# All the particular properties that give a language its unique phonological character can be expressed in numbers.
# -Nicolai Trubetzkoy

John Goldsmith
University of Chicago

September 19, 2005

# Probabilistic phonology

Why a phonologist should be interested in probabilistic tools for understanding phonology, and analyzing phonological data…

- Because probabilistic models are very powerful, and can tell us much about data even without recourse to structural assumptions, and
- Probabilistic models can be used to teach us about phonological structure.

The two parts of today’s talk will address each of these.

# Automatic learning of grammars

Automatic learning of grammars: a conception of what linguistic theory *is.*

Automatic learning techniques:

- In some respects they teach us *more*, and in some respects they teach us *less,* than non-automatic means.
- Today’s talk is a guided tour of some applications of known techniques to phonological data.

# Probabilistic models

- Are well-understood mathematically;
- Have powerful methods associated with them for learning parameters from data;
- Are the ultimate formal model for understanding *competition*.

# Essence of probabilistic models:

- Whenever there is a choice-point in a grammar, we must assign degrees of *expectedness* of each of the different choices.
- And we do this in a way such that these quantitites add up to 1.0

# Frequencies and probabilities

- **Frequencies** are numbers that we observe (or count);
- **Probabilities** are parameters in a theory.
- We can set our probabilities on the basis of the (observed) frequencies; but we do not *need* to do so.
- We often do so for one good reason:

# Maximum likelihood

- A basic principle of empirical success is this:
  - Find the probabilistic model that assigns the highest probability to a (pre-established) set of data (observations).
- Maximize the probability of the data.

# Brief digression on Minimum Description Length (MDL) analysis

- Maximizing the probability of the data is not an entirely satisfactory goal: we also need to seek economy of description.
- Otherwise we risk *overfitting* the data.
- We can actually define a better quantity to optimize: this is the *description length*.

# Description Length

- The description length of the analysis A of a set of data D is the sum of 2 things:
  - The *length* of the grammar in A (in “bits”);
  - The *(base 2) logarithm of the probability assigned to the data D*, by analysis A, times -1 (“log probability of the data”).
- When the probability is high, the “log probability” is small; when the probability is low, the log probability gets large.

# MDL (suite)

- If we aim to minimize the sum of the description length ( = length of the grammar, as in 1st generation generative grammar) + log probability (data), then we will seek the best overall grammatical account of the data.

# Morphology

- Much of my work over the last 8 years has been on applying this framework to the discovery of morphological structure.
- See http://linguistica.uchicago.edu
- Today, though: phonology.

# Assume structure?

- The standard argument for assuming structure in linguistics is to point out that there are empirical generalizations in the data that cannot be accounted for without assuming the existence of the structure.

- Probabilistic models are capable of modeling a great deal of information without assuming (much) structure, and
- They are also capable of *measuring* exactly how much information they capture, thanks to information theory.
- Data-driven methods might be especially of interest to people studying dialect differences.

# Simple segmental representations

- “Unigram” model for French (English, etc.)
- Captures only information about segment frequencies.
- The probability of a word is the product of the probabilities of its segments.
- Better measure: the complexity of a word is its average log probability:

$$\frac{1}{length(W)} \sum_{i=1}^{length(W)} -\log_2 prob(w_i)$$

# Let’s look at that graphically…

- Because *log probabilities* are much easier to visualize.
- And because the log probability of a whole word is (in this case) just the sum of the log probabilities of the individual phones.

# Add (1st order) conditional probabilities

- The probability of a segment is conditioned by the preceding segment.
- Surprisingly, this is mathematically equivalent to *adding* something to the “unigram log probabilities” we just looked at: we add the “mutual information” of each successive phoneme.

$$MI(pq) = \log \frac{prob(pq)}{prob(p)\,prog(q)}$$

# Let’s look at that

# Complexity = average log probability

- Find the model that makes this equation work the best.
- Rank words from a language by complexity:
  - Words at the top are the “best”;
  - Words at the bottom are…what?

borrowings,
onomatopeia,
rare phonemes,
and errors.

- The pressure for nativization is the pressure to rise in this hierarchy of words.
- We can thus define the direction of the phonological pressure…

# Nativization of a word

- Gasoil [gazojl] or [gazọl]
- Compare average log probability (bigram model)
  - – [gazojl] 5.285
  - – [gazọl] 3.979
- This is a huge difference.
- Nativization *decreases the average log probability of a word.*

# Phonotactics

- Phonotactics include knowledge of 2nd order conditional probabilities.
- Examples from English…

1 stations
2 hounding
3 wasting
4 dispensing
5 gardens
6 fumbling
7 telesciences
8 disapproves
9 tinker
10 observant
11 outfitted
12 diphtheria

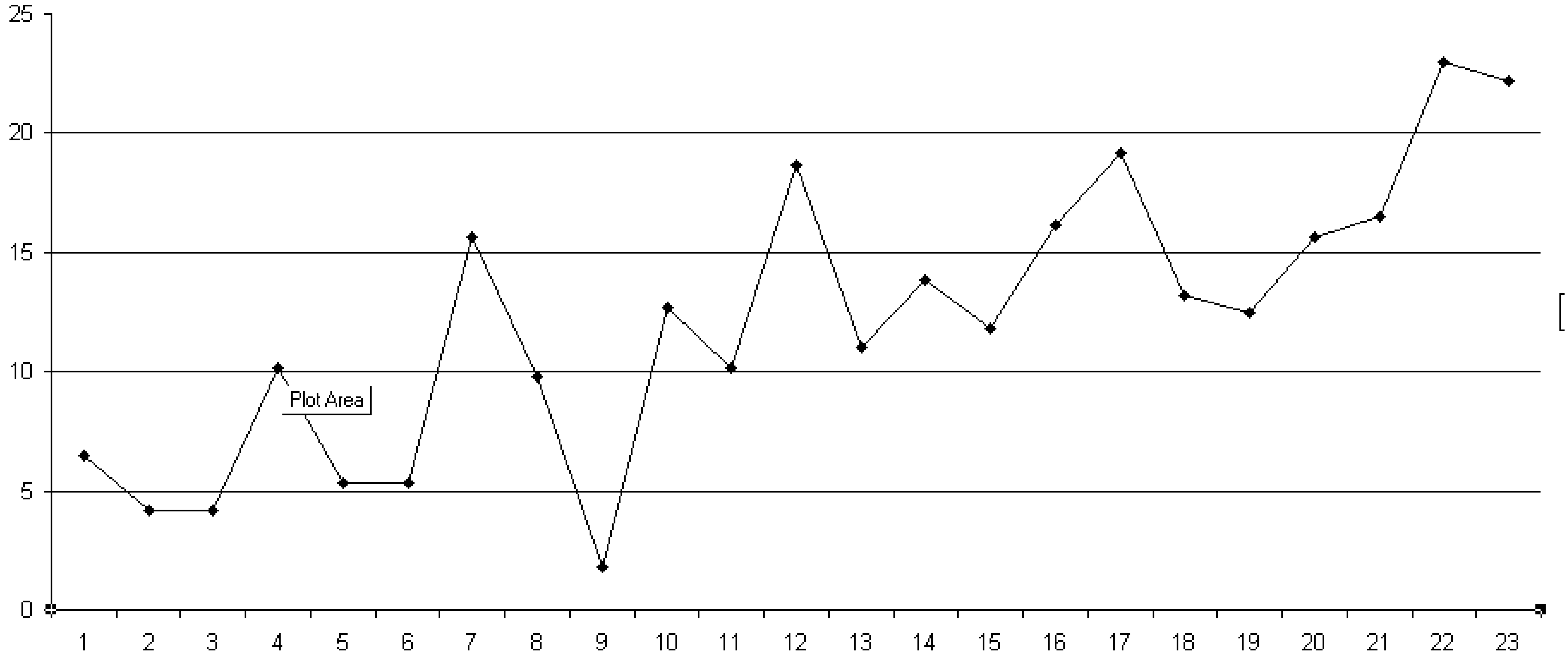
13 voyager
14 schafer
15 engage
16 Louisa
17 sauté
18 zigzagged
19 Gilmour
20 Aha
21 Ely
22 Zhikov
23 kukje

But speakers didn't always agree. The biggest disagreements were:

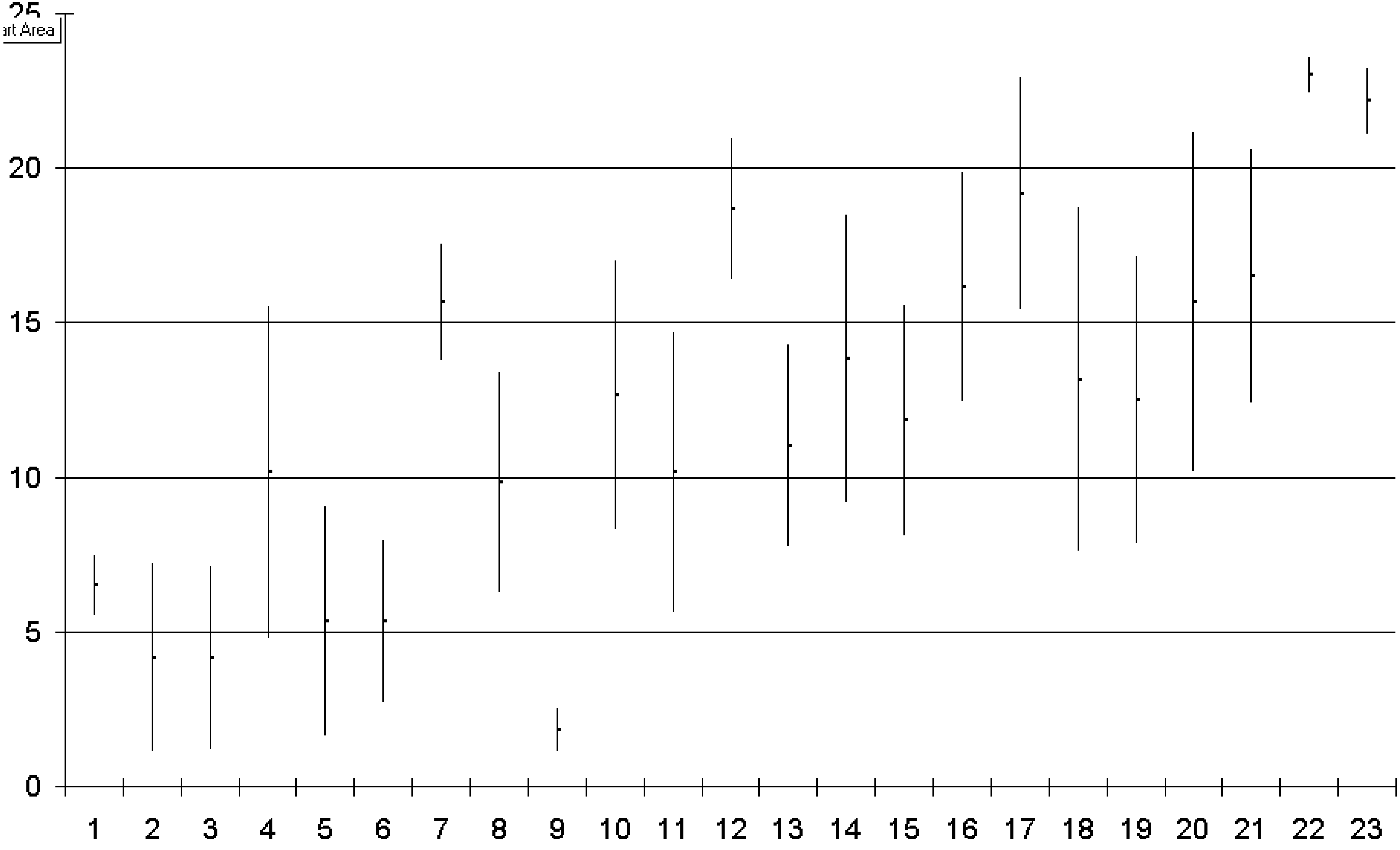
People liked this better than computer: **tinker**

Computer liked this better than people: **dispensing, telesciences, diphtheria, sauté**

Here is the average ranking assigned by six speakers:


25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Plot Area

and here is the same score, with an indication of one standard deviation above and below:



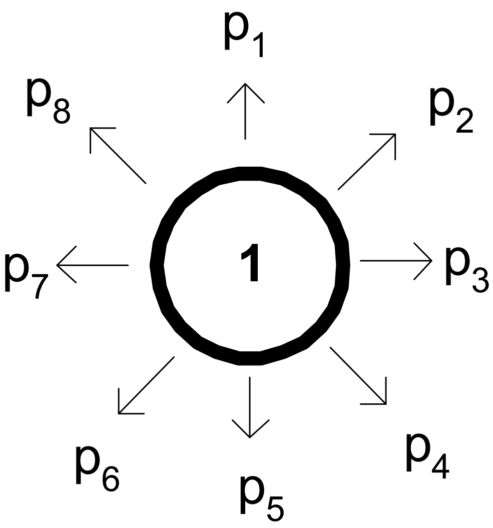
# Part 2: Categories

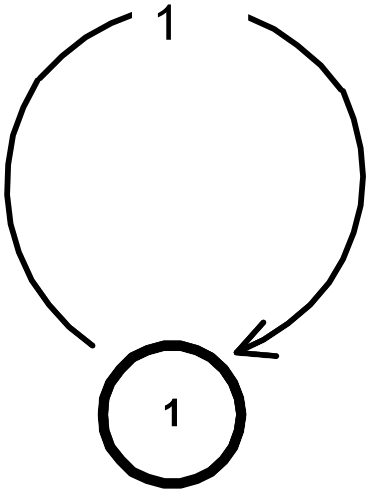
- So far we have made no assumptions about categories.
- Except that there are “phonemes” of some sort in a language, and that they can be counted.
- We have made no assumption about phonemes being sorted into categories.

# Emitting a phoneme

- We will look at models that do *two* things at each moment:
- They *move from state to state*, with a probability assigned to that movement; and
- They emit a symbol, with a probability assigned to emitting each symbol.
- The probability of the entire path is obtained by multiplying together all of the state-to-state transition probabilities, *and* all of the emission probabilities.

# Simplest model for producing the strings of phonemes observed for a corpus (language)



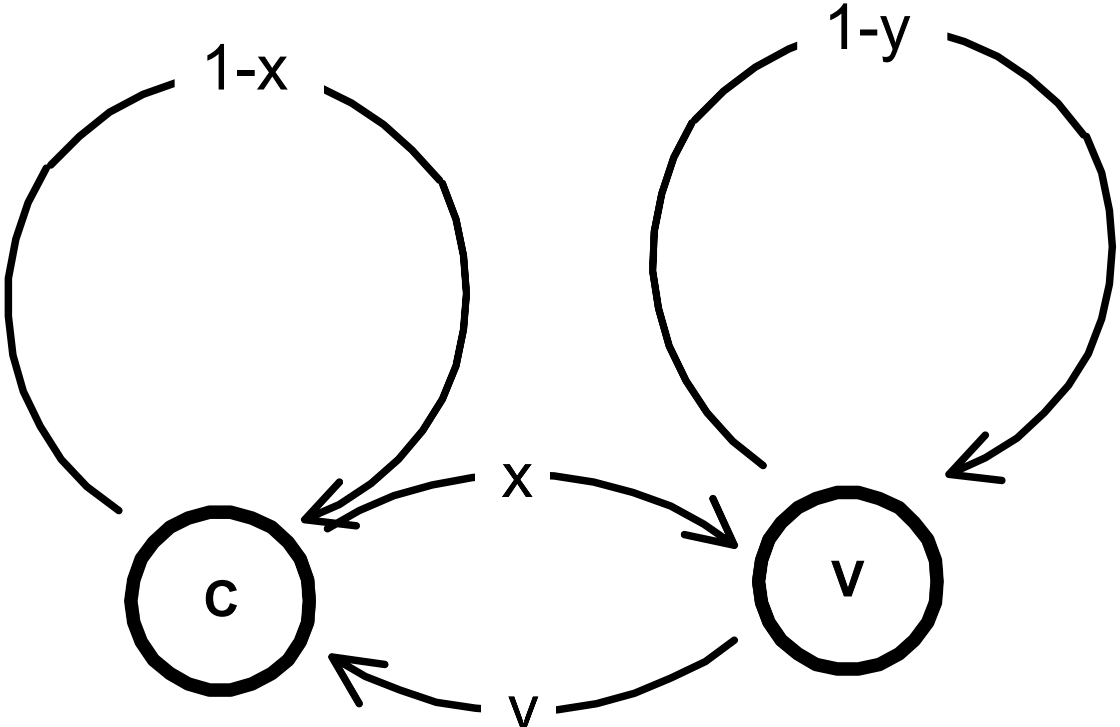


To emit a sequence $p_1p_2$ and stop, there is only one way to do it:

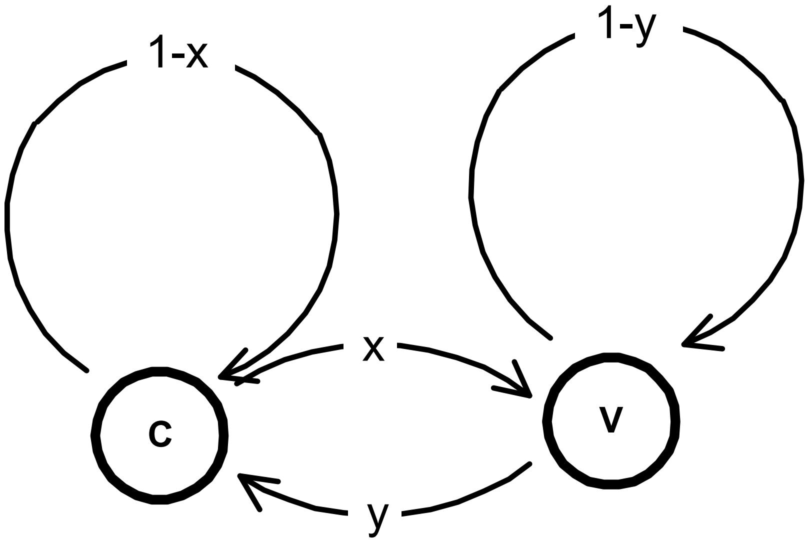
Pass through state 1 *twice*, then stop.

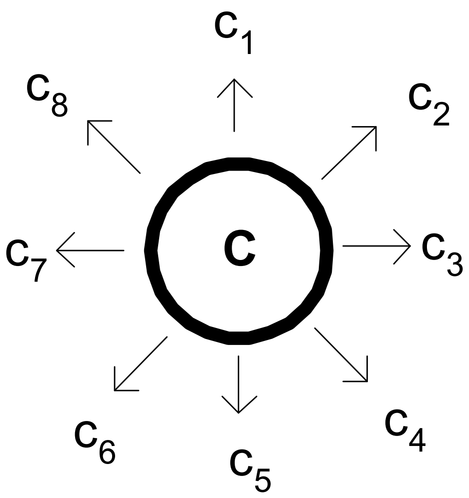
The steps will “cost”: $p_1 * p_2$

# Much more interesting model:

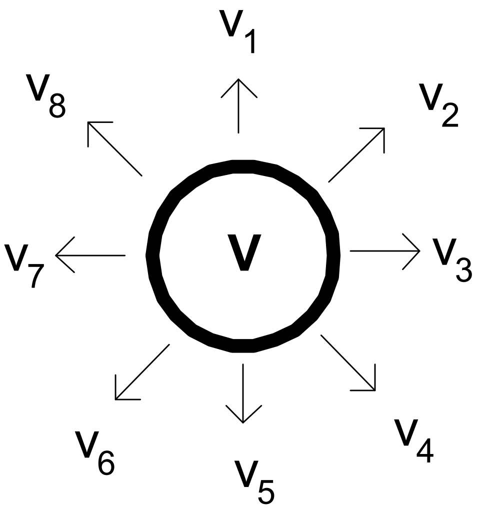


For state transitions; and the same model for emissions: both states emit all of the symbols, but with different probabilities….





$$\sum_i c_i = 1$$



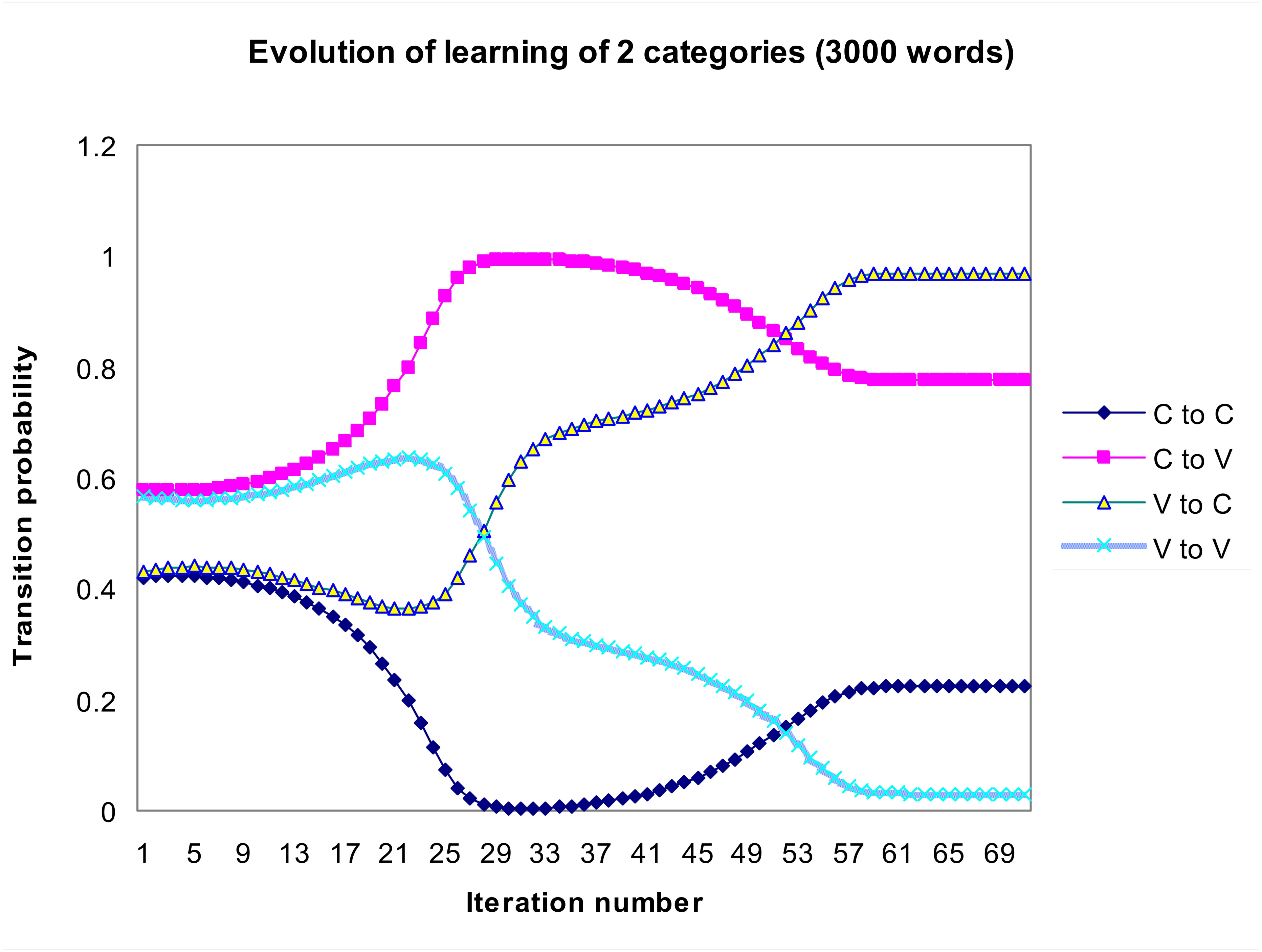
$$\sum_i v_i = 1$$

# The question is…

- How could we obtain the *best* probabilities for *p, q,* and all of the emission probabilities for the two states?
- [Bear in mind: each state generates *all* of the symbols. The only way to ensure that a state does *not* generate a symbol is to assign a zero probability that the emission of the symbol in that state.]

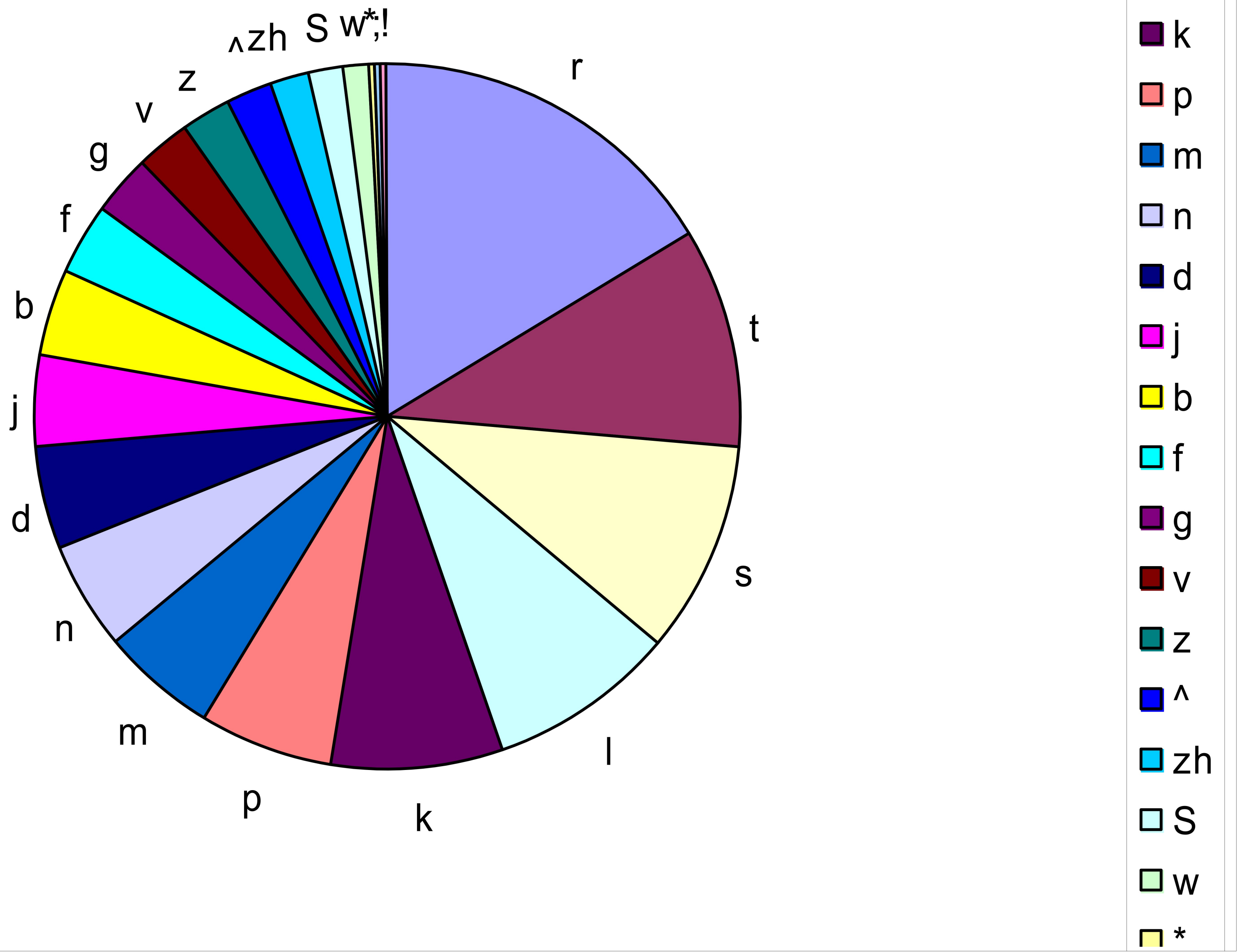
# Results for 2 State HMM

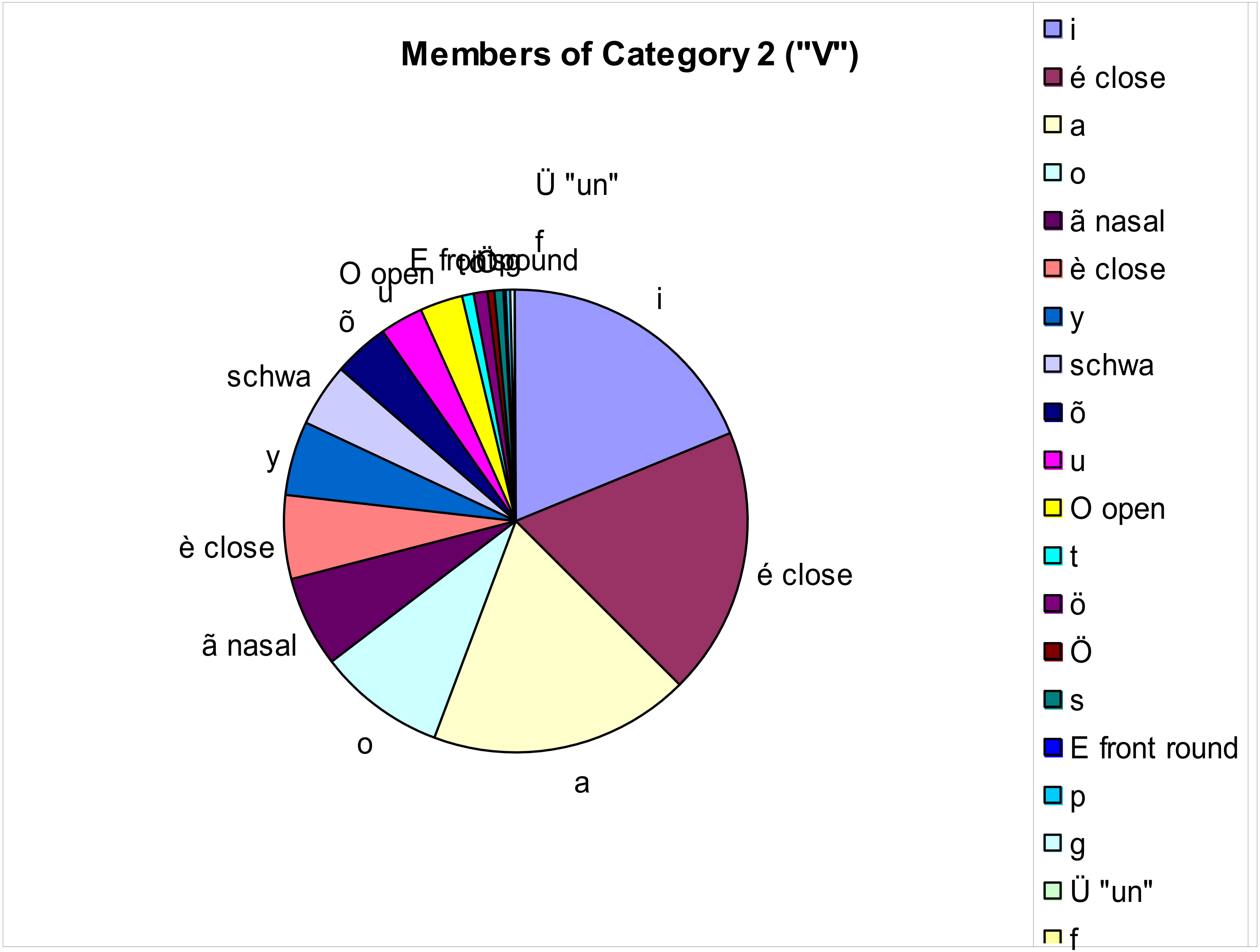
- Separates Cs and Vs



Evolution of learning of 2 categories (3000 words)

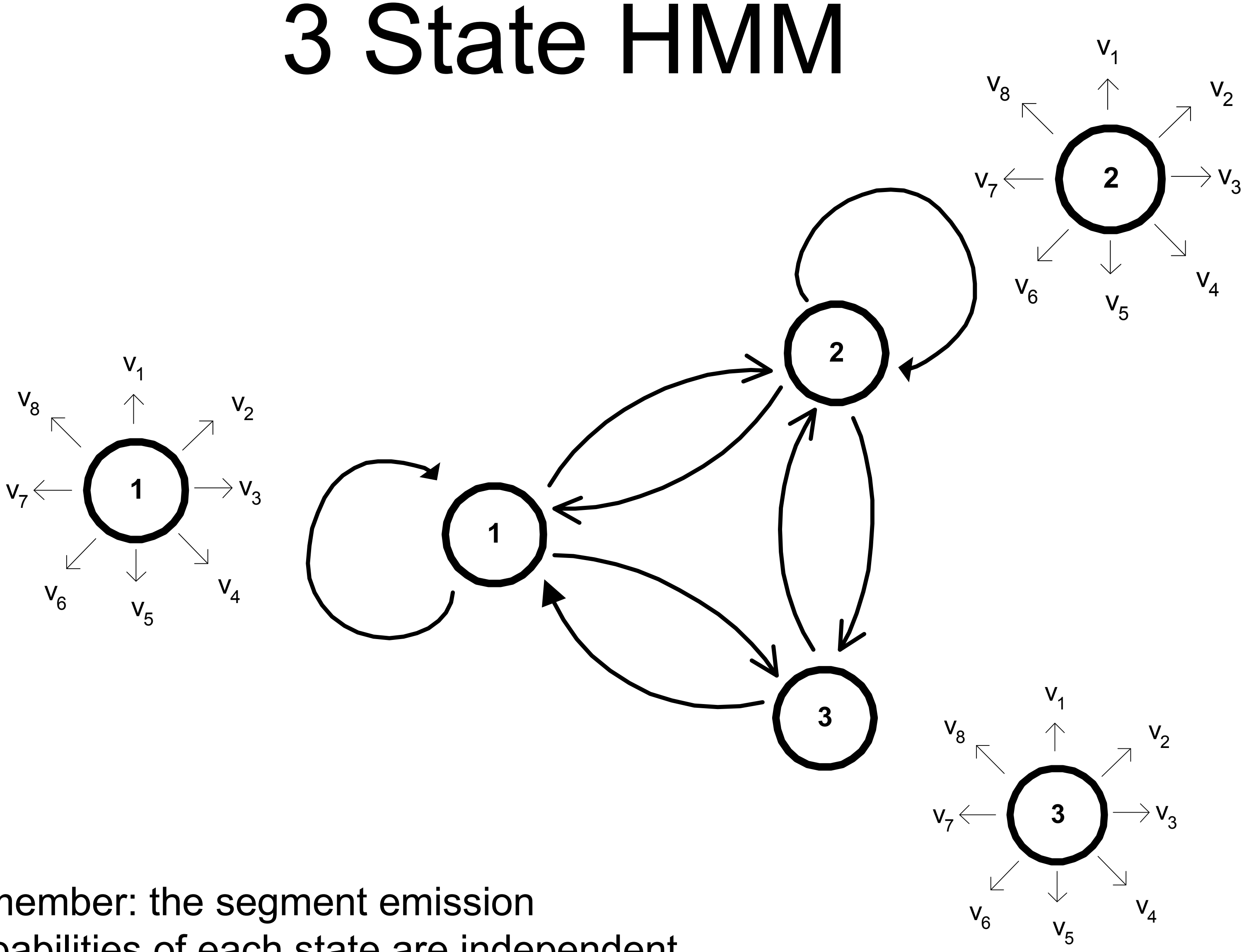
# Members of Category 1 ("C")



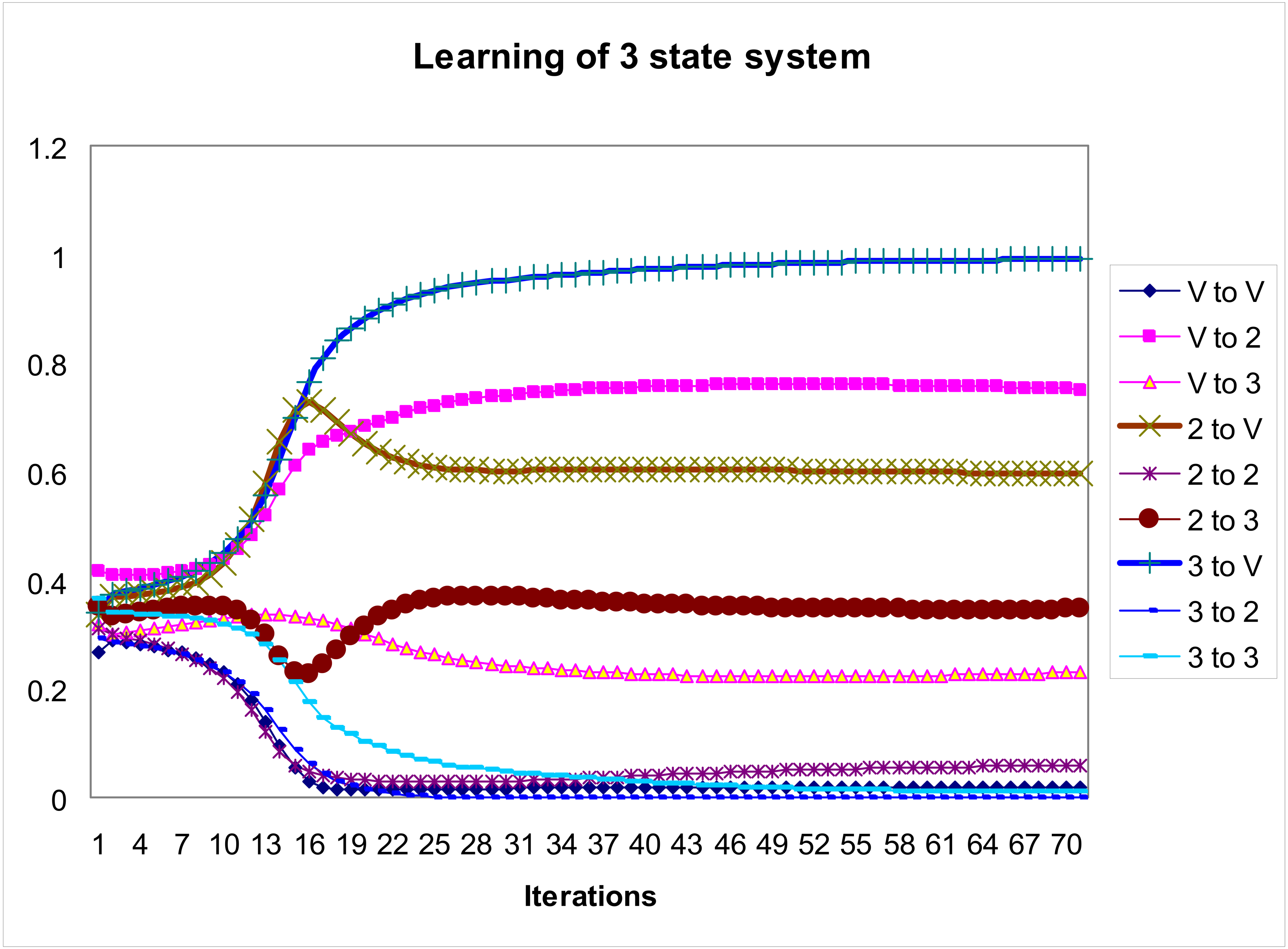


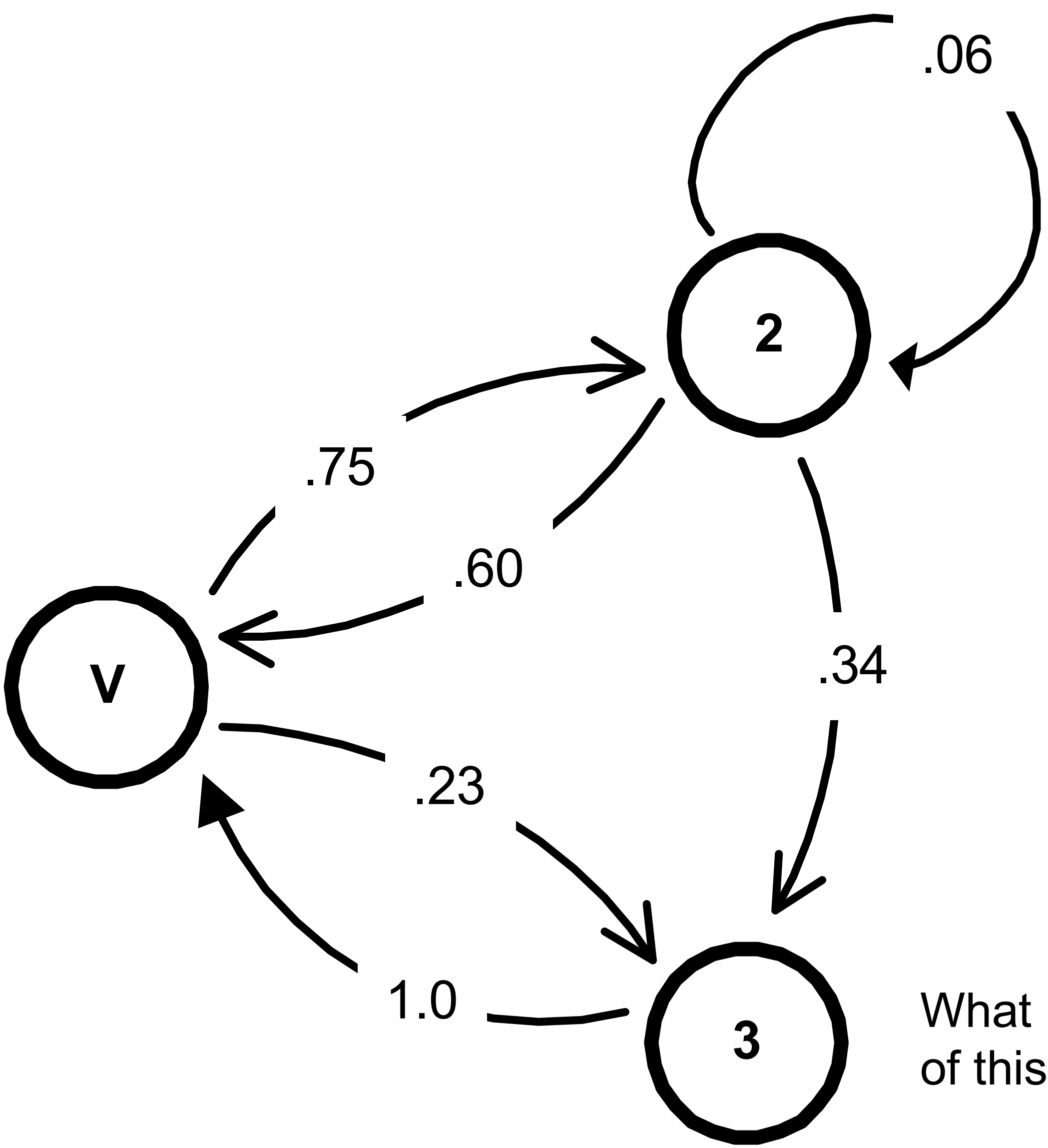
Members of Category 2 ("V")

# 3 State HMM

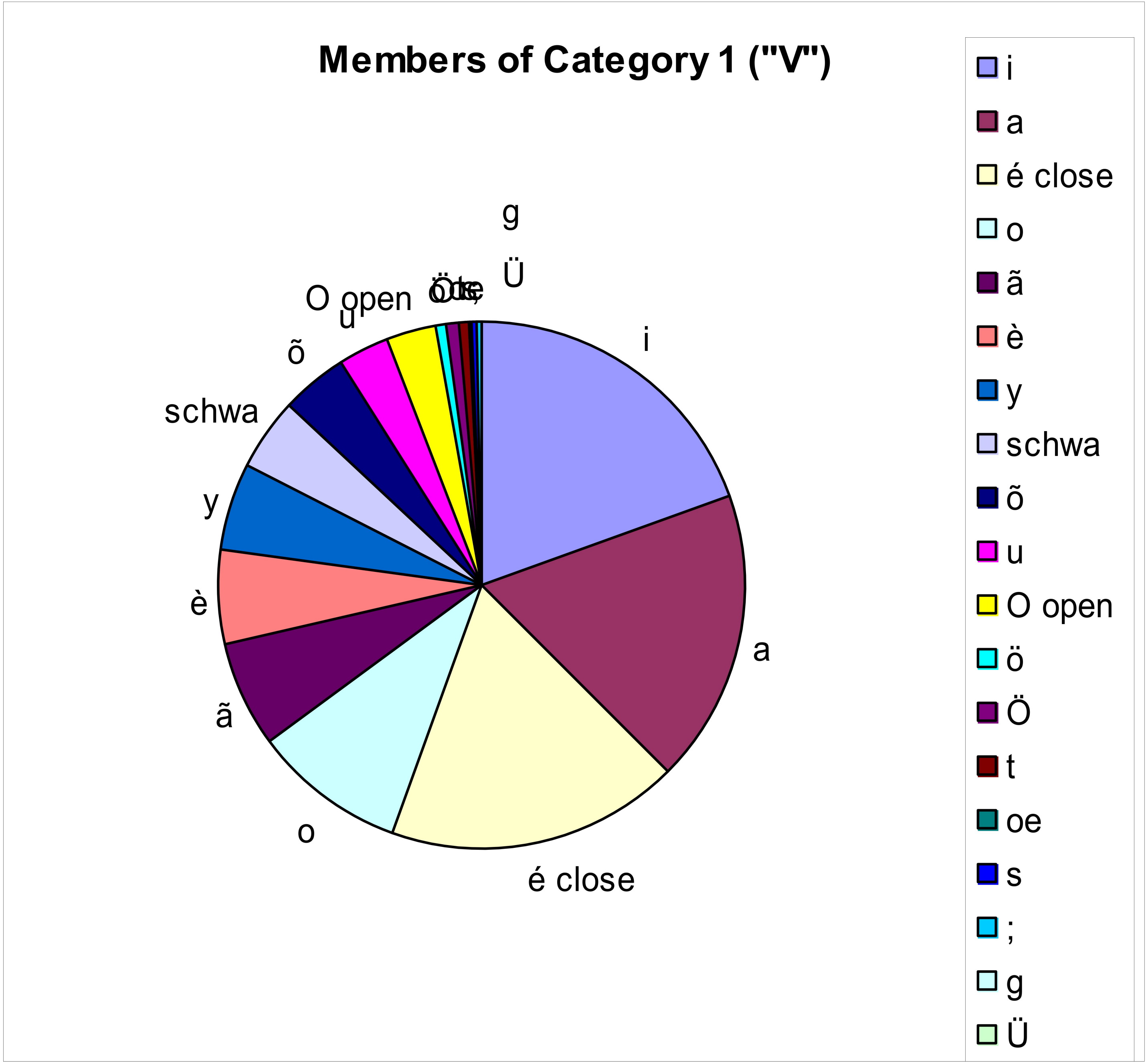


Remember: the segment emission probabilities of each state are independent.


Learning of 3 state system
1.2
1
0.8
0.6
0.4
0.2
0
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70
Iterations
V to V
V to 2
V to 3
2 to V
2 to 2
2 to 3
3 to V
3 to 2
3 to 3

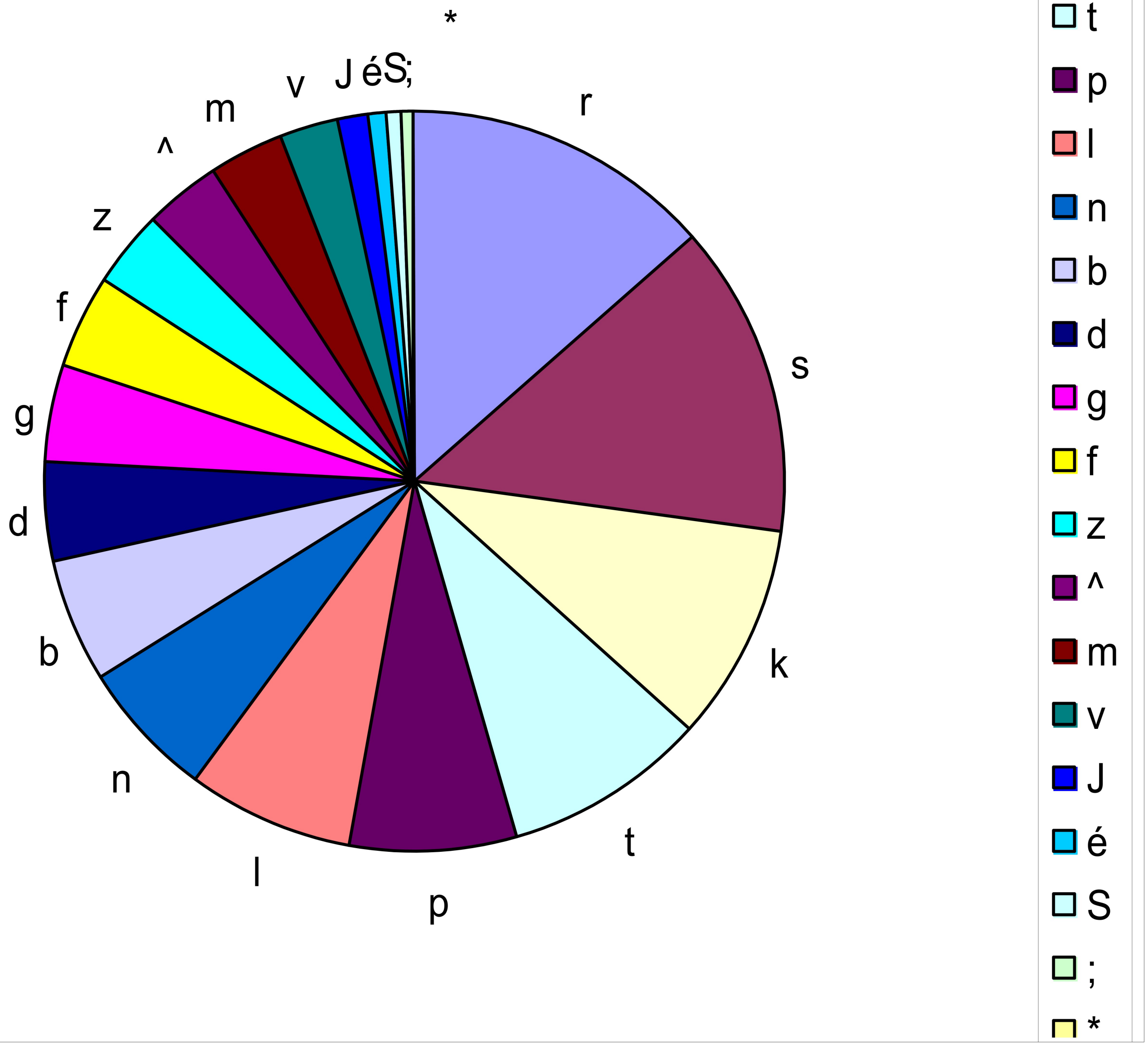


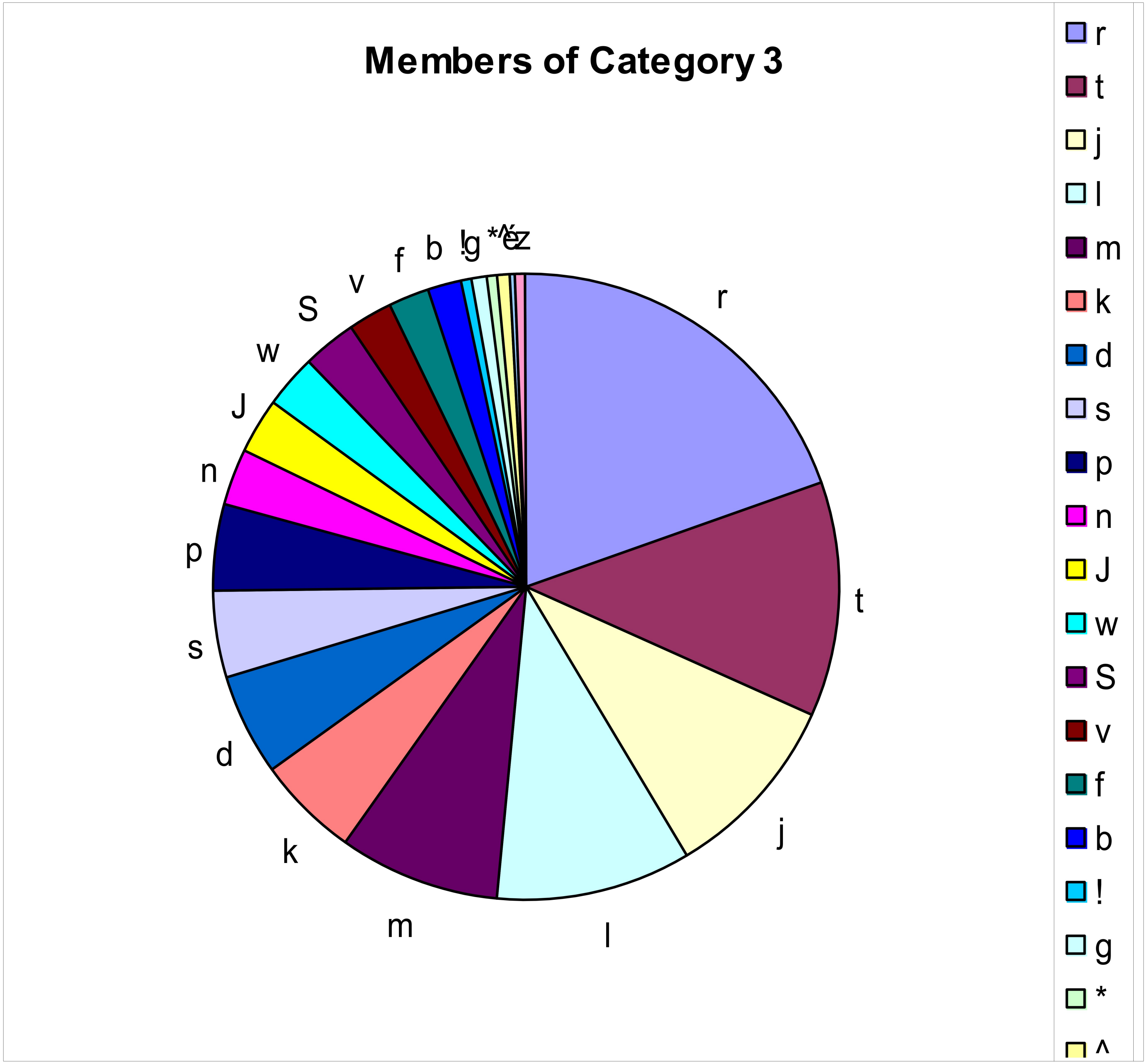
What is the "function" of this state?



Members of Category 1 ("V")

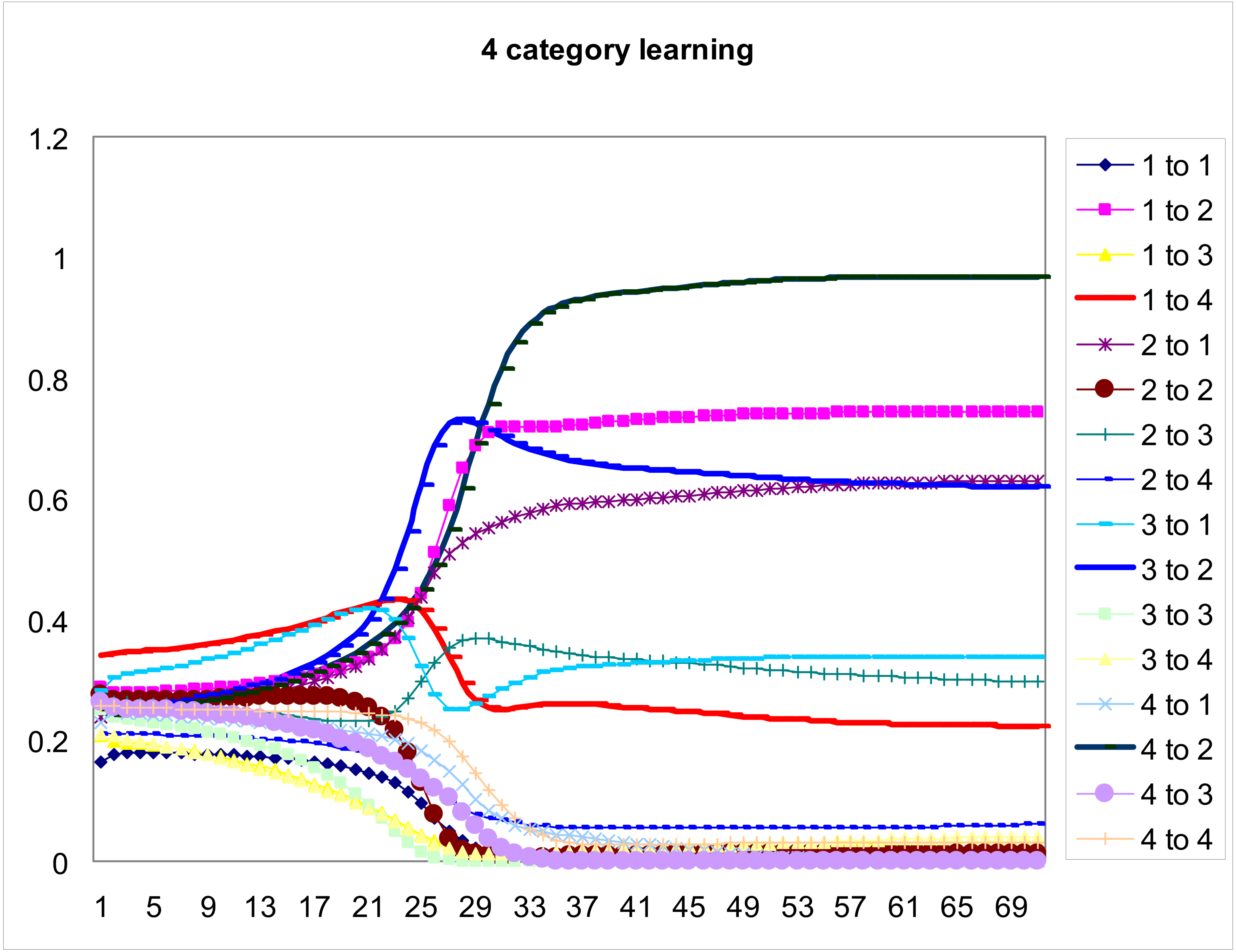
## Members of Category 2

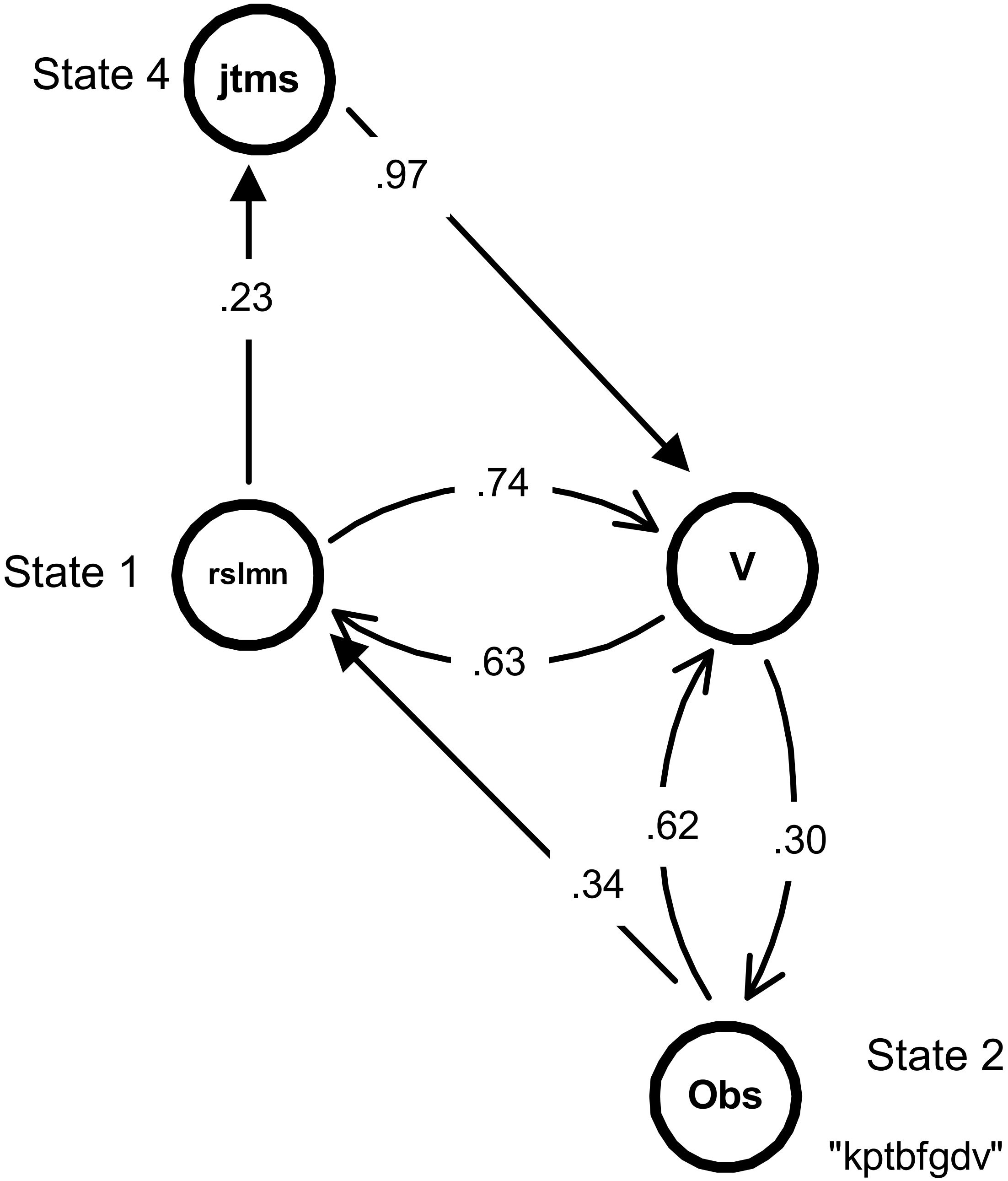


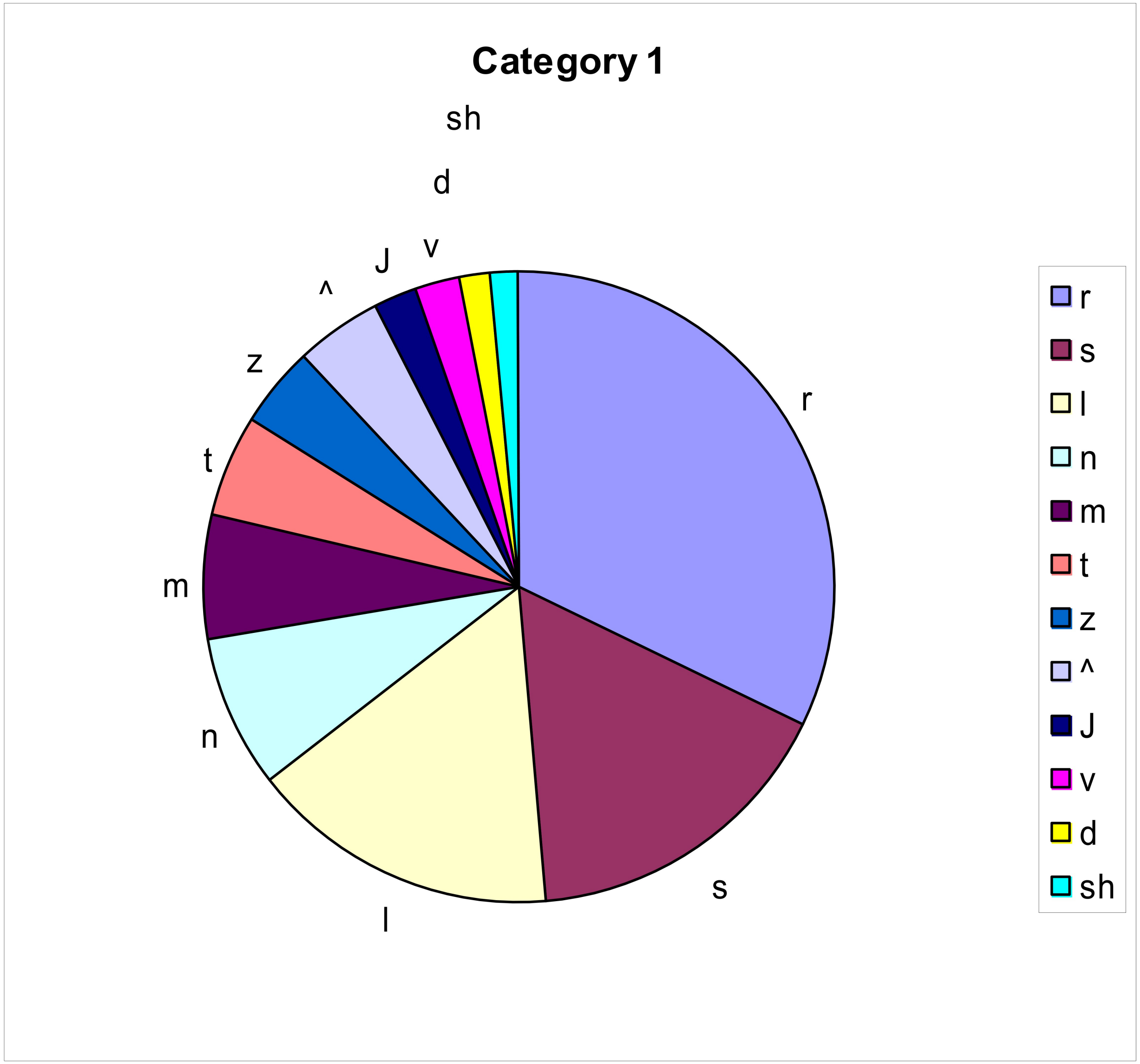


Members of Category 3

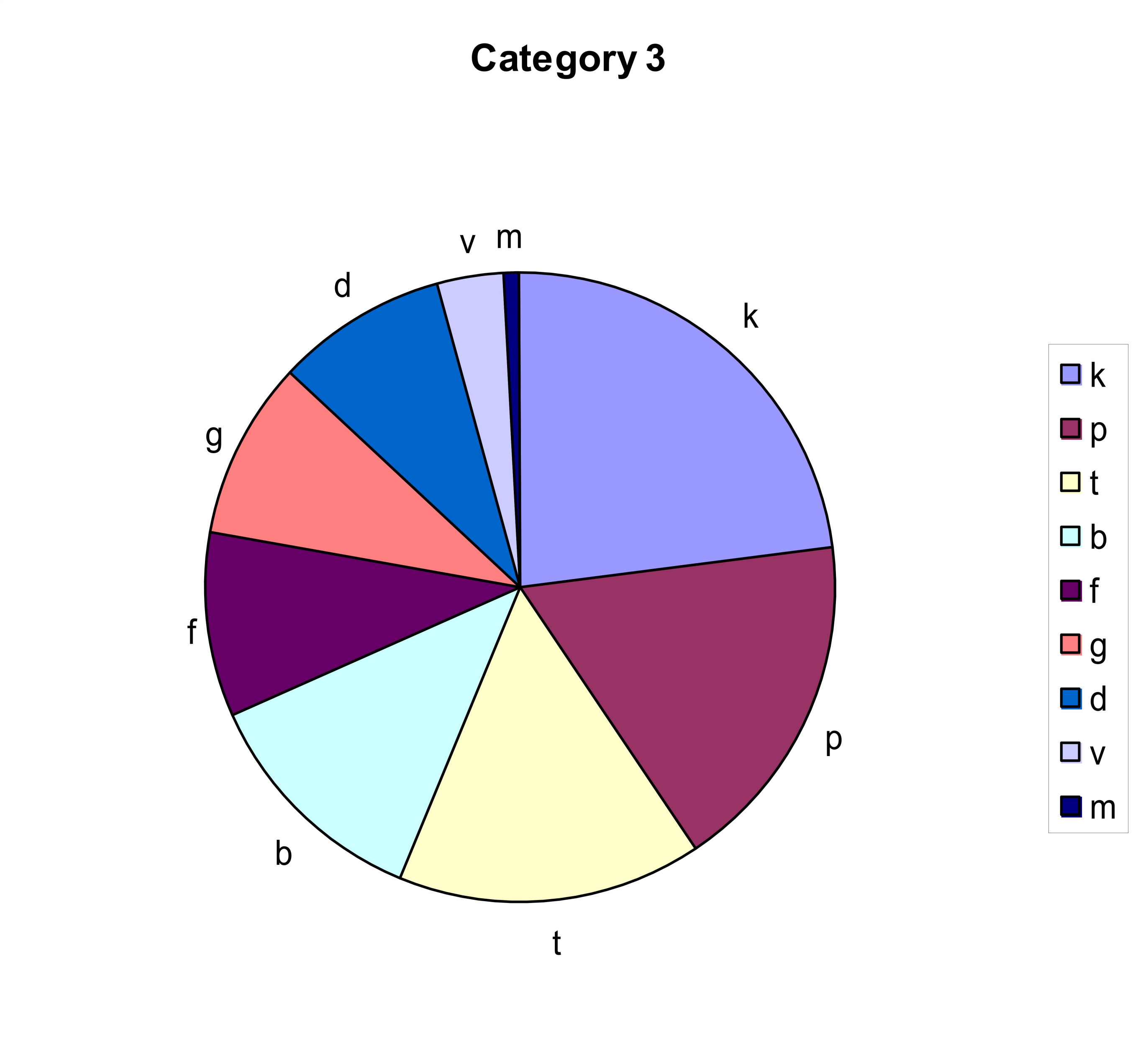
# 4 State HMM learning



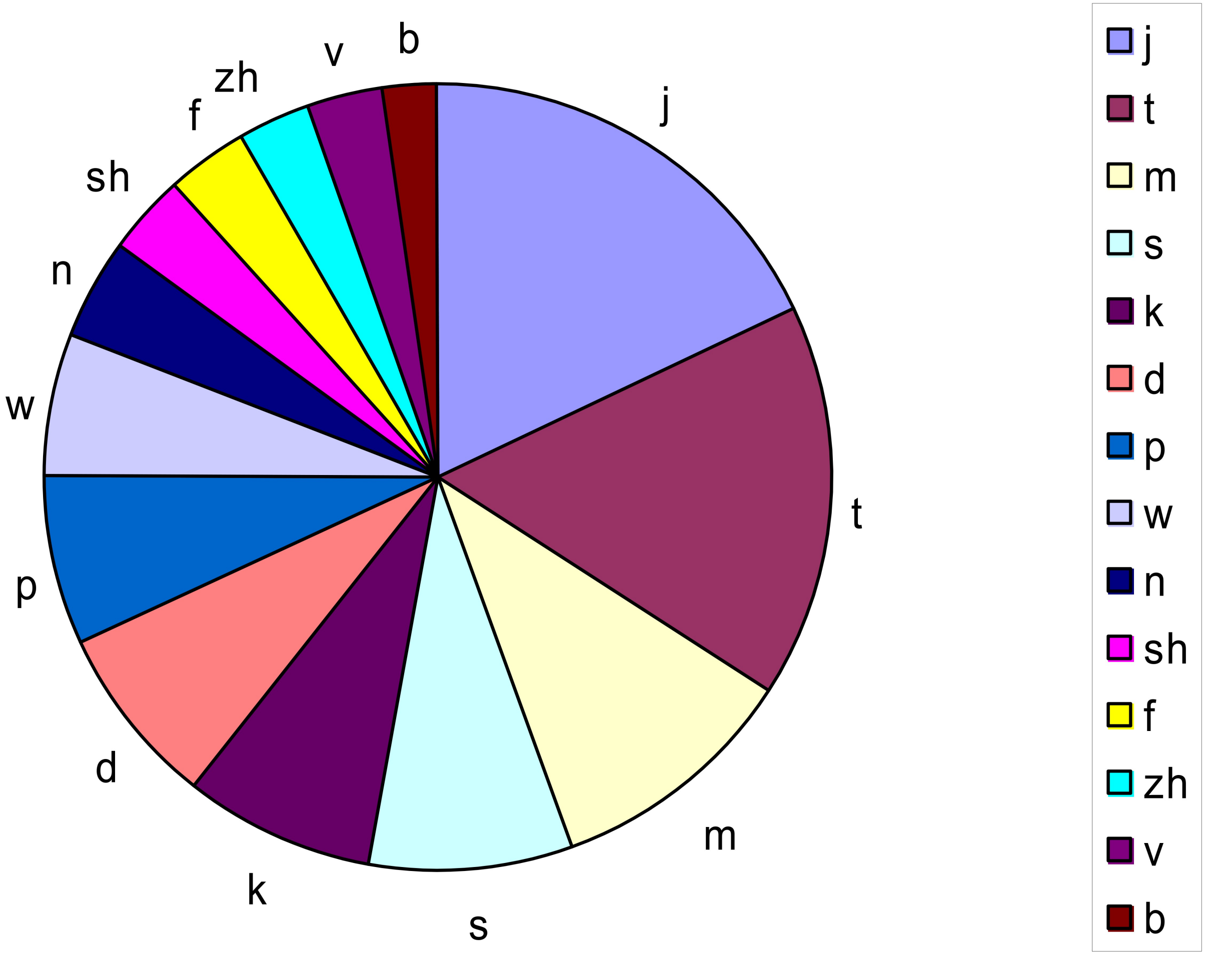

State 4
jtms
.97
.23
.74
State 1
rslmn
V
.63
.62
.30
.34
Obs
State 2
"kptbfgdv"



Category 1


Category 3
v
m
d
k
g
f
p
b
t
k
p
t
b
f
g
d
v
m

## Category 4



# Concluding remarks