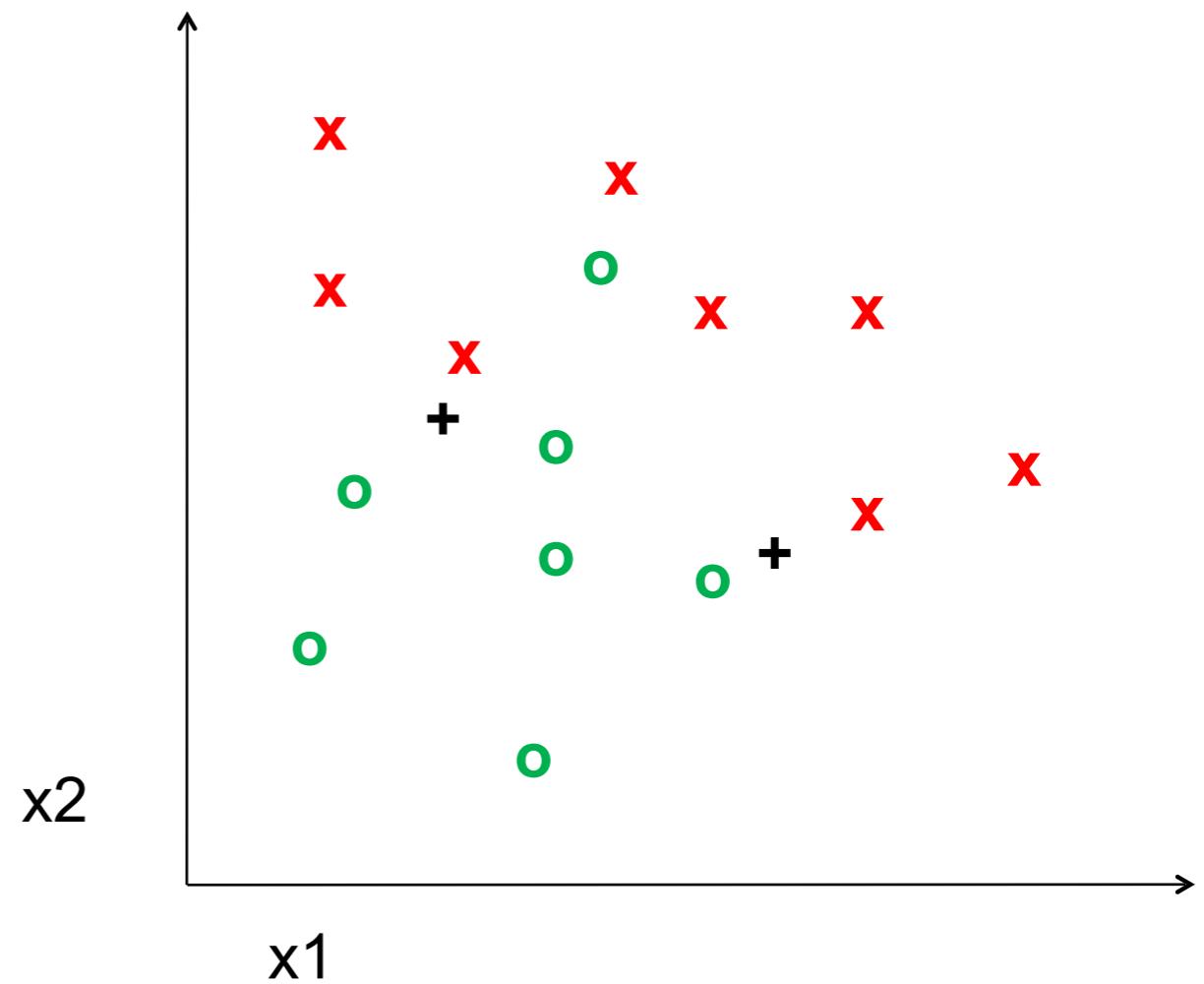


from Duda et al.

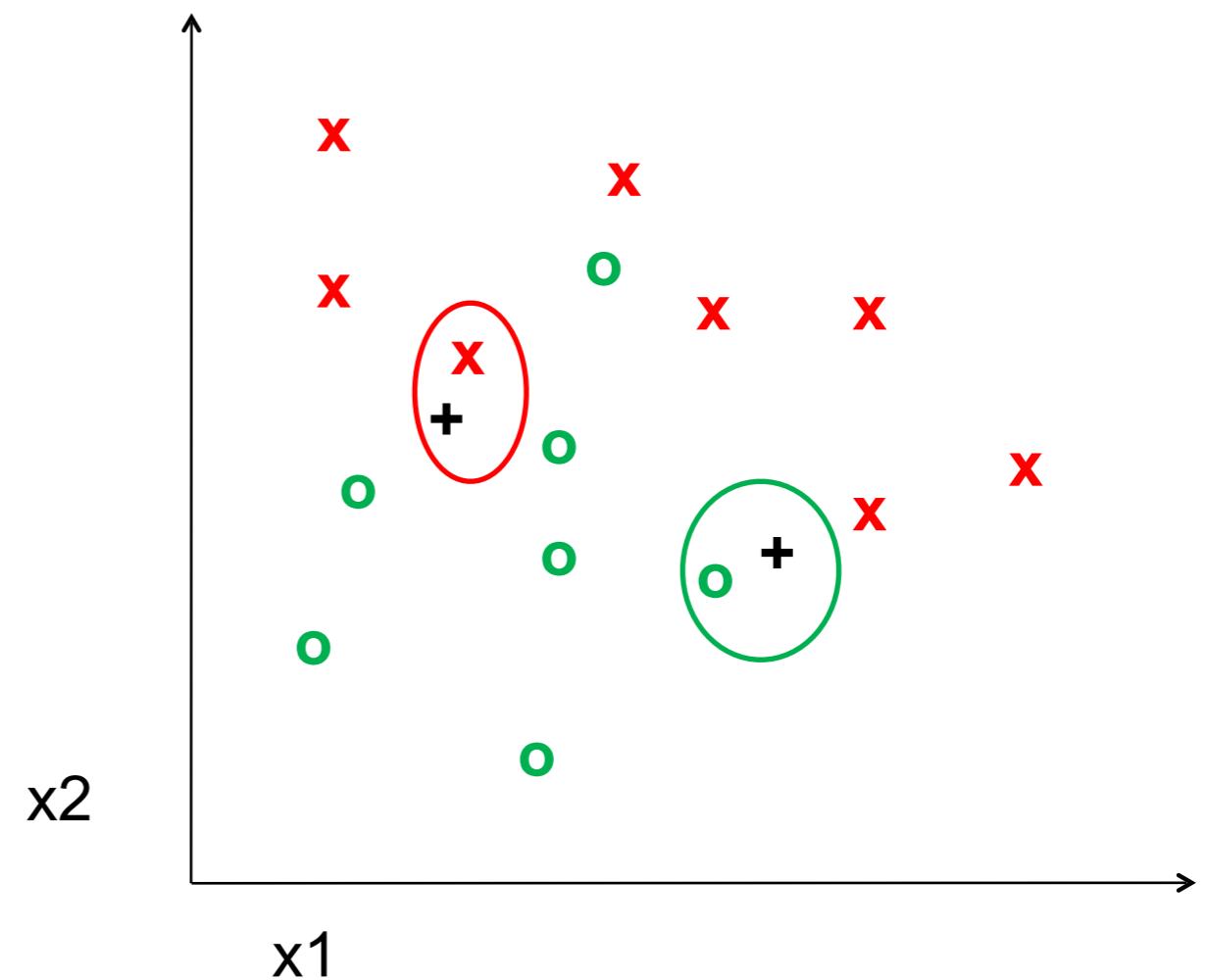
Voronoi partitioning of feature space
for two-category 2D and 3D data

Source: D. Lowe

Case 2: Nearest Neighbor Methods (instance based learning)

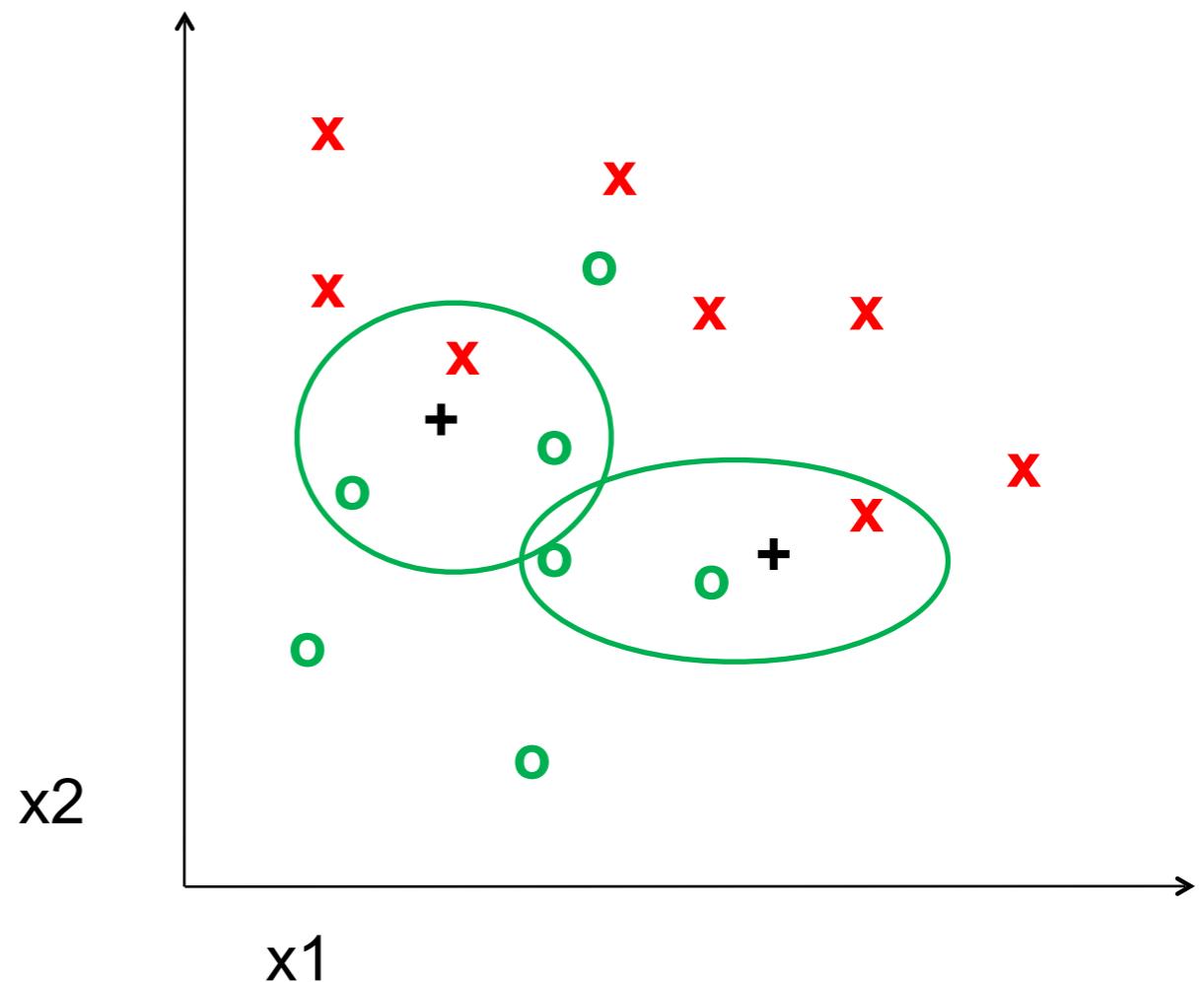


Case 2: Nearest Neighbor Methods (instance based learning) - 1 nearest

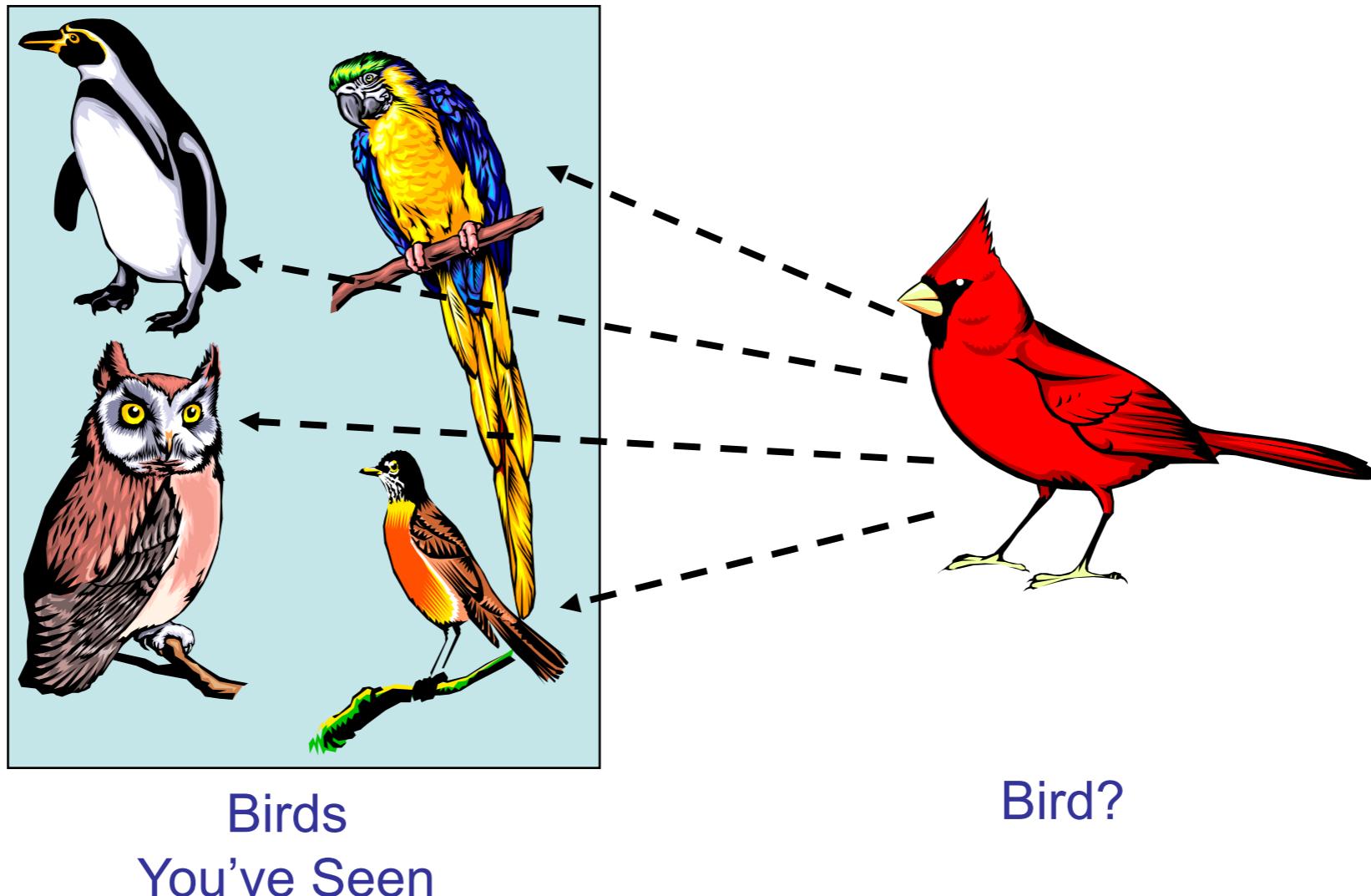


Case 2: Nearest Neighbor Methods (instance based learning) - 3 nearest

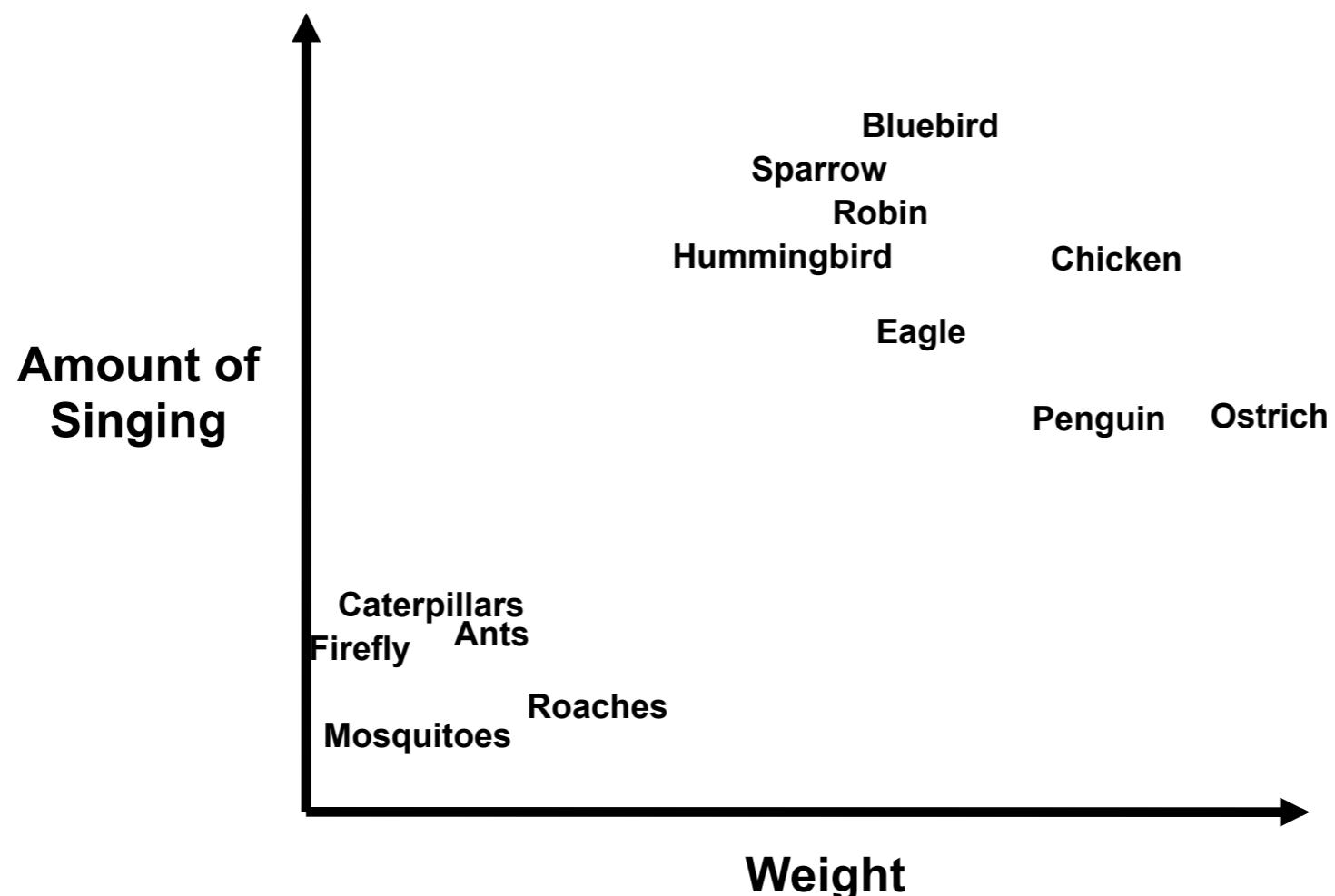
- Take the majority vote

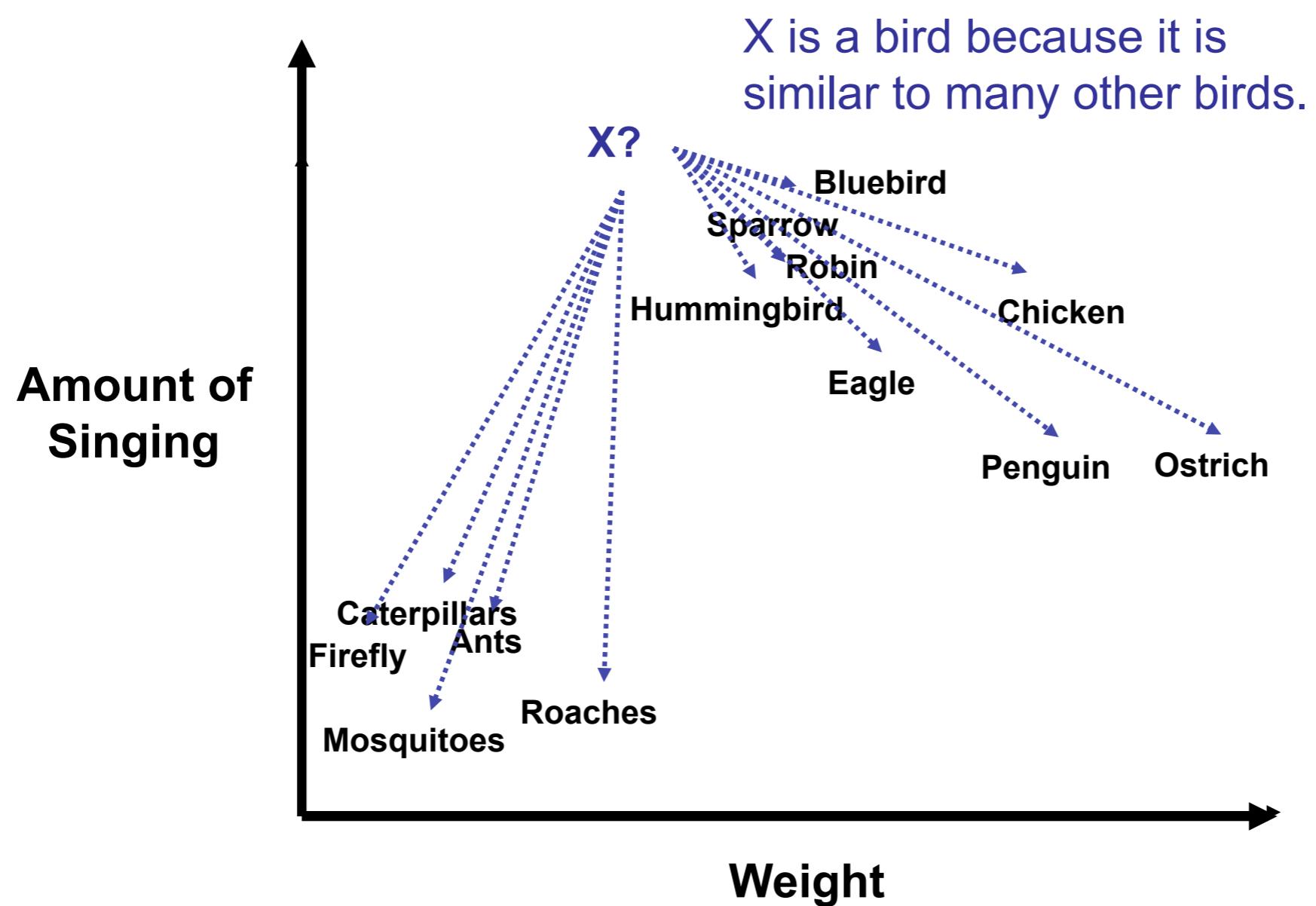


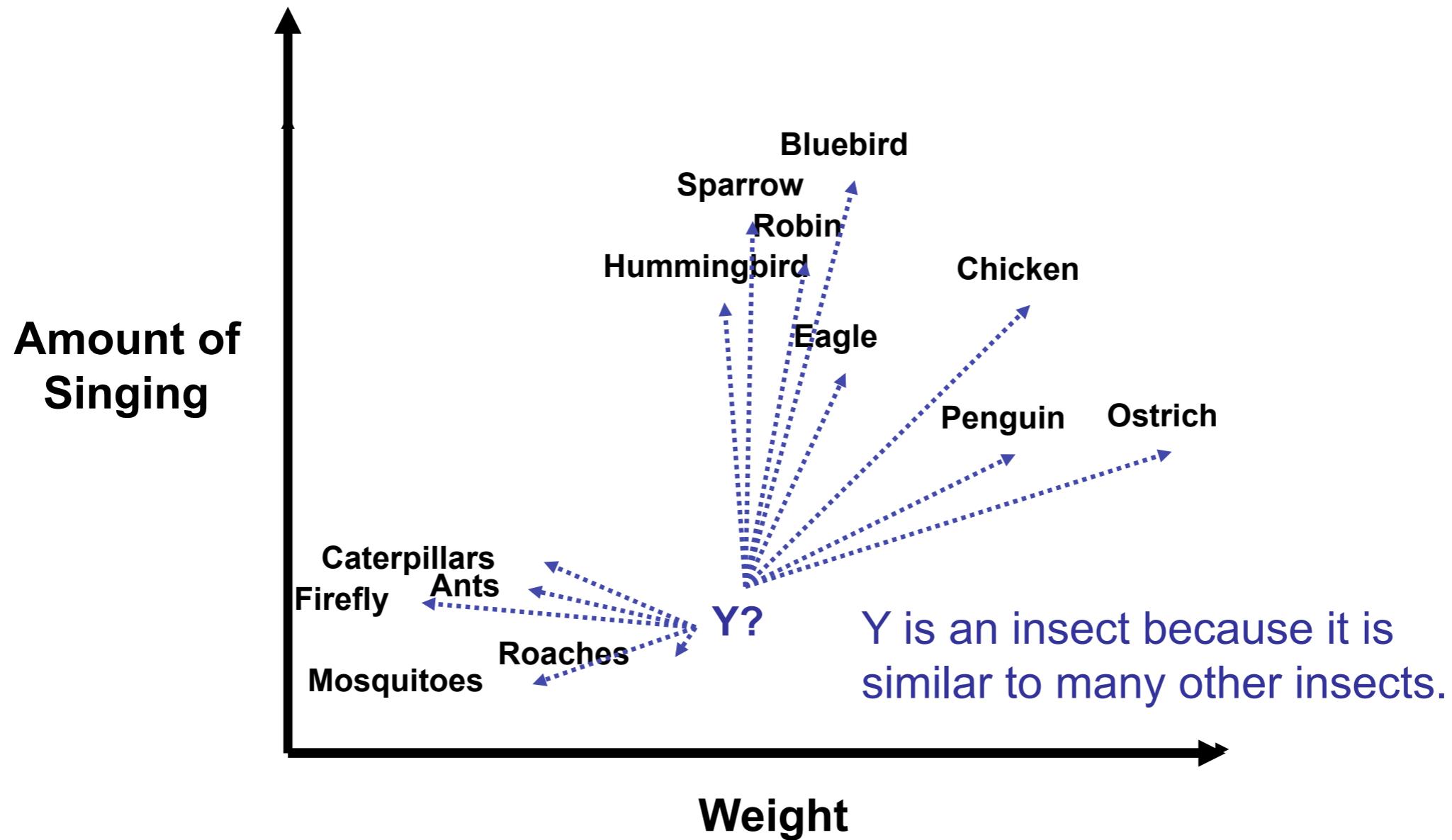
Is $y = f(x)$ that people used based on instances?

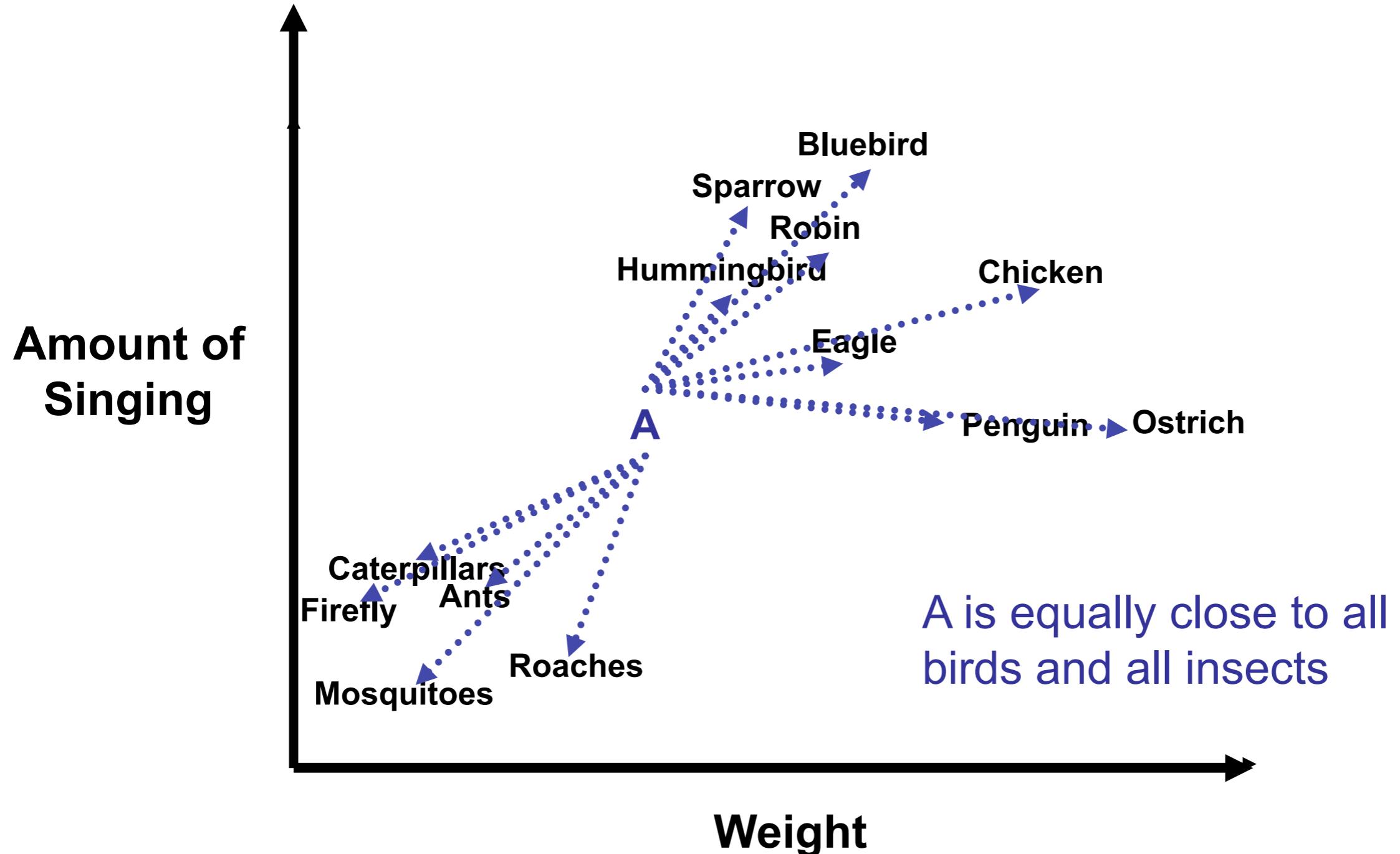


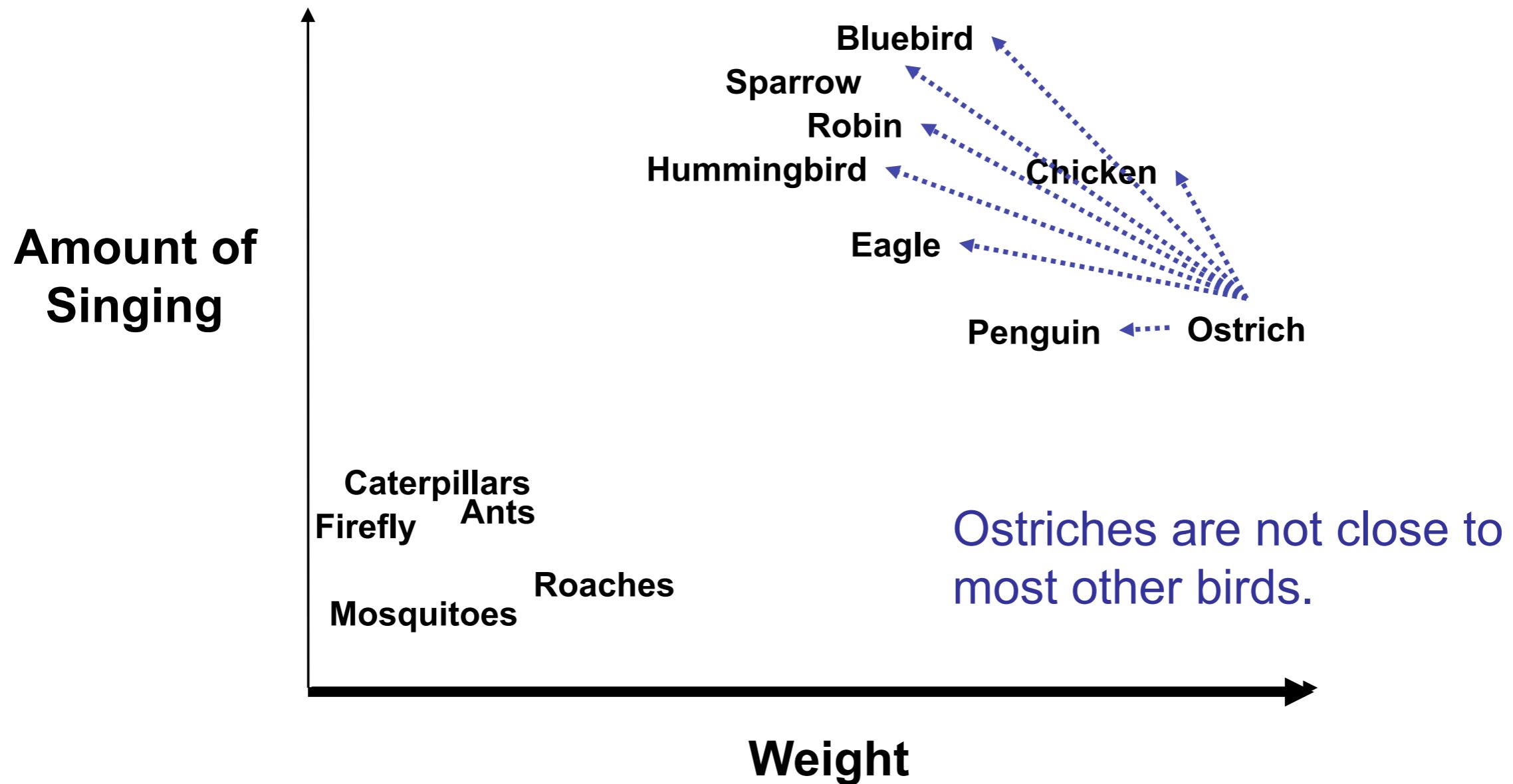
Is $y = f(x)$ that people used based on instances?







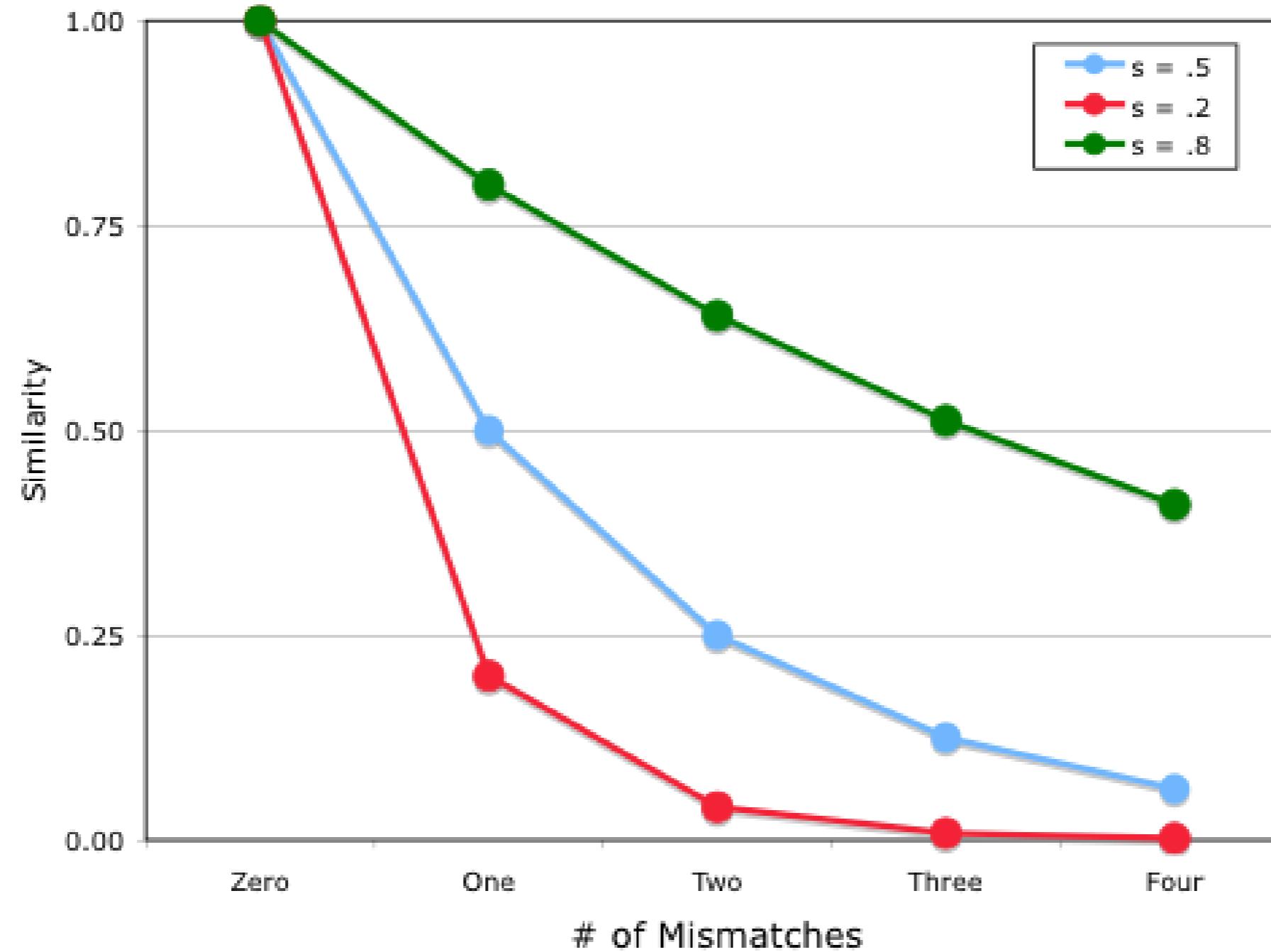




Similarity and Exemplar Models

- How is similarity to the stored examples computed?
- Medin & Schaffer (1978) proposed the **context model** of classification
 - A model of similarity for binary dimensions
 - A simple model of evidence accrual
 - A simple model of decision making
- Each dimension has an associated importance or weight
 - An s parameter (0-1) which controls importance
 - When comparing two items, compute a match score, m , on each dimension
 - $m_i = 1$ if values on dimension i match
 - $m_i = s_i$ if values on that dimension mismatch
 - Overall similarity is the product of the m values

Similarity and Exemplar Models



Evidence accrual

- Similarity of item S_i to a category C_j is the sum of its similarities to the category's exemplars

- $$sim(C_j, S_i) = \sum_k sim(S_k, S_i)$$

Decision making

- The probability of classifying S_i as a C_j is the ratio of its evidence relative to other categories

- $$p(C_j | S_i) = \frac{sim(C_j, S_i)}{\sum_k sim(C_k, S_i)}$$

The Generalized Context Model

Nosofsky (1984; 1986)

- The **generalized context model (GCM)**
 - Application of the context model to continuous dimensions.
 - Unification of Luce's work on choice behavior and Shepard's work on stimulus generalization
- Similarity is a function of the **distance** between two objects in psychological space (Shepard!!).

$$d_{ij} = c \left(\sum_{k=1}^N w_k |x_{ik} - x_{jk}|^r \right)^{1/r} \quad d_{ij} = \left(\sum_{k=1}^K |x_{ik} - x_{jk}|^r \right)^{1/r}$$

The Generalized Context Model

Nosofsky (1984; 1986)

- Actual similarity of two objects is a function of their distance:

Exponential

$$\eta_{ij} = e^{-d_{ij}}$$

Gaussian

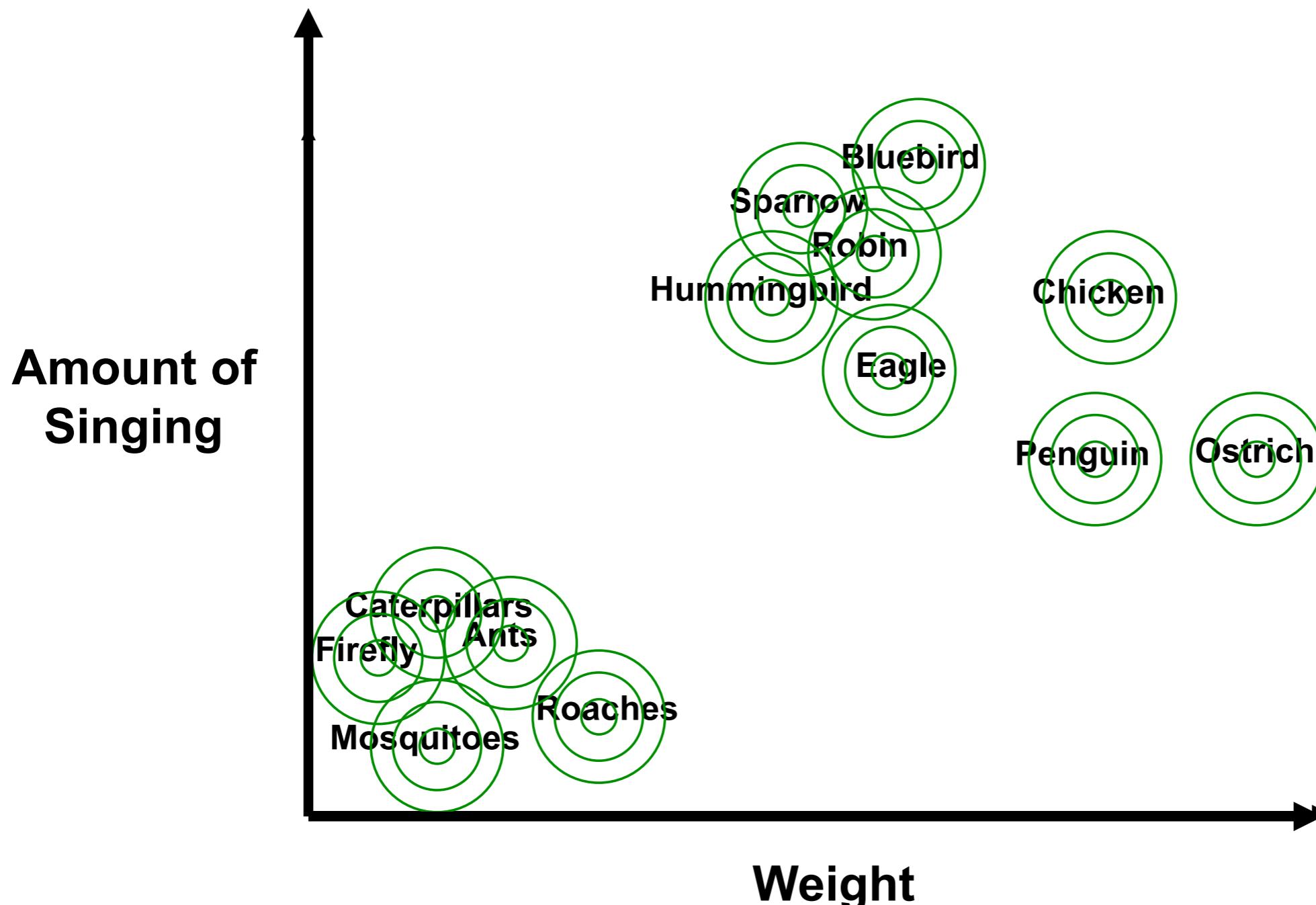
$$\eta_{ij} = e^{-d_{ij}^2}$$

- Response rule

$$p(R_j | S_i) = \frac{b_j \sum_{j \in C_j} n_{ij}}{\sum_{k=1}^m (b_k \sum_{j \in C_k} n_{ik})}$$

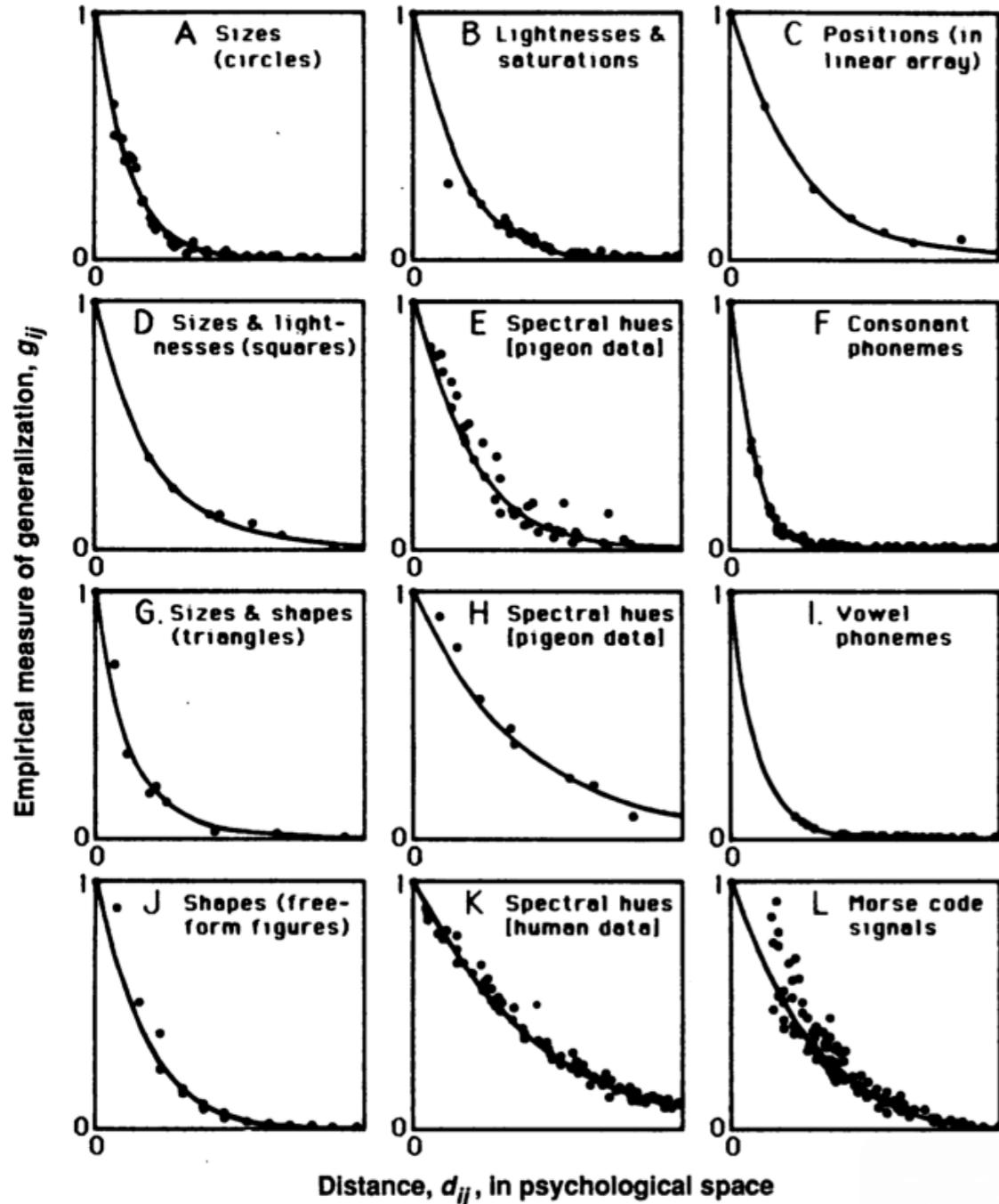
The Generalized Context Model

Nosofsky (1984; 1986)



The Generalized Context Model

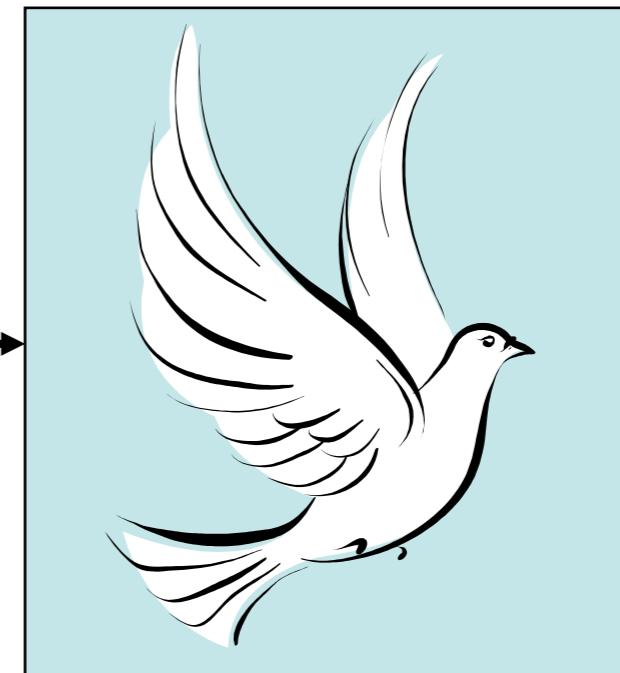
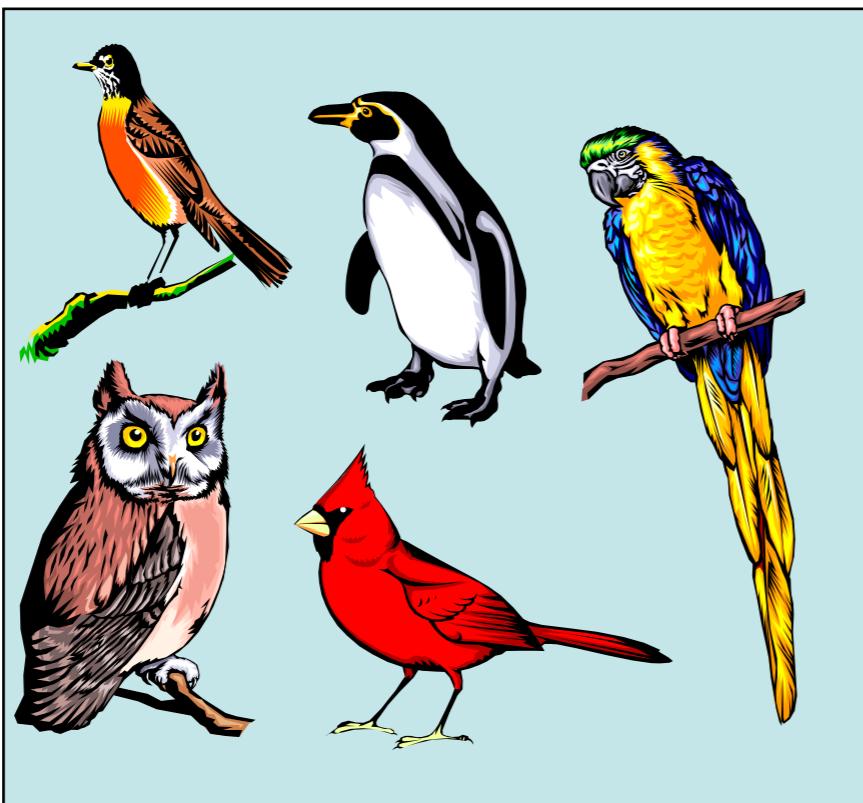
Nosofsky (1984; 1986)



- The **c** parameter in the model matches the exponential generalization gradient in Shepard's work

Prototype Theory

- According to **prototype theory**, the mental representation of a category consists of a prototype or central tendency of the examples
- Learning is about abstracting this schema or prototype across all the examples you have seen so far.

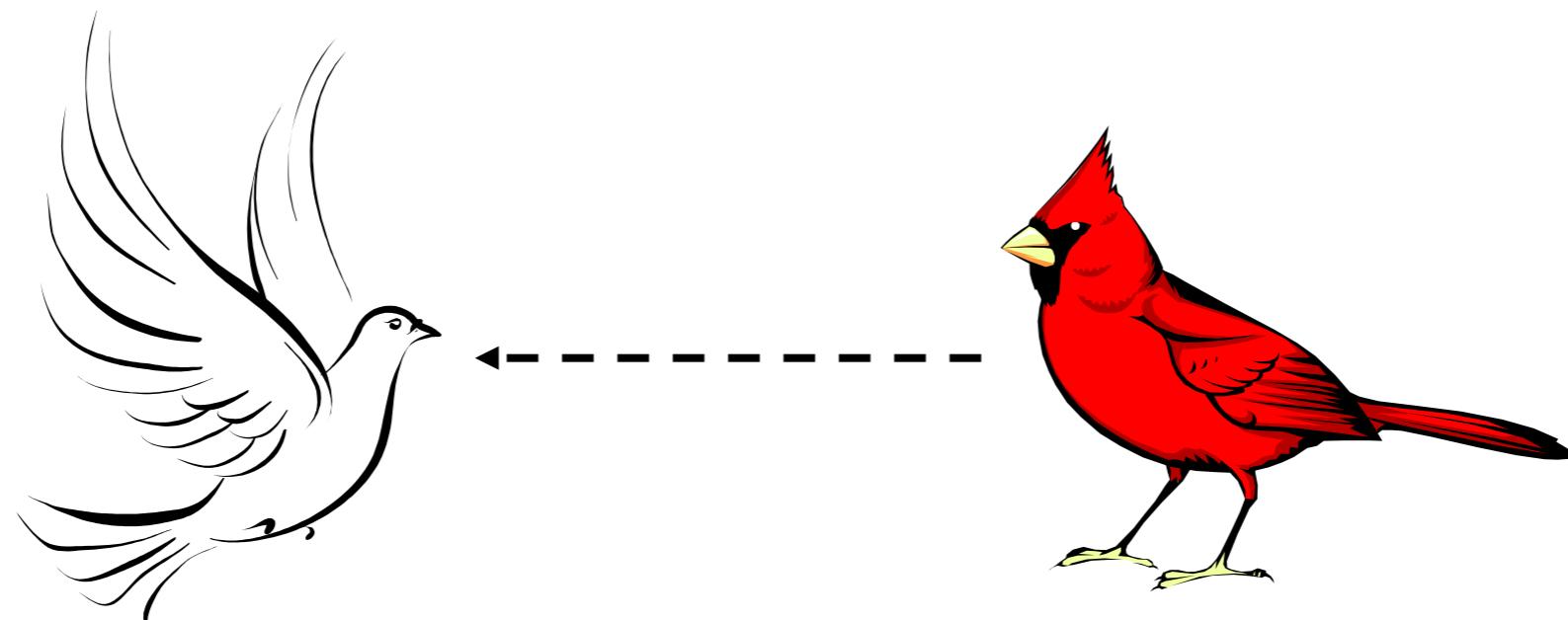


Birds
You've Seen

Prototypical
Bird

Prototype Theory

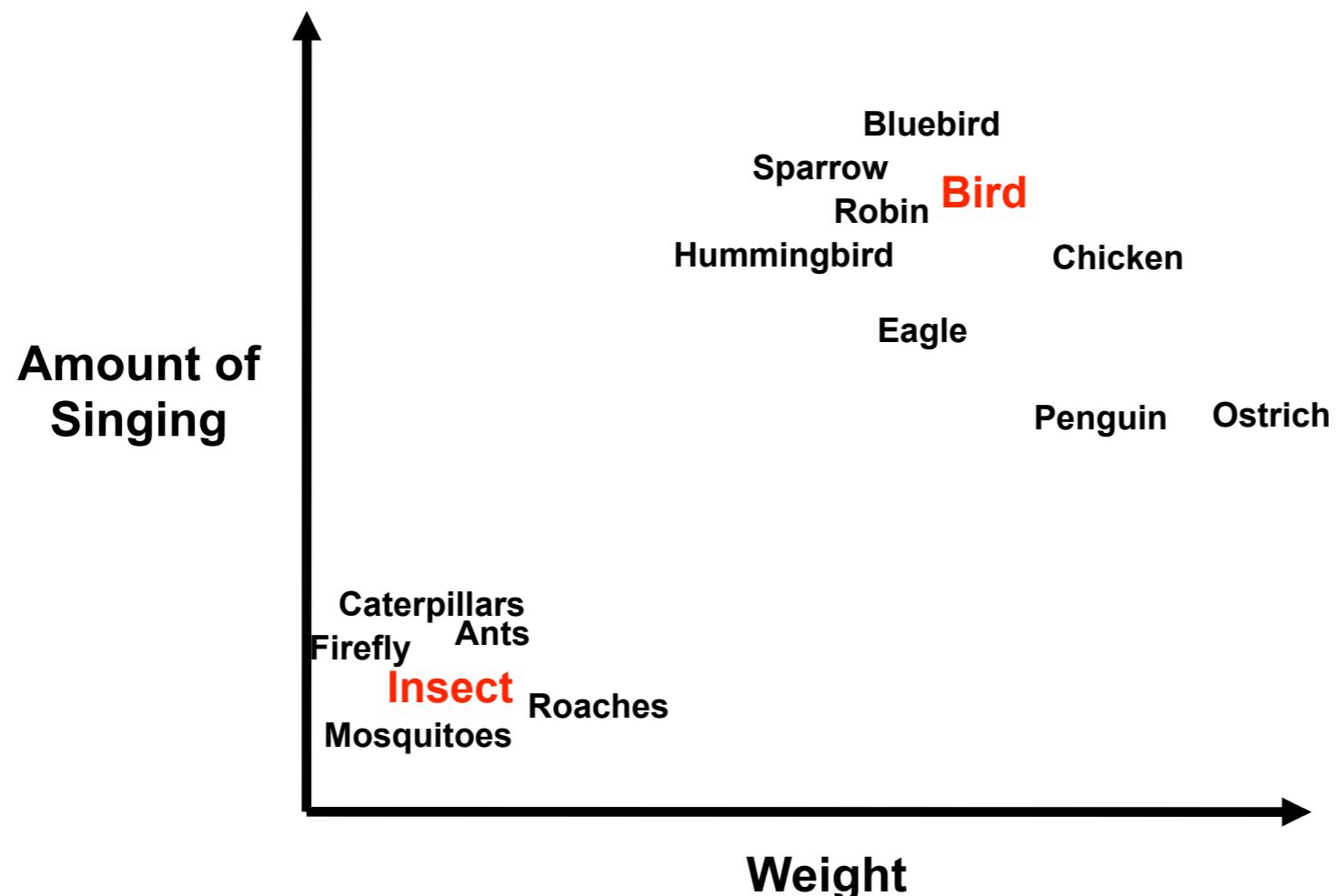
- According to **prototype theory**, the mental representation of a category consists of a prototype or central tendency of the examples
- Learning is about abstracting this schema or prototype across all the examples you have seen so far.



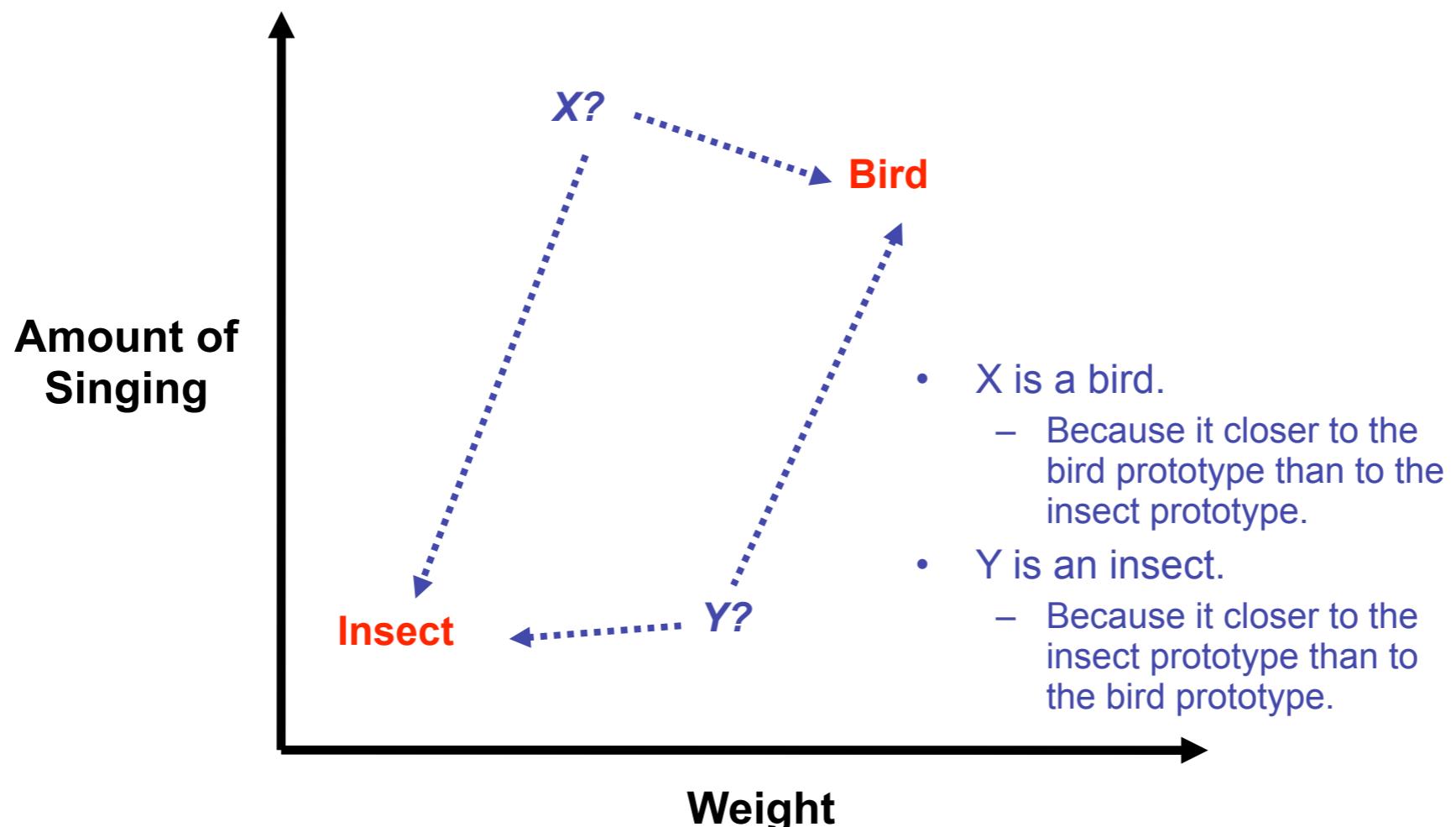
Prototypical
Bird

Bird?

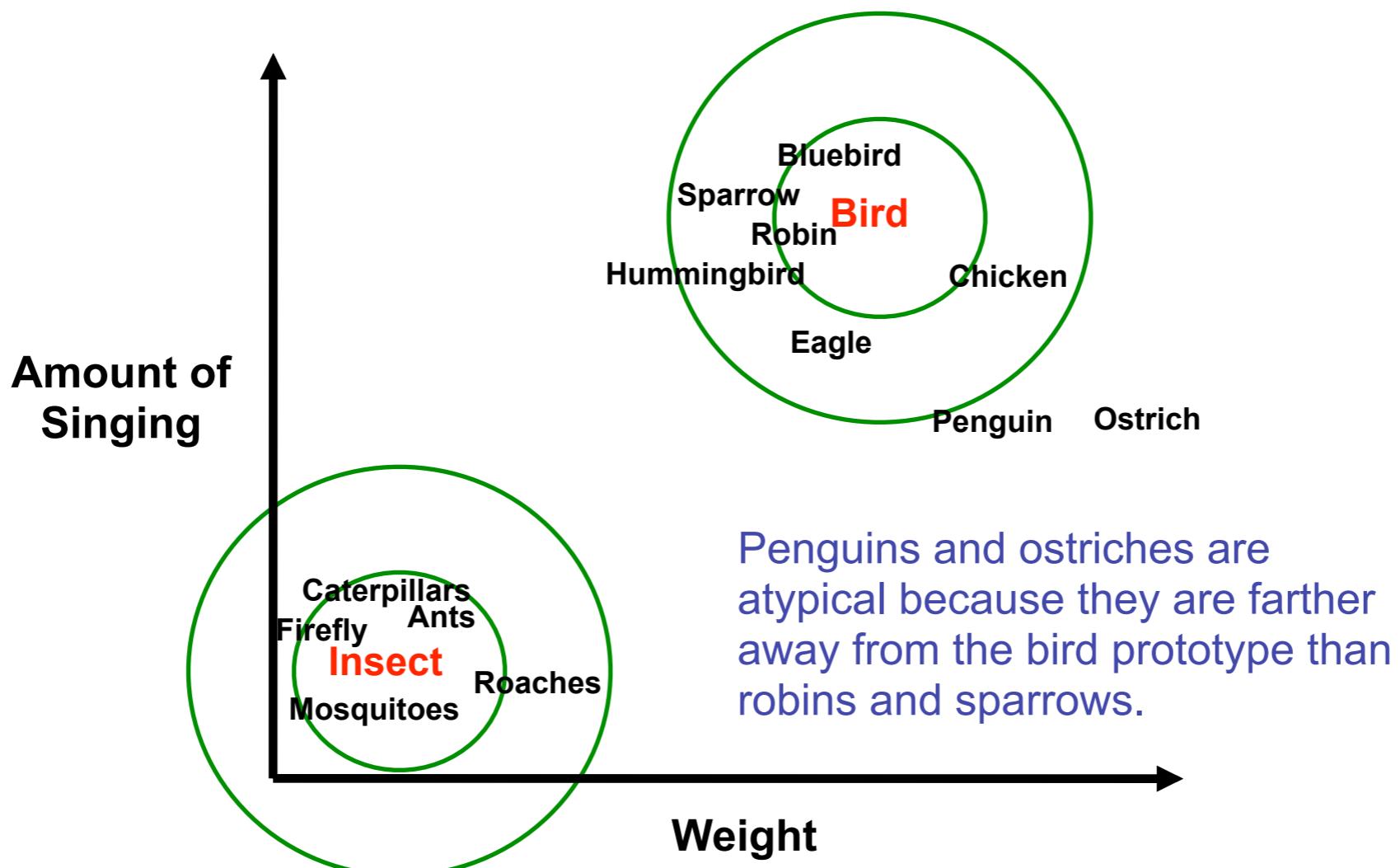
Two key effects: prototype enhancement and borderline cases/graded structure



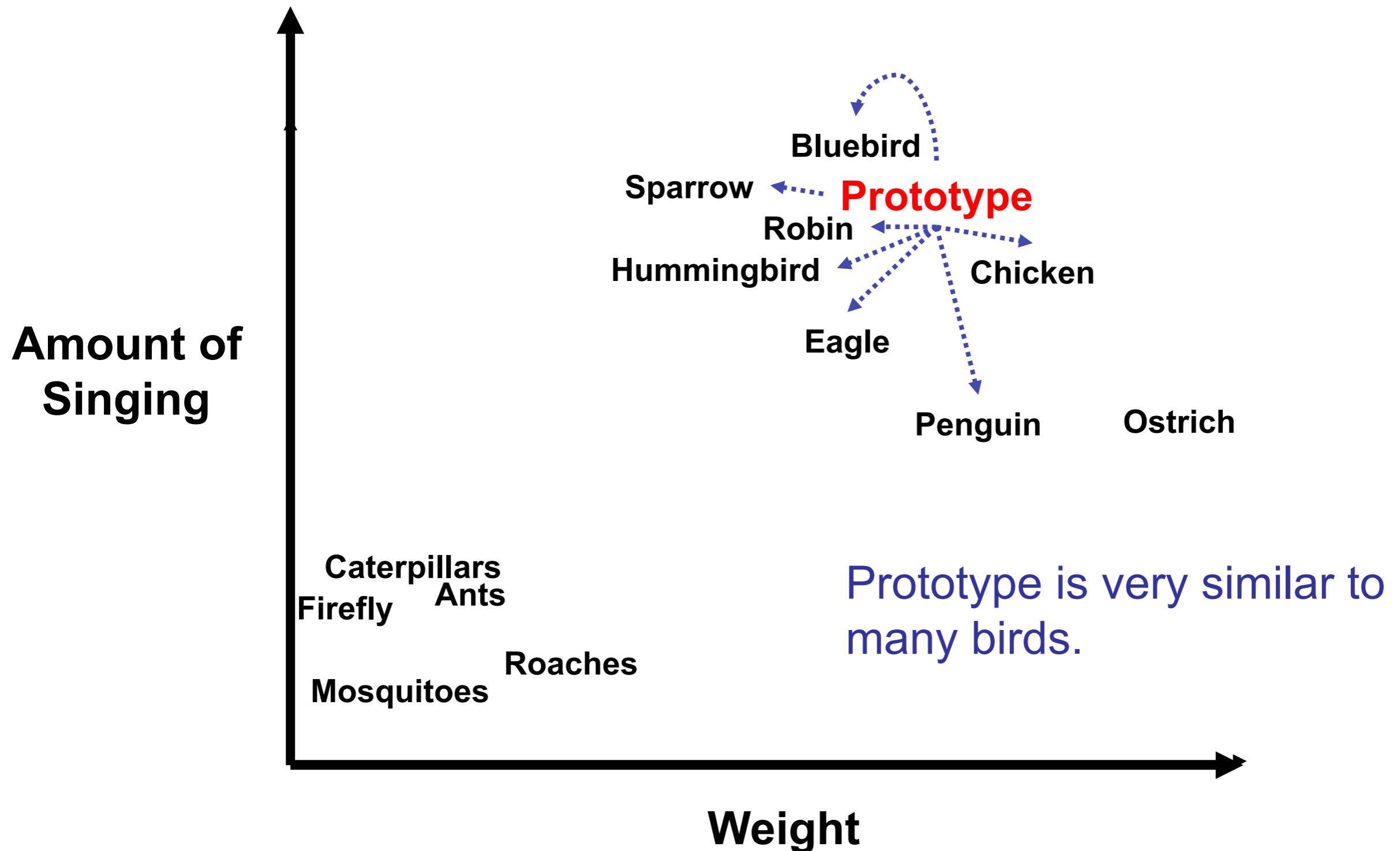
Two key effects: prototype enhancement and borderline cases/graded structure



Two key effects: prototype enhancement and borderline cases/graded structure



However, prototype effects can be explained in terms of exemplar models too!



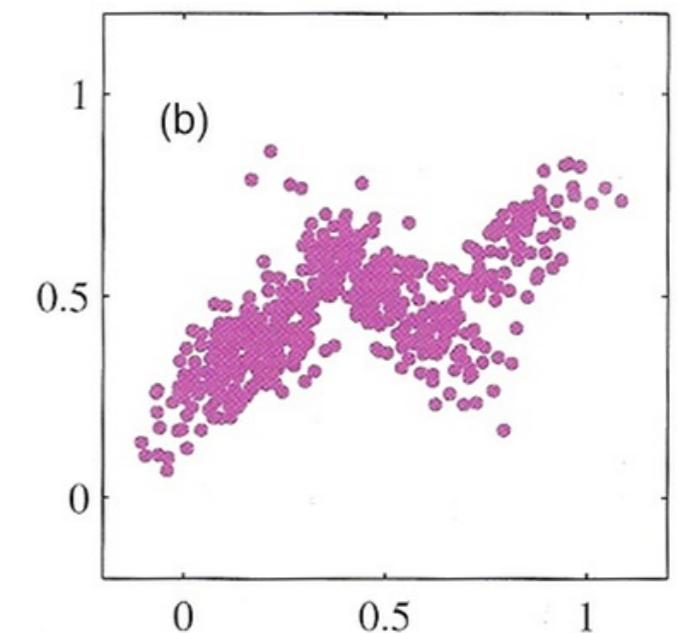
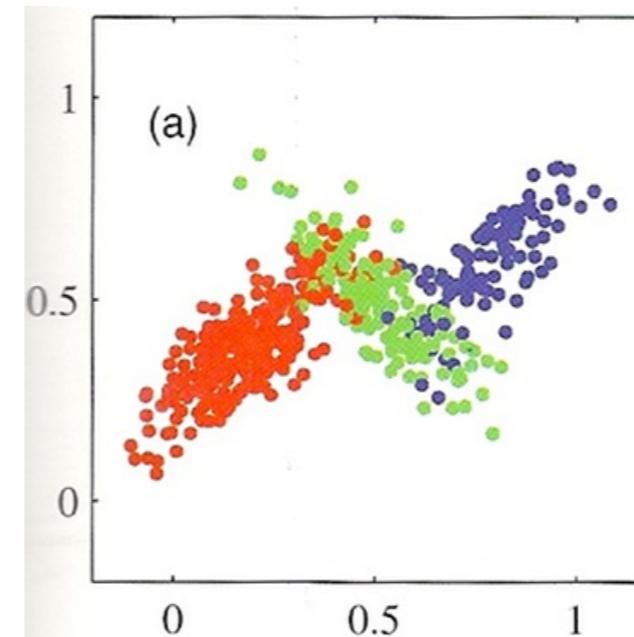
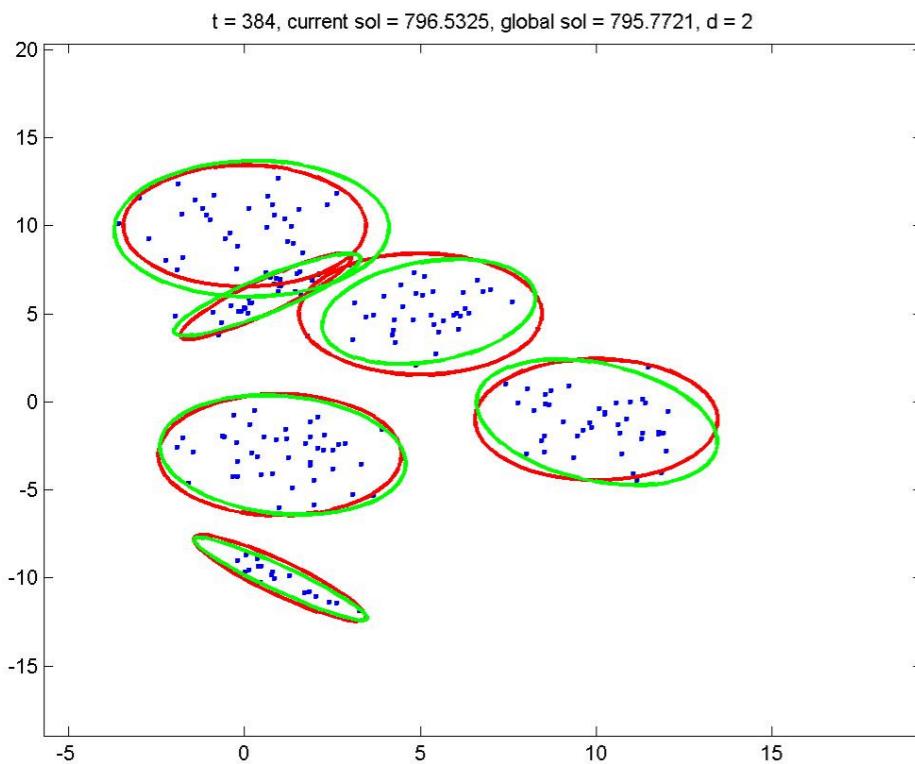
Empirical Effect	Classical View	Prototype Model	Exemplar Model
No defining features	✓	✓	
Borderline cases	✓	✓	
Graded typicality	✓	✓	
Prototype effect	✓	✓	
Exemplar effects			✓

The Exemplar and Prototype Debate

- Exemplar and Prototype Models are titans in the field of cognitive psychology.
- These model are important beyond just the categorization literature because the issues of memory representation and stimulus generalization come up in many areas
 - “Prototypical” or “Average” faces are rated as more attractive (Langlois & Roggman, 1990)
 - The E and P models share deep similarities to Bayesian template-matching models in visual perception (Gold, Cohen, Shiffrin, 2006)
 - In Memory literature: MINERVA (Hintzman, 1988)
 - Speech Perception: Fuzzy Logic Model is a “prototype”-like model (Massaro, 1989); The prototype-magent effect (Kuhl, 1991), “Rich Phonology” (Port, 2007)
- However, is this really all there is?

Case 3: Mixture models

- Problem: You have data that you believe is drawn from N populations
- You want to identify parameters for each population
- You don't know anything about the population a-priori (except maybe Gaussian)
- Fit a set of K Gaussians to the data, compute maximum likelihood over a mixture
- Our first **generative** algorithm because the inferred distribution explicitly models covariance structure of features
- Can be accomplished in an *unsupervised fashion* to pick out clusters which “hang together”



Case 3: Mixture models

- Formally, a mixture model is weighted sum of pdfs where weights are determined by a distribution, π

$$p(x) = \sum_{i=0}^k \pi_i f_i(x)$$

where $\sum_{i=0}^k \pi_i = 1$

$$p(x) = \sum_{i=0}^k \pi_i f_i(x)$$

Case 3: Mixture models

- Gaussian Mixture Model: Special case where each mixture component is a gaussian.

$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \dots + \pi_k N(x|\mu_k, \Sigma_k)$$

where $\sum_{i=0}^k \pi_i = 1$

$$p(x) = \sum_{i=0}^k \pi_i N(x|\mu_k, \Sigma_k)$$

- Typical inference strategy is the Expectation Maximization algorithm:
 - Step 1: Expectation (E-step)
 - Evaluate the “responsibilities of each cluster with current parameters”
 - Step 2: Maximization (M-step)
 - Re-estimate parameters using the existing “responsibilities”
 - Related to k-means clustering

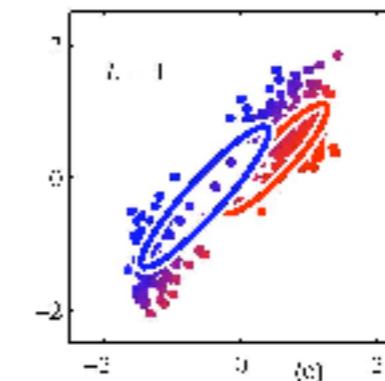
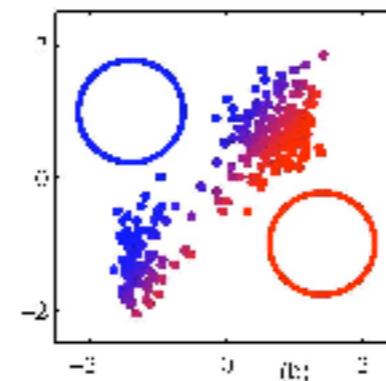
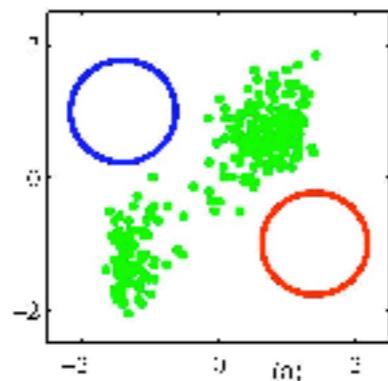
Case 3: Mixture models

- Gaussian Mixture Model: Special case where each mixture component is a gaussian.

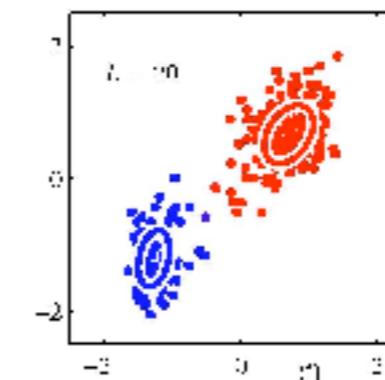
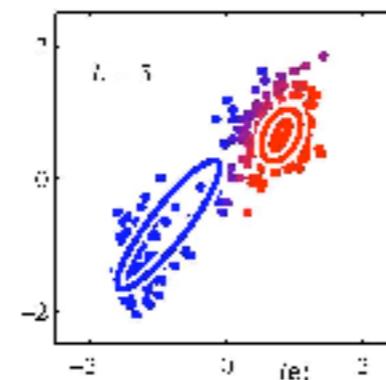
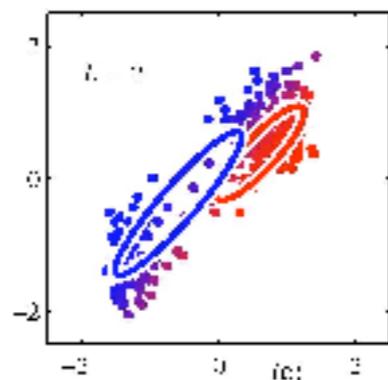
$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \dots + \pi_k N(x|\mu_k, \Sigma_k)$$

where $\sum_{i=0}^k \pi_i = 1$

$$p(x) = \sum_{i=0}^k \pi_i N(x|\mu_k, \Sigma_k)$$



Example:



Case 3: Mixture models

- Issues include...
 - What is the right number of components? (in standard GMM, k is chosen by hand)
 - Singularities when a single data point goes into a component, the inferred variance on this point goes to zero, and as a result the likelihood approaches infinity (this cluster dominates)



- One solution is non-parametric models (let the number of mixture components be determined by the data).
- In this case we assume there is actually an infinite number of latent cluster but assume only a few of them are actually used to generate the data e.g., Chinese Restaurant Process (Aldous, 1985; Pitman, 2002)

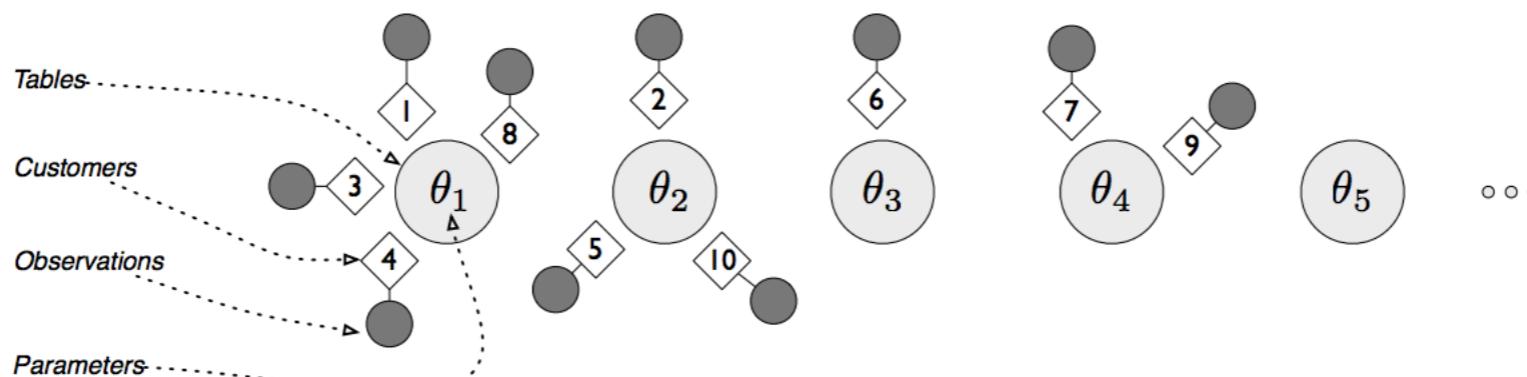
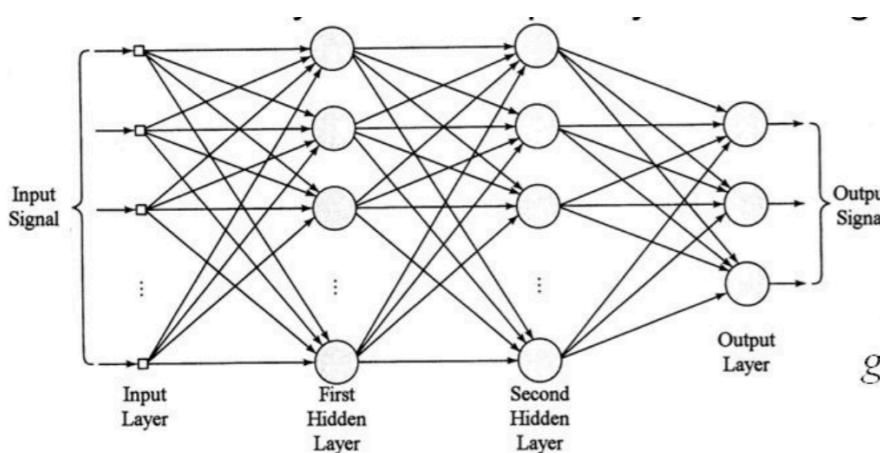


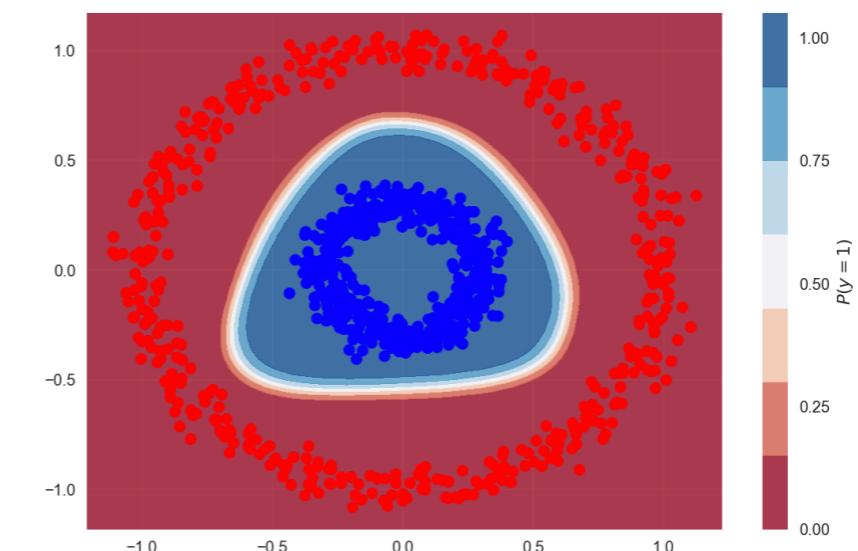
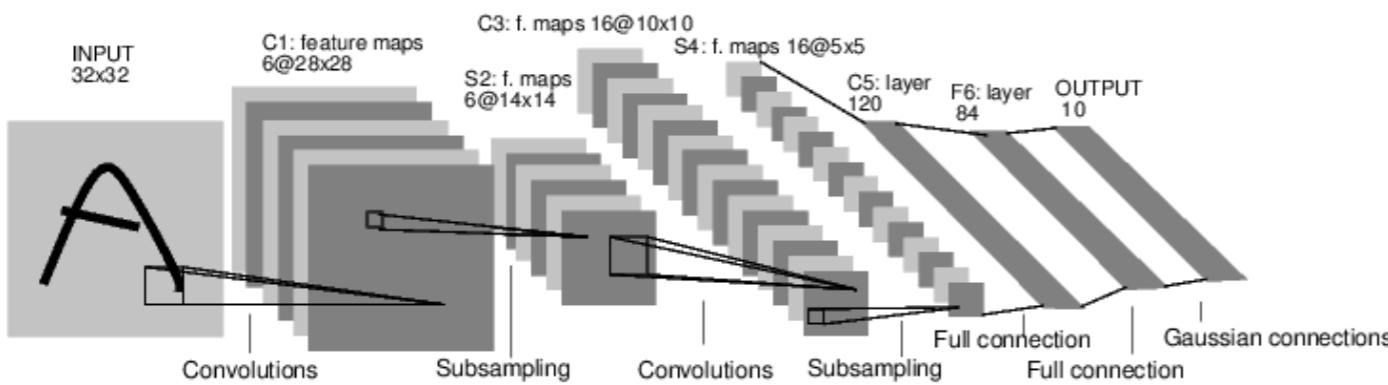
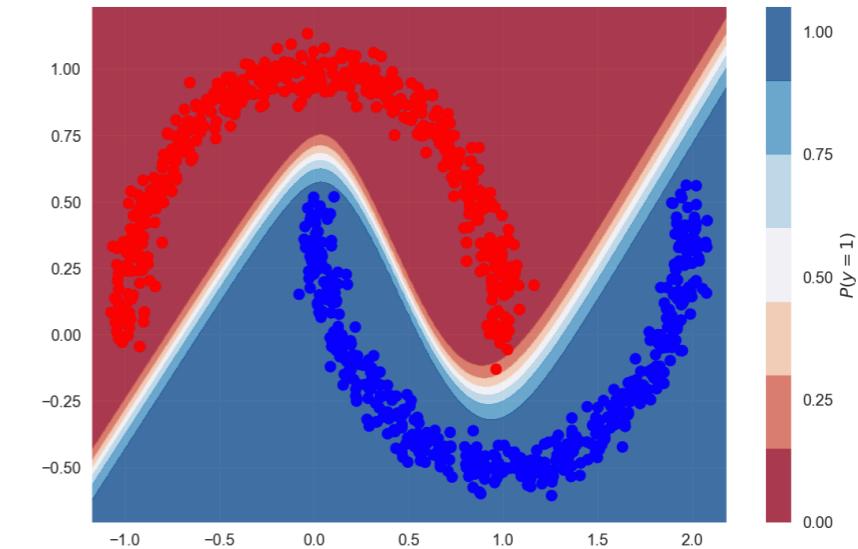
Figure 2: **The Chinese restaurant process.** The generative process of the CRP, where numbered diamonds represent customers, attached to their corresponding observations (shaded circles). The large circles represent tables (clusters) in the CRP and their associated parameters (θ). Note that technically the parameter values $\{\theta\}$ are not part of the CRP *per se*, but rather belong to the full mixture model.

Case 4: Neural Network Models

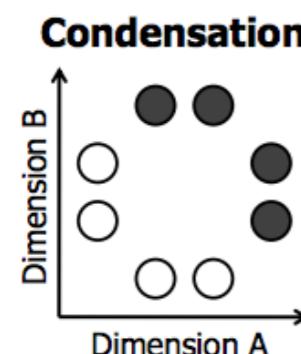
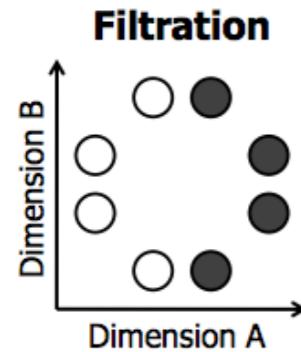
- Multi-layer perceptron
 - One input layer (e.g., stimulus features)
 - One or more hidden layers
 - Out output later roughly coding the category labels
 - Training is based on the Backpropagation Algorithm to minimize prediction error over training set
 - Hidden layer includes differentiable nonlinearities (e.g., sigmoid function)
- **Discriminative** method
- **Supervised** method
- Extremely flexible with additional layer, very complex boundaries can be represented
- When combined with convolutional layer, state of the art on image classification (more on this next week!)



$$g(a) \equiv \frac{1}{1 + \exp(-a)}$$



Is $y = f(x)$ that people used based on neural networks?



- Kruschke (1993) found that filtration type categorization tasks are easier than condensation tasks for humans
- However, they are predicted to be equally difficult for a vanilla multi-layer backprop network (simply a rotation of the space)
- What is going on? Influence of selective attention.

Is $y = f(x)$ that people used based on neural networks?

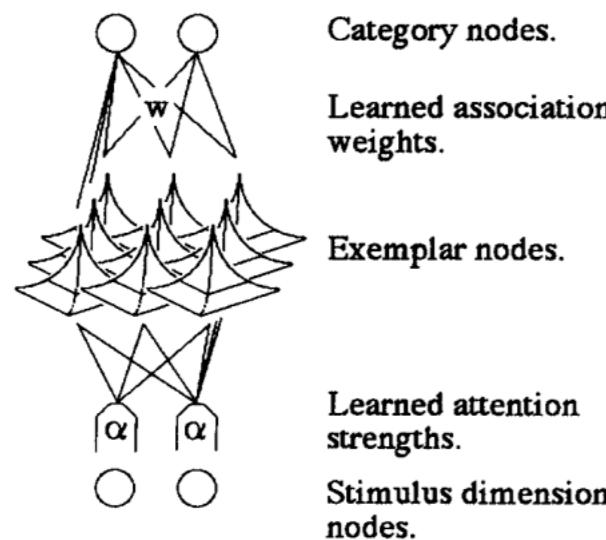


Figure 1. The architecture of ALCOVE (attention learning covering map). (See The Model section.)

- Kruschke (1992) ALCOVE model unifies backdrop networks with exemplar models and includes a selective attention mechanism.
- Layer of hidden units represents exemplars in the task.
- Learned association weight from the exemplar node to categorization labels are adjusted using backprop
- The activation function for the hidden/exemplar nodes is not the standard sum+non-linearity but is a exponential kernel reflecting previous work in psychology on stimulus generalization (e.g., Shepard)
- Attentional weights on the input as also adjusted using the backprop gradient to reduce network error (learns to give more weight to some features than others).



Figure 2. Stretching the horizontal axis and shrinking the vertical axis causes exemplars of the two categories (denoted by dots and xs) to have greater between-categories dissimilarity and greater within-category similarity. (The attention strengths in the network perform this sort of stretching and shrinking function. From "Attention, Similarity, and the Identification-Categorization Relationship" by R. M. Nosofsky, 1986, *Journal of Experimental Psychology: General*, 115, p. 42. Copyright 1986 by the American Psychological Association. Adapted by permission.)



Figure 3. Attentional learning in ALCOVE (attention learning covering map) cannot stretch or shrink diagonally. (Compare with Figure 2)

Is $y = f(x)$ that people used based on neural networks?

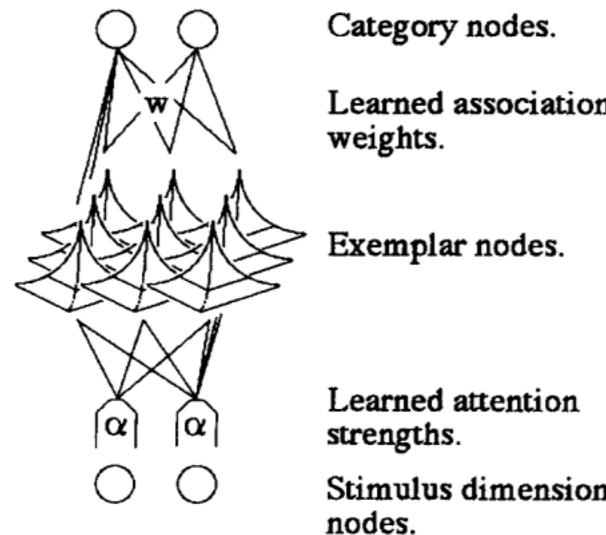


Figure 1. The architecture of ALCOVE (attention learning covering map). (See The Model section.)

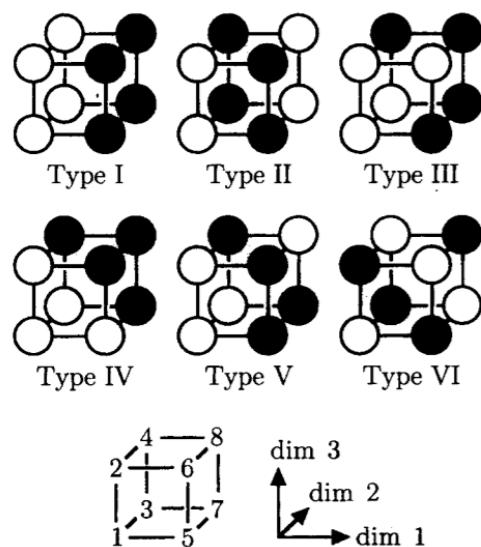


Figure 4. The six category types used by Shepard, Hovland, and Jenkins (1961). (The three binary stimulus dimensions [labeled by the trident at lower right] yield eight training exemplars, numbered at the corners of the lower-left cube. Category assignments are indicated by the open or filled circles. From "Learning and Memorization of Classifications" by R. N. Shepard, C. L. Hovland, & H. M. Jenkins, 1961, *Psychological Monographs*, 75, 13, Whole No. 517, p. 4. In the public domain.)

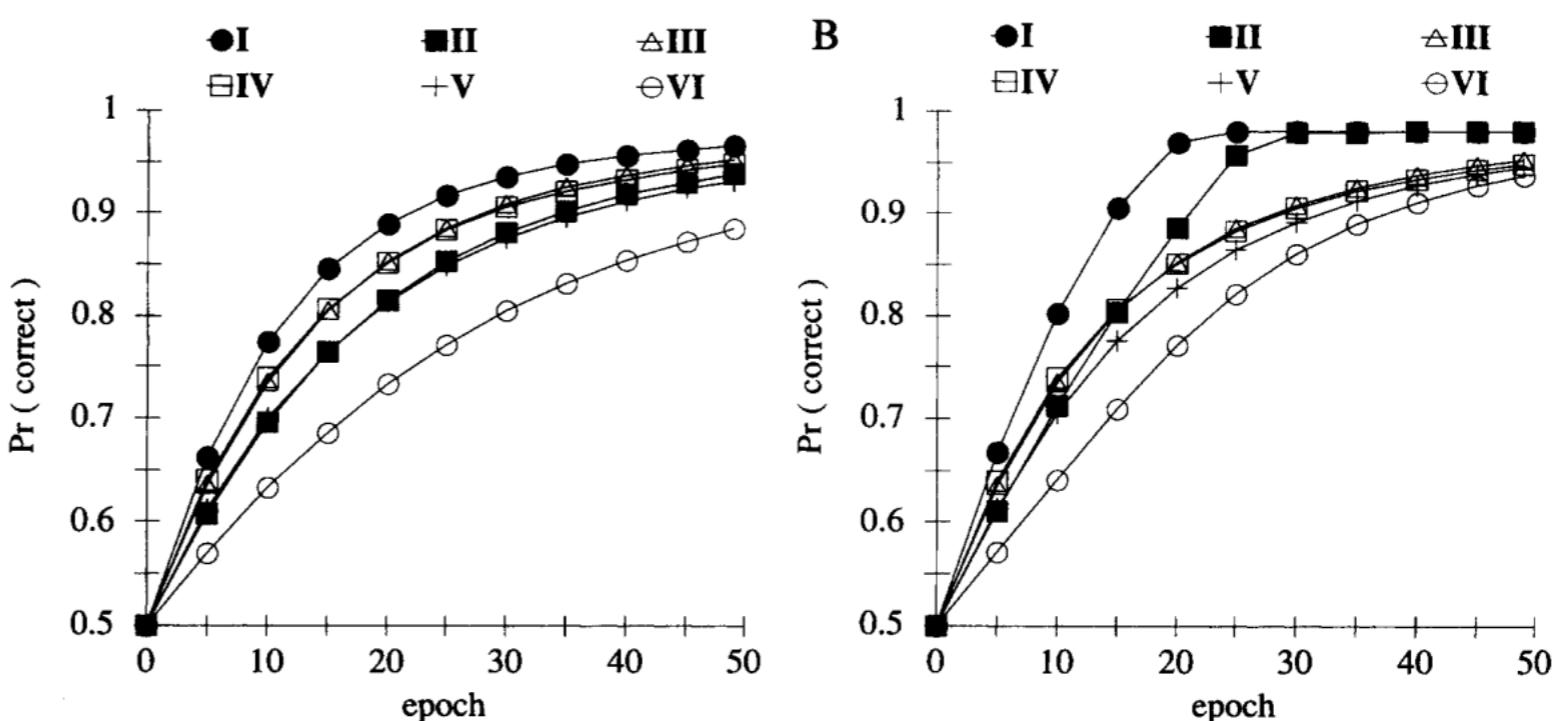
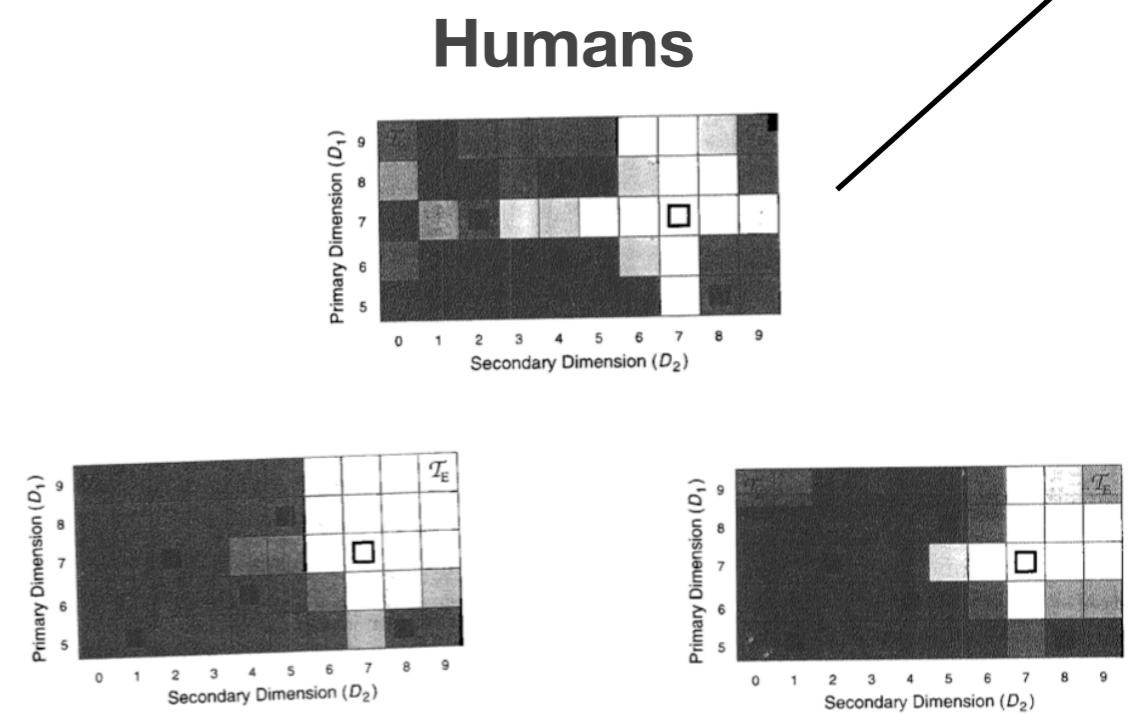
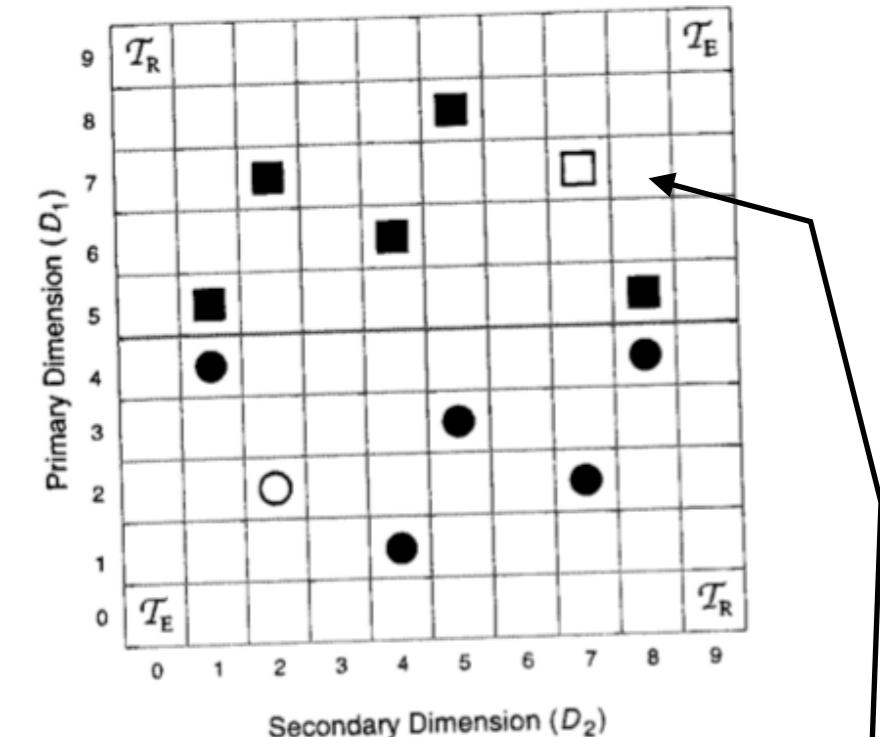
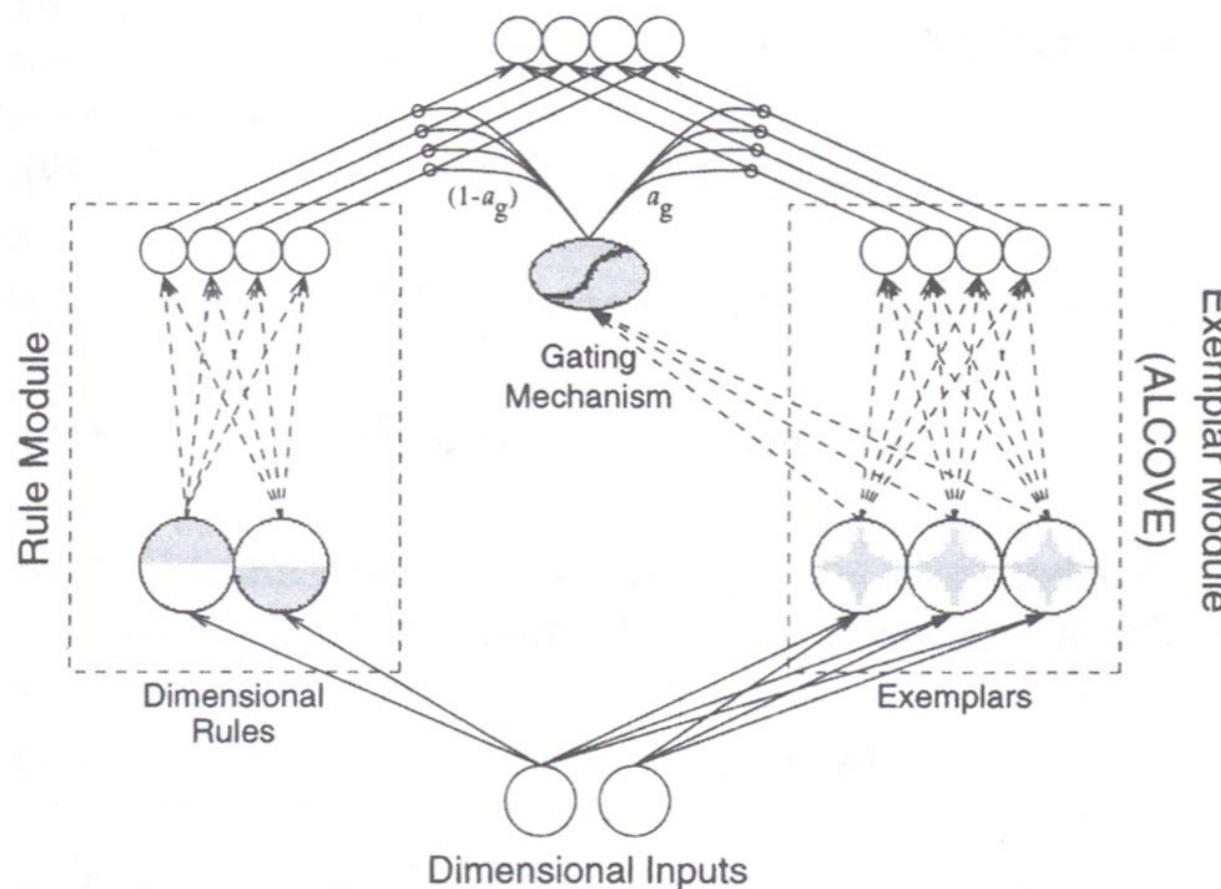


Figure 5. A: Results of applying ALCOVE (attention learning covering map) to the Shepard, Hovland, and Jenkins (1961) category types, with zero attention learning. Here Type II is learned as slowly as Type V (the Type V curve is mostly obscured by the Type II curve). B: Results of applying ALCOVE to the Shepard et al. category types, with moderate attention learning. Note that Type II is now learned second fastest, as observed in human data. Pr = probability.

ATRIUM (Ericsson & Kruschke, 1999)

- Hybrid approaches that combined rule modules and exemplar based mechanisms under a single gradient-based objective.
- Can help to explain interesting patterns of extrapolation outside of the training set.



ALCOVE

ATRIUM

categorization: where human and machine learning meet

- **classification is a central problem in machine learning** (what category does this image show? what topic does this document best fit?)
- many important algorithms developed for this problems (e.g., decision trees, support vector machines, bayes classifiers, deep neural networks, hidden markov models, etc...)
- **what algorithms best characterize how people learn to categorize?**
- theories developed to account for this ability share much in common with classic machine learning approaches and even empirical approaches are similar.

open questions

- **how are people so efficient at learning categories (e.g., we learn a lot from a single example)**
- now that classification algorithms in machine learning are being applied at a larger scale (e.g., millions of images rather than 100s of training examples from the 1990s) what new insights can we get from this work for human psychology?