Priors in Bayesian Learning of Phonological Rules

Michael Zhong

Three Questions

How do you represent and learn phonological rules?

Where does a MDL-based prior fail in this situation?

How can we do better?

Terminology

Stem: [walk, jump, bark, food, multiply]

Suffixes: [-ing, -ed, -s, -es]

Signatures: [walk, jump, bark], **<€.ed.ing.s>**

Minimum Description Length

Max(Pr(H)Pr(D|H))

Min(-logPr(H) – logPr(D|H)

-logPr(H) => proportional to the length of H

-logPr(D|H) => length of D using H encoding

Linguistica (Goldsmith 2001)

Trade-off:

- Grouping words into signatures makes modeling individual words more difficult
- Assigning words to signatures reduces the number of stems, and thus the length of the grammar

Sample Linguistica Grammar

1. ({work, roll}x{€, ed, ing, er})
2. ({din, bik}x{€, ed, ing, er})
3. ({wait}x{€, ed, er})
4. ({carr}x{y, ied, ier})
5. ({carry}x{€, ing})
6. ({beach, match}x{€, es})

Problem with Linguistica

The (e)ing problem. Beach(+es) and Stomach(+s)

Can only handle Stem-Final Deletion

Not mentioned, but I found this awkward too:

({din, bik}x{e, ed, ing, er})

Morpho-Phonological Grammar (their approach)

Keep signatures, stems, suffixes

Add idea of rule

e.g. e € / CeiC

Representing Rules

$$Xy_ty_fX$$

E.g. jump+ed => CpeC

Why?

Allows insertions, deletions, and substitutions to be handled

Linguistica

1. ({work, roll}x{€, ed, ing, er})
2. ({din, bik}x{€, ed, ing, er})
3. ({wait}x{€, ed, er})
4. ({carr}x{y, ied, ier})
5. ({carry}x{€, ing})
6. ({beach, match}x{€, es})

New approach

1. ({work, roll, dine, carry}x{€, ed, ing, er})
2. ({bike}x{€, ed, ing, er, s})
3. ({wait}x{€, ed, er})
4. ({booth, worker, beach, match}x{€, s})

With 5 rules

Problems with the author's algorithm?

(Where Bayesian Analysis comes into play)

Collapsing signatures lowers corpus likelihood

Stronger explanatory power of signature (prior) often not enough to counterbalance

Prior Used

Trivially, the number of bits used to describe the grammar

Problem?

Poor Incentivizing

{certif, empt, hurr} x {ied, y}

{certify, empty, hurry} x{€, ed}

Tweaking the prior

Assign a fixed cost to each stem

Assign signature cost that varies based on the length of all the suffixes

Final question: What makes this approach better?

What are some weaknesses?

Questions

1. Can the prior be adjusted to include prefixes? If so, why isn't it?
2. Can a similar prior be designed for other languages?
3. To what degree are the modifications arbitrary?