Simple learning models

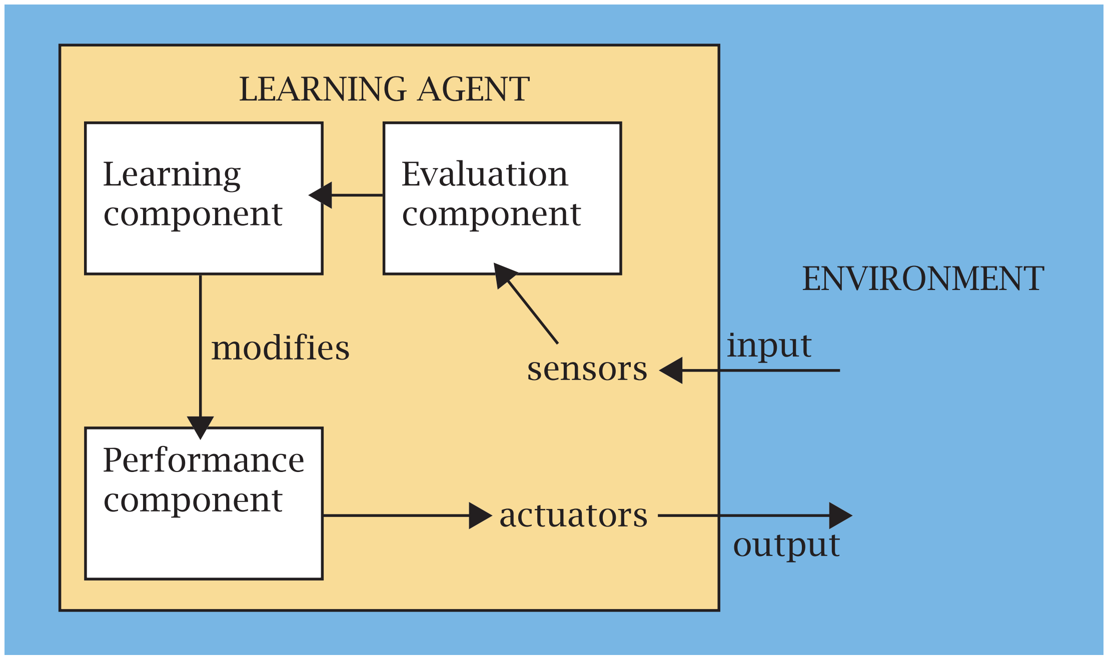
24.964—Fall 2004
Modeling phonological learning

Class 2 (16 Sept 2004)

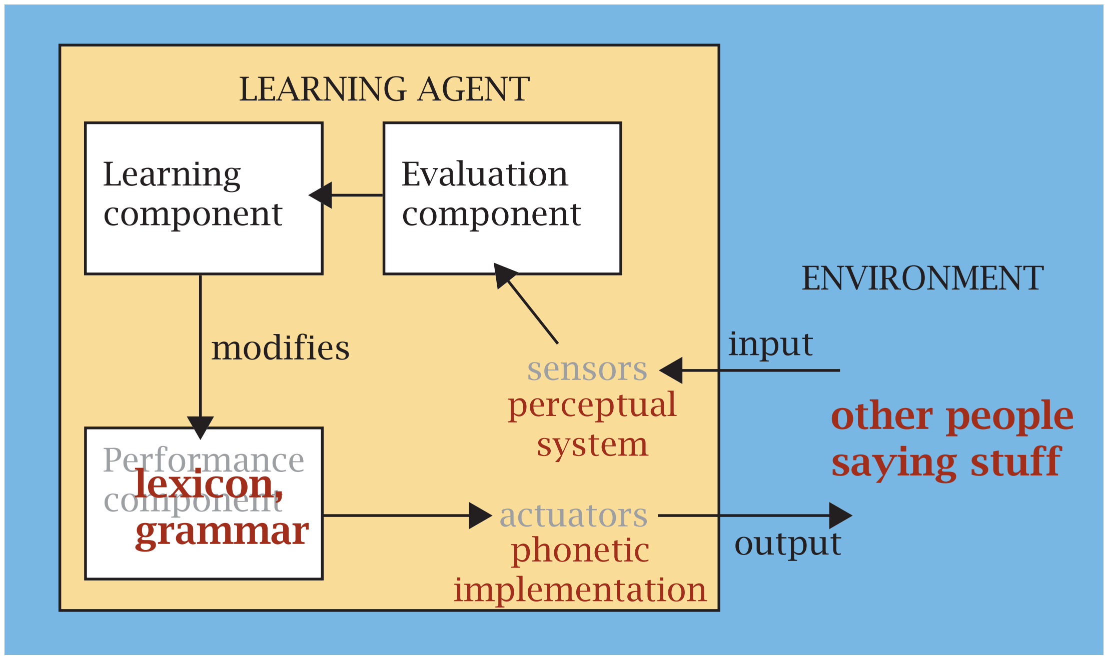
Learning (from the phonology perspective)

Learner hears (existing) words, and learns them Learner notices certain patterns are (or are not) present Constructs grammar to capture those patterns Uses grammar in production of words (both known and new)

Learning (from the AI perspective)



Learning phonology



Learning phonology

Pieces of a learning model:

- Way to receive input
- Way of representing that input
- Way to compare new input with what is already known
- Way to modify what is known (if necessary)
- Way to use knowledge to produce new outputs

Constructing a learning model

A performance component that we saw last week

```
$input_file = "Japanese-ToConvert.txt";
open (INFILE, $input_file) or die "Warning! Can’t open input file: $!\n";

while ($line = <INFILE>) {
    # Crucial rule ordering: this needs to go first
    $line =~ s/hu/fu/g;

    # The major difference is use of <y> after t,s,z
    $line =~ s/ty/ch/g;
    $line =~ s/sy/sh/g;
    $line =~ s/zy/j/g;
    # Also, palatalization before i
    $line =~ s/ti/chi/g;
    $line =~ s/si/shi/g;
    $line =~ s/zi/ji/g;
    # And assibilation of t before u
    $line =~ s/tu/tsu/g;

    print "$line";
}
```

Constructing a learning model

Some easy steps

- Reading in the input
- Doing something to it, and outputting it
- Compare to a “given” answer, to see if the model is right

(See `checkmath.pl` from last week for an example of comparing calculated and given answers; also `hepburn2.pl`)

Terminology

Supervised learning

- Learner is given stimuli (inputs) and also answers (outputs)
- Comparing the input and the output lets the learner see what it needs to learn
- Task is to learn a function converting inputs to their corresponding outputs

Terminology

Unsupervised learning

- Learner receives only input, but no output values
- Model is not told “what to do”
- It looks at the data and tries to find patterns; figure out what types of inputs are likely to occur

Constructing a learning model

We will be looking (primarily) at *supervised* models here

- In this case, the model is given both the input (Monbushô) and output (Hepburn) forms
- Feedback/evaluation: the model produces an output and checks its own answer, to determine whether more learning must occur

The importance of feedback

Error-driven learning

- *Error rate* = (number of errors / number of cases)
  - (1 − error rate) = *accuracy*, or *coverage* of the hypothesis

The importance of feedback

Distinguishing between different types of errors

	Class Positive	Class Negative
Prediction Positive	True Pos	False Pos
Prediction Negative	False Neg	True Neg

- Correct applications: true positive, true negative
- Misclassifications: false positive, false negative

(See `hepburn3.pl`; this program attempts to calculate false positives and false negatives, but can't quite do it accurately—why not?)

Constructing a learning model

The interesting part: actually learning

- We need to be able to modify the performance component somehow
- In the program `hepburn3.pl`, that would require modifying the replacement statements themselves (that is, rewriting the program)
- If we want to modify the statements of the grammar, we must store them in variables somehow

Constructing a learning model

Some programs

- `hepburn4.pl`: storing rules in an array of arrays
- `hepburn5.pl`: reading rules from a file, and "pushing" them onto the array of rules
- `hepburn6.pl`: same, but reads in forms and answers first and stores then, and then runs through them to derive outputs for them

Constructing a learning model

Modifying the grammar itself

- What are the two aspects of the grammar that must be right in order to get the right answer??

Constructing a learning model

Starting in the middle of things:

- Let's assume some rules have been learned, and the goal is to get them in the right order

- In other words, already starting with a hypothesis about the function

Constructing a learning model

More terminology:

- Hypothesis space: set of all possible functions the learner could (in principle) discover to explain the data
- Consistent hypotheses: set of functions that correctly explain the data

Constructing a learning model

Two possible tasks:

- Find a hypothesis that is consistent with the data
- Learn what characterizes the set of consistent hypotheses

Constructing a learning model

What is the hypothesis space for grammars of ordered rules? (assuming, for the moment, that the rules themselves are fixed)

- What is the set of possible hypotheses?
- How big is it

Comparison: decision lists

A similar problem with an even larger search space

- Decision lists: predicting an outcome based on a series of yes/no or forced choice answers
- Demo: http://www.cs.ubc.ca/labs/lci/CIspace/Version3/dTree/index.html

Comparison: decision lists

“All electronics” data set

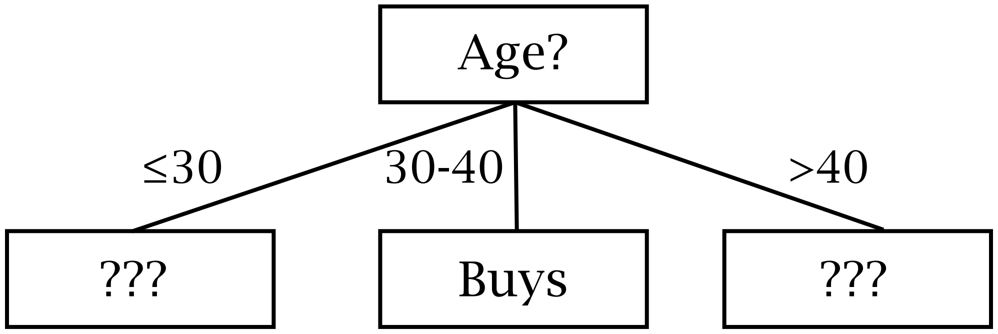
Age	Income	Student?	Credit	Buys
$\leq$30	high	no	fair	no
$\leq$30	high	no	excellent	no
30-40	high	no	fair	yes
$>$40	med	no	fair	yes
$>$40	low	yes	fair	yes
$>$40	low	yes	excellent	no
30-40	low	yes	excellent	yes
$\leq$30	med	no	fair	no
$\leq$30	low	yes	excellent	yes
$>$40	med	yes	fair	yes
$\leq$30	med	yes	excellent	yes
30-40	med	no	excellent	yes
30-40	high	yes	fair	yes
$>$40	med	no	excellent	no

Comparison: decision lists

Predictive power of factors:

Factor	Level	How many buy
Age	≤30	2/5
	30-40	4/4
	>40	3/5
Income	low	3/4
	med	4/6
	high	2/4
Student	yes	6/7
	no	3/7
Credit	fair	5/7
	excellent	4/7

Comparison: decision lists



Comparison: decision lists

Step 2

Age	Income	Student?	Credit	Buys	Explained
$\leq$30	high	no	fair	no	
$\leq$30	high	no	excellent	no	
30-40	high	no	fair	yes	✓
$>$40	med	no	fair	yes	
$>$40	low	yes	fair	yes	
$>$40	low	yes	excellent	no	
30-40	low	yes	excellent	yes	✓
$\leq$30	med	no	fair	no	
$\leq$30	low	yes	excellent	yes	
$>$40	med	yes	fair	yes	
$\leq$30	med	yes	excellent	yes	
30-40	med	no	excellent	yes	✓
30-40	high	yes	fair	yes	✓
$>$40	med	no	excellent	no	

Comparison: decision lists

Step 2

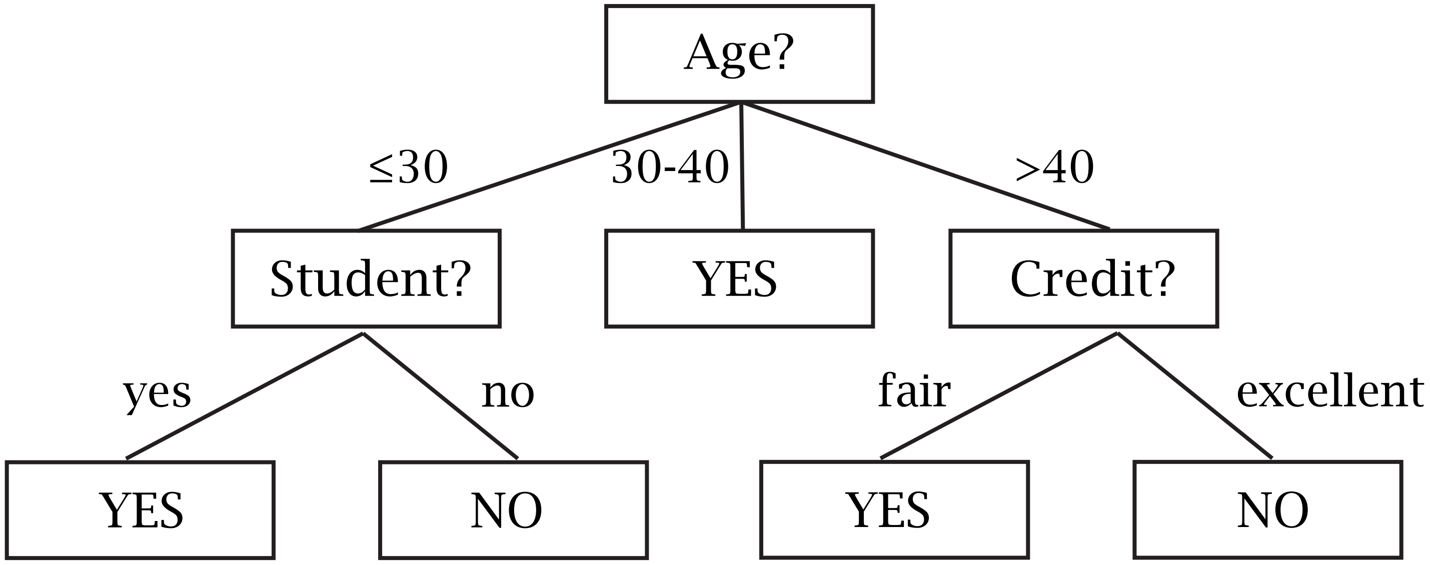
Age	Income	Student?	Credit	Buys	Explained
$\leq$30	high	no	fair	no	
$\leq$30	high	no	excellent	no	
30-40					
$>$40	med	no	fair	yes	
$>$40	low	yes	fair	yes	
$>$40	low	yes	excellent	no	
$\leq$30	med	no	fair	no	
$\leq$30	low	yes	excellent	yes	
$>$40	med	yes	fair	yes	
$\leq$30	med	yes	excellent	yes	
$>$40	med	no	excellent	no	

Comparison: decision lists

Predictive power among remaining cases:

Factor	Level	$\leq$30 buy	$>$40 buy
Income	low	1/1	1/2
	med	1/2	2/3
	high	0/2	0/0
Student	yes	2/2	2/3
	no	0/3	1/2
Credit	fair	2/3	3/3
	excellent	0/2	0/2

Comparison: decision lists



Back to ordered rules

Why would this same strategy not work for ordered rules?

Learning rule orderings

- Naive first approach: `hepburn7.pl` (why does this work relatively well?)
- Comparison: `italian.pl`

Reducing the hypothesis space for ordered rules

What are some principles that would reduce the number of possibilities to explore for ordered rules?

The Elsewhere condition (Kiparsky 1982)

Rules A, B in the same component apply disjunctively to a form ϕ if and only if

1. The structural description of A (the special rule) includes the structural description of B (the general rule)

2. The result of applying A to ϕ is distinct from applying B to ϕ In that case, A is applied first, and if it takes effect, then B is not applied.

Does the Elsewhere condition reduce search space?

An attempt to test this (imperfectly): `italian2.pl`

For next week

- Assignment 2: two parts (see handout)
- Readings: (on course website)
  - Hutchinson, chapter 1
  - Weiss & Kulikowski, chapter 2
    - Focus on the general concepts and techniques presented here (don’t worry about all the mathematical details)