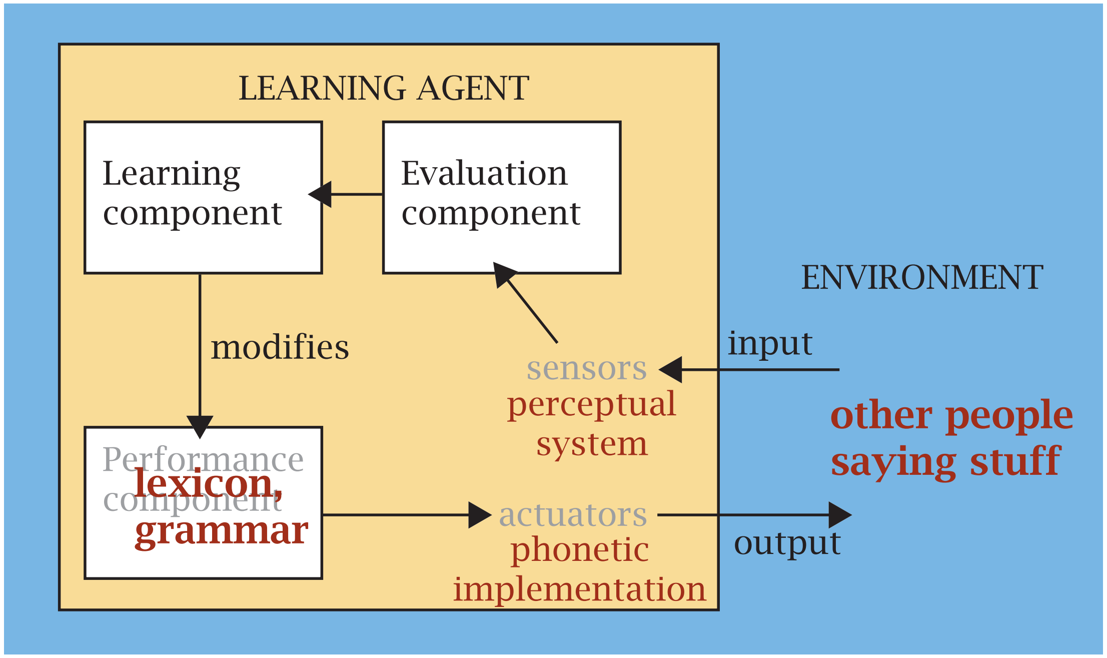
Learning models for phonology

24.964—Fall 2004
Modeling phonological learning

Class 3 (23 Sept, 2004)

Reminder: components of a learning module



A very stupid learner

`italian.pl` from last week:

```
Start with a predefined set of rules in a particular order;
While (some forms are derived incorrectly) {
   Pick two rules at random;
   Swap the two rules in the ordered grammar;
   For each input form {
       Use current grammar to derive an output;
       Compare output to correct (given) output;
       Score 1 if correct;
   }
}
```

A very stupid learner

Review:

- What is the hypothesis space?
- How big is it?
- What are some reasons why this approach is so inefficient?

What might be some strategies to explore the hypothesis space to arrive at a solution more efficiently?

One possible approach

```
Start with a predefined set of rules in a particular order;
See how many correct to start with;
while (not everything is correct) {
    from (R1 = last rule to second rule) {
         from (R2 = rule before R1 to first rule) {
            if (R1 and R2 could potentially interact AND
                haven't already tried swapping them before) {
               Try putting R1 before R2;
               if (number correct is greater than before) {
                    keep the new ordering;
               } else {
                    make a note of the failed ordering;
                    revert to previous state;
               }
            }
         }
    }
}
```

Characterizing the learning task

‘Right’ answer vs optimal answer/convergence to an adequate answer

- What is the actual goal of learning phonology? Is there a right answer? What is an adequate answer? Does time to convergence play a role in any way?

Characterizing the learning task

Open vs. closed domain

- Does `italian3.pl` operate on an open or closed domain?
- How about someone solving a phonology problem set?

Is this even a useful distinction for us? How would it impact building a learning model?

Characterizing the learning task

Clean vs noisy data

- Does `italian3.pl` assume clean or noisy data?
- Are problem sets clean or noisy?

How about irrelevant factors (like meanings of the words): are they noise?

Characterizing the learning task

Negative evidence

- Does `italian3.pl` use negative evidence?
- How about learners solving problem sets?

Plausible vs. implausible sources of negative evidence

Characterizing the learning task

Size of the training set

- Hutchinson: "the more sophisticated algorithms learn a lot from just a few examples (maybe just one)"
- What size training sets do we usually use in phonology? What type of training set is available to the human learner?

Why does size of training set matter?

Excursus: why would we even care about the size of the training set?

- What are some problems that would arise in a small training set?

Why does size of training set matter?

Two problems that arise with small data sets

- Sampling error: not representative of the population as a whole
- Ambiguity: data is representative, but there are not enough cases to distinguish similar hypotheses

Error rate estimation

$$\textit{apparent error rate} = \frac{\text{Errors on training sample}}{\text{Number of items in training sample}}$$

Error rate estimation

$$\textit{apparent error rate} = \frac{\text{Errors on training sample}}{\text{Number of items in training sample}}$$

- Weiss & Kulikowski, p.24: "With an unlimited design sample used for learning, the apparent error rate will itself become the true error rate eventually. However, in the real world, we usually have relatively modest sample sizes with which to design a classifier and extrapolate its performance to new cases. For most types of classifiers, the apparent error rate is a poor estimator of future performance.. In general, apparent error rates tend to be biased optimistically. The true error rate is almost invariably higher than the apparent error rate."

Error rate estimation

Two sources of inaccuracy in estimating error rate from the training set (reclassification):

- Training samples are finite
  - ○ Sample might be too small, and may differ from the true error rate simply because of probability
  - ○ (The sample wasn't truly representative)
- Hypothesis learned from the sample was too short-sighted
  - ○ Captures the training data well, but won't extend correctly to further examples
  - ○ *Overfitting*, or *overspecialization*
  - ○ What might an "overspecialized" solution be in phonology?

Error rate estimation

Dealing with error from uncertainty due to small sample size: confidence intervals

Weiss & Kulikowski's rule of thumb: by the time the sample size reaches 1000, the estimate is "extremely accurate" (By 5000, it's essentially equal to the true rate)

Error rate estimation

Dealing with short-sighted hypotheses (overfitting)

- Separate training data vs. test data ("holdout")
  - Weiss & Kulikowski suggest convention of 2/3 training to 1/3 test
  - Target size of testing set is 1000 or greater (why?)
  - Proportion therefore must vary according to how much data is available

- Cross-validation
  - "Leave-one-out"
  - k-fold cross validation (k usually = 10)

Error rate estimation

Weiss & Kulikowski's general purpose suggestions:

- For $n > 100$, use cross validation (10-fold or leave-one-out)
- For $n < 100$, use leave-one-out
- For $n < 50$, try repeated 2-fold CV, or the .632 bootstrap:
- Take a sample of n items with resampling
- Train on that sample, test on anything not gotten in the sample to calculate error rate (e0)

  - ○ (On avg, that will lead to .632 samples chosen, .368 in test batch)

- e0 can also be approximated by repeated 2-fold cross validation (for reasons that are not clear to me)

- .632B = .368*e_{app} + .632*e0, where e_{app} = apparent error rate when trained on all cases

(Other than being quite complicated, why would we not want to do things like this for phonology data sets with less than 50 items?)

Error rate estimation

Back to Weiss & Kulikowski, p.24:

“With an unlimited design sample used for learning, the apparent error rate will itself become the true error rate eventually. However, in the real world, we usually have relatively modest sample sizes with which to design a classifier and extrapolate its performance to new cases. For most types of classifiers, the apparent error rate is a poor estimator of future performance.. In general, apparent error rates tend to be biased optimistically. The true error rate is almost invariably higher than the apparent error rate.”

- Is it always the case that the apparent error rate will become the true error rate with an unlimited design sample? When might it not be?
  - May depend on what we mean by "true"
- When might resubstitution give a HIGHER error rate than cross-validation?

Error rate estimation

Weiss & Kulikowski, p. 46: Common mistakes

- Testing on the training data
  - ○ "This mistake occurs infrequently these days, except perhaps for overzealous money managers promoting their successfully backfitted investment scheme"

- "Estimates for small sample sizes are much less reliable than those for large samples. A common oversight is that while the overall sample size may not be considered small, the subsample for any given class may be small. If the error rate on that class is particularly important, then the sample must be analyzed as a small sample."

Error rate estimation

Stepping back a minute:

☞ Why do all of these techniques seem sort of inapplicable to the task of learning a phonological grammar?

- What might we learn from them, even if we don't intend to use them directly?

Characterizing the learning task

Order of examples

- Could the order of examples matter for `italian3.pl`
- Could the order of examples matter for phonologists solving problem sets?

Characterizing the learning task

Does learning ever stop?

- In `italian3.pl` ?
- In solving a problem set?

Characterizing the learning task

Is the learned information accessible for inspection/analysis?

- In `italian3.pl` ?
- In solving a problem set?

Why would we care?

Characterizing the learning task

Form of data:

- Small sets of well defined, distinct attributes vs. large sets of similar attributes

(How does the data in `italian3.pl` differ from data in a phonology problem?)

Characterizing the learning task

What is the solution space? How big is it?

- For `italian3.pl`

- For a problem set

Characterizing the learning task

Is learning incremental or done in batch?
(Poorly described in Hutchinson; usually used in more common-sense way to specify whether all the data is required in advance)

- In `italian3.pl` ?
- In solving a problem set?

Characterizing the learning task

Is learning supervised? (Model gets both the data and the right answers to learn from)

- For `italian3.pl`
- For a problem set

Learning of phonology by human learners

Extremely broad brushstroke:

- 0 months
  - ○ No lexicon or morphology
  - ○ Some knowledge of prosody
  - ○ Whatever biases/constraints/boundaries are innate
- 6 months
  - ○ Still no words or morphology
  - ○ Showing sensitivity to L1 phonological categories (“perceptual magnet” effect; Kuhl)

Learning of phonology by human learners

Extremely broad brushstroke:

- 8-10 months
  - Words? (probably still little or no morphology)
  - Lose ability to distinguish non-native constrasts (Werker and colleagues)
  - Knowledge of native inventory, and also some sequencing constraints (Jusczyk and colleagues)

Learning of phonology by human learners

Extremely broad brushstroke:

- Beyond the first year
    - Lexicon expands rapidly
    - Morphology lags behind for quite some time
        - English-learning two-year olds don't necessarily have command of plural suffix (Smith 1973)
        - Even 4-year olds aren't always very good with it (Berko 1958, and subsequent work by Derwing and others)

What this means for us

- The earliest phonological learning operates without much in the way of "higher level" knowledge
  - Domain is more closed than it might otherwise be: no semantics, syntax, morphology, etc.
- Knowledge of categories precedes phonotactic knowledge
  - Reasonable to assume that learner operates over representations of some time (?)
- Surface phonotactic learning precedes learning alternations
  - Mechanism for learning alternations could make use of knowledge of phonotactics

So how might we characterize real learning?

(Phonotactic learning, and learning alternations)

- Open or closed domain?
- Data: Clean or noisy data? Negative evidence? How big is the training set?
- Does order matter?
- Is learning incremental, or batch?
- Does learning ever stop?
- Supervised, or unsupervised?

Realism of learning models

Hutchinson, p. 2:

> "There are two approaches to artificial learning. The first is motivated by the study of mental processes and says: *It is the study of the algorithms embodied in the human mind, and how they can be translated into formal languages and programs*.
>
> The second is much more mundane. It arises from practical computing, which ostensibly has nothing to do with psychology: *It is a branch of data processing, concerned with programs which extrapolate from past data and alter their behaviour accordingly*.
>
> I claim that the second is the right approach. Psychology is a valuable motivator, but writing a program is so very unlike any task that a psychologist ever faces that practitioners of the two subjects are likely to mislead each other. A program is best viewed first as an algorithm acting on data and then perhaps later as an embodiment of some attempt at psychological reality."

(Do you agree? What might some plausible intermediate stances be?)

Getting started

`findinventory.pl`, `findpairs.pl`

Assignment 3, for next week (9/30)

- Readings:
  - Jusczyk, Luce, and Charles-Luce (1994) Infants sensitivity to Phonotactic Patterns
  - Kessler and Treiman (1997) Syllable Structure and the Distribution of Phonemes

Assignment 3, for next week (9/30)

- Programming:
  The Jusczyk, Luce, and Charles-Luce study employed sets of monosyllables which were claimed to have high and low phonotactic probabilities in English. Your task is to check their claim, by computing the phonotactic probability of their test items. There is a file called CelexWordsInTranscription.txt, which contains a list of English words. Your task is to write a program that reads in this file, computes the probabilities of their items, by the criteria used in that study. (That is, by the "positional" probabilities). You will need to perform several sub-tasks:
  - You will need to figure out how to break the syllables up into onsets, nuclei, and rhymes (a key to the symbols that are used is provided on the web site along with the file)
  - You will need to calculate the probability of each phoneme in each position
  - You will then need to find a way to translate these individual probabilities to a single score for the entire word
  - I will provide a test file with the Jusczyk et al stimuli, that you can run your program on to see what their scores are.