Lesson 6

Conclusion

6.1 Lessons

1. There are explainable and interpretable methods for learning, and for learning formal grammars.
2. Computional Learning Theory is about defining criteria for successful learning and asking what algorithms satisfy that criteria.
    - PAC
    - Identification in the Limit
        (a) from positive data, and complete text
        (b) from positive data, and complete text generable with a primitive recursive function
        (c) from positive and negative data, any complete text
    - Consistent estimators
    - Maximum Likelihood Estimate
    - many others
3. These criteria generally come in two varieties
    (a) quality of output in the long term (asymptotic behavior when data keeps coming)
    (b) quality of output in the short term (some guarantees on any data set)
    (c) CLT is about defining what learning means. This means determining the instance space of the learning problem.
        - What kinds of inputs do you want the learner to succeed on?
        - What kinds of functions do you want the learner to learn?
        - What outputs of the learners count as successful learning?
4. CLT is not meant to replace the empirical approach based on Train/Dev/Test sets. It is mean to accompany it. They are distinct, complementary approaches and both are valuable.
5. In my view, it is unfortunate that most of the world focuses solely on empirical evaluations.
6. The primary result from CLT, under several different criteria, is that **in order to generalize correctly from examples with feasible resources, the concept class (hypothesis space) needs to be appropriately structured (constrained) in some appropriate way**.
    - Examples:
        - For PAC, concept classes need finite VC Dimension
        - For id in the limit from positive data, Angluin's Tell-Tale property must hold.
        - For parametric models, there are certain parameters and not others
    - In other words, No Blank Slate.
    - Valiant calls this computational limits on learning. I call it computational laws of learning.
    - Like any other computational problem, if the instance space is reduced in a meaningful way, problems can become easier to solve.
        - Reducing the class of inputs needed to succeed helps (e.g. from any text to one generated by primitive recursive functions)
        - Reducing the class of functions needed helps (e.g. from superfinite to SL, zero-reversible, or substitutable)
    - Reducing the class of inputs has been shown to make learning any computably enumerable function possible *in principle* but not feasibly.
        - Gold (1967) can learn any function from positive data generated by primitive

recursive functions via enumeration
    - Chater and Vitányi (2007) can learn any function from positive date generated by monotone Turing Machines via the uncomputable universal distribution in Kolmogorov information theory
- Reducing the class of functions to be learned has been shown to make learning those functions feasibly without further constraints on the inputs
    - See the entire field of grammatical inference (de la Higuera, 2010; Heinz *et al.*, 2015; Heinz and Sempere, 2016; Wieczorek, 2017)
    - Even Clark and Lappin (2011) acknowledge this point.

6.2 Open Questions

1. Finding a way to measure distances between formal languages to more easily apply PAC to grammar learning
2. String to string functions
    (a) with optional outputs
    (b) with linguistic represenations like phonological features, autosegmental representations, ...
    (c) using grammars based in logic as opposed to automata
3. Learning guarantees in the presence of noise
4. Use of Tolerance Principle
5. Language modeling (predicting likelihood of next item in a sequence) with "designer" hypothesis spaces
6. All of the above for Trees
7. Language change and evolution in the light of learning
8. Tasks
    (a) g2p, p2g, transliteration
    (b) machine translation with tree transductions (See earlier work by Kevin Knight)
    (c) In morphology, Unimorph inflectional challenge (Kodner and Khalifa, 2022)
    (d) In morpho-phonology, automatic lexicon and grammar induction from linguistic data set (Ellis *et al.*, 2022)
    (e) In morpho-phonology, automatic lexicon and grammar induction from child acquisition data

Bibliography

Chater, Nick, and Paul Vitányi. 2007. 'Ideal learning' of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology* 51:135–163.

Clark, Alexander, and Shalom Lappin. 2011. *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

Ellis, Kevin, Adam Albright, Armando Solar-Lezama, Joshua B. Tenenbaum, and Timothy J. O'Donnell. 2022. Synthesizing theories of human language with bayesian program induction. *Nature Communications* 13. Https://doi.org/10.1038/s41467-022-32012-w.

Gold, E.M. 1967. Language identification in the limit. *Information and Control* 10:447–474.

Heinz, Jeffrey, Colin de la Higuera, and Menno van Zaanen. 2015. *Grammatical Inference for Computational Linguistics*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Heinz, Jeffrey, and José Sempere, eds. 2016. *Topics in Grammatical Inference*. Springer-Verlag Berlin Heidelberg.

de la Higuera, Colin. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Kodner, Jordan, and Salam Khalifa. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Modeling inflection in language acquisition. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 157–175. Seattle, Washington: Association for Computational Linguistics.
URL https://aclanthology.org/2022.sigmorphon-1.18

Wieczorek, Wojciech. 2017. *Grammatical Inference: Algorithms, Routines and Applications*. Springer.