Grammatical Inference:
formal and heuristic methods

May 1997

Pierre Dupont
pdupont@cs.cmu.edu

Informal Definition

What *grammatical inference/grammar induction* is about:

- Induce structural patterns from data
- Represent them by a formal grammar

What *grammatical inference* is not (yet) about:

$\Rightarrow$ Automatic discovery of a complete natural language theory

- GI uses simple formalisms
- GI is applied to limited domains
- No a priori knowledge is assumed (in most cases)
- GI can be applied to any kind of "strings", not only natural language sentences

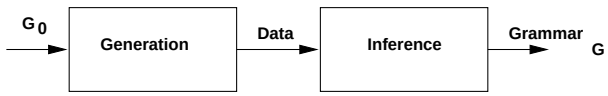
Carnegie Mellon

Formal definition

Grammatical Inference is a particular instance of *Inductive Learning* which can be formulated as the task of discovering common structures in examples which are supposed to be generated by the same process. In this case, the set of examples, also called *positive sample*, is usually a set of strings defined on a specific *alphabet*. A *negative sample*, that is a set of strings not belonging to the target language, may also help the induction process.

Carnegie Mellon

The fundamental hypothesis

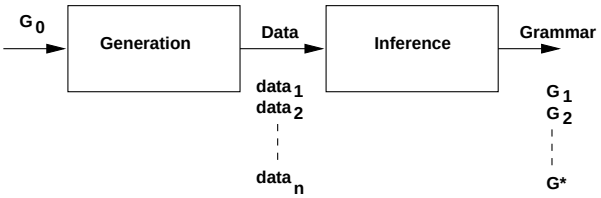


- Find G as close as possible to G_0
- Nobody claims the existence of G_0. It is simply a modeling hypothesis

Carnegie Mellon

How to characterize learning

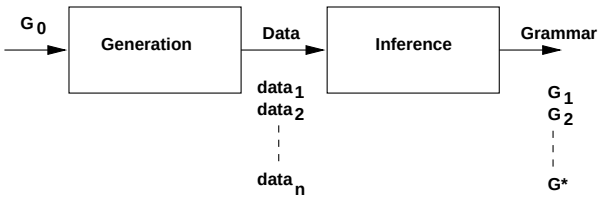
Identification in the limit [12]



- convergence to G^*
- G^* is one representation of $L(G_0)$

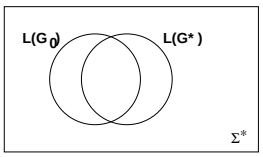
PAC [23, 20] (Probably approximately correct)

Note: a slightly simplified view !



- convergence to G^*
- G^* is close enough to G_0 with high probability

Define a probability distribution D on the strings



$$P[P_D(L(G^*) \oplus L(G_0)) < \epsilon] > 1 - \delta$$

The result must hold for any distribution D.

Bad and good news

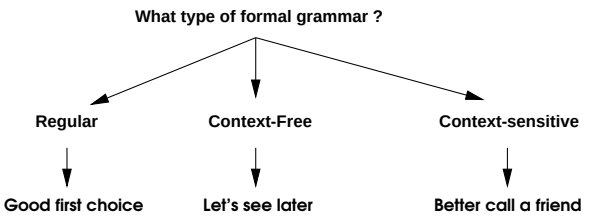
The bad one...

Theorem 1 No *superfinite* class of languages is identifiable in the limit from positive data only [12]

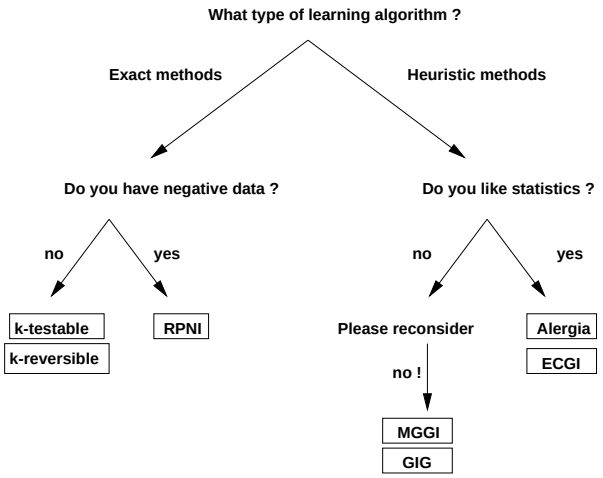
The good one...

Theorem 2 Any *admissible* class of languages is identifiable in the limit from positive *AND* negative data [12]

Choosing an inference method



Regular grammars [9, 2, 5, 17, 6, 4, 22, 10, 8]

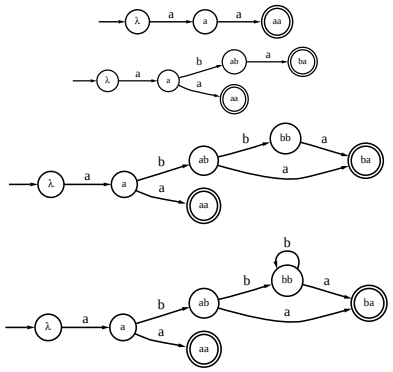


The k-TSSI algorithm

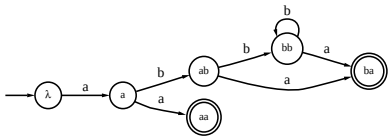
Inference of k-testable languages in the strict sense [9]...

- Choose a value of k
- Build an automaton such that all strings sharing the k-1 last symbols point to the same state

Example: $S_+ = \{aa, aba, abba, abbba\}$

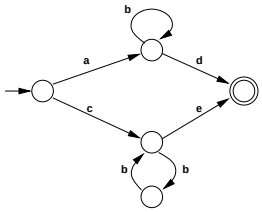


...a funny name for Ngrams



$$\hat{p}(a|bb) = \frac{C(bba)}{C(bb)}$$

Not all regular languages are k-testable

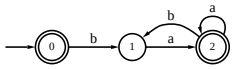


$\Rightarrow$ One can potentially do better than Ngrams

Regular Inference from Positive and Negative Data

Property 1 Any *regular language* has a *canonical automaton* $A(L)$ which is *deterministic* and *minimal*

Example: $L = (ba^*a)^*$

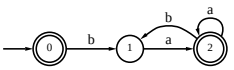


A few definitions

Definition 1 A positive sample S_+ is *structurally complete* [7] with respect to an automaton A if when generating S_+ from A

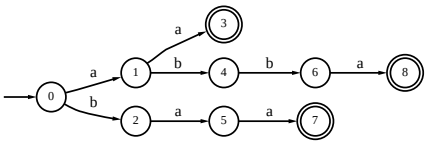
- every transition of A is used at least once
- every accepting state is used to accept at least one string

Example: $\{ba, baa, baba, \lambda\}$



Definition 2 The *prefix tree acceptor (PTA)* of a sample S_+

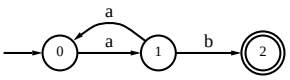
Example: $\{aa, baa, abba\}$

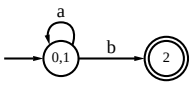


Merging is fun...

Definition 3 A *quotient-automaton* A_π of an automaton A

- Define a partition π on the state set of A
- Merge all the states belonging to the same block in the partition π

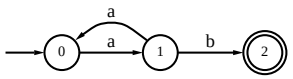


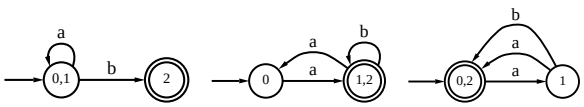


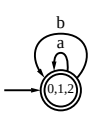
$$\pi = \{\{0,1\},\{2\}\} \Rightarrow L(A) \subseteq L(A_\pi)$$

Merging states $\Leftrightarrow$ generalize language

A lattice of automata





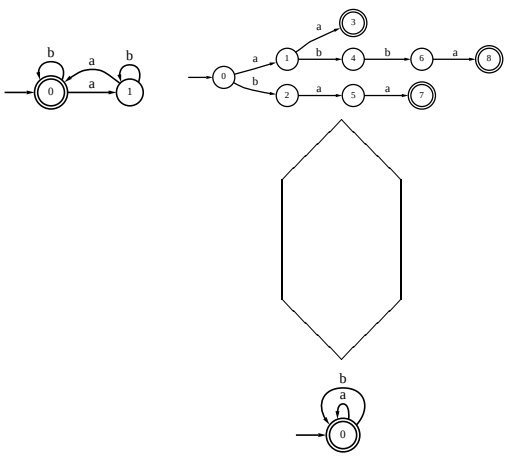


Let's summarize

- we observe some data
- we assume the data is a positive sample S_+ from some unknown regular language L_0
- we assume the sample is structurally complete with respect to the canonical automaton $A(L_0)$ of L_0
- we build the Prefix Tree Acceptor of S_+
- we know that $L(PTA) = S_+$
- we can generalize S_+ by merging states of the PTA

This is the last theorem...

Theorem 3 *There exists a partition π of the state set of PTA such that $PTA/\pi = A(L_0)$ [2, 7]*

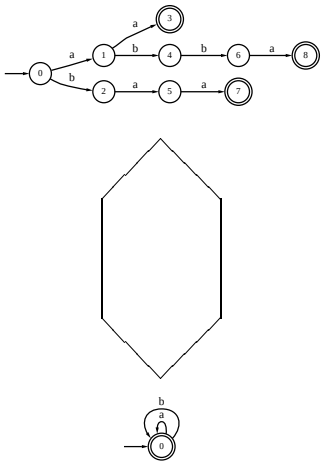


Infer a regular language $\Leftrightarrow$ find the partition π.

How are we going to choose the right partition ? Remember the good news.

The RPNI algorithm [17, 6]

- Follow a path down into the lattice by merging states in a specified order
- Use *negative data* to select the optimal partition



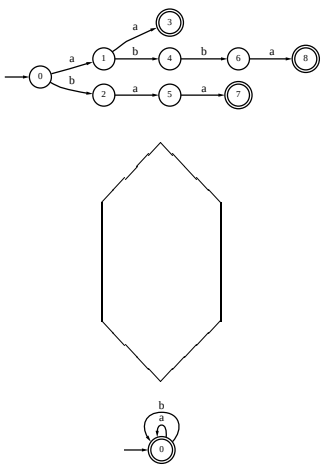
RPNI properties

Believe it or not...

- RPNI is another instance of the "generalization as search" paradigm
- RPNI time complexity is a polynomial function of the sample (S_+, S_-) size
- RPNI is guaranteed to identify in the limit any regular language
- The size of *a characteristic sample*, i.e. a sample such that RPNI is guaranteed to converge to the exact solution, is a quadratic function of the size of $A(L_0)$

The ALERGIA algorithm [4]

- Follow a path down into the lattice by merging states in a specified order.
- Use *statistical information* to select the optimal partition

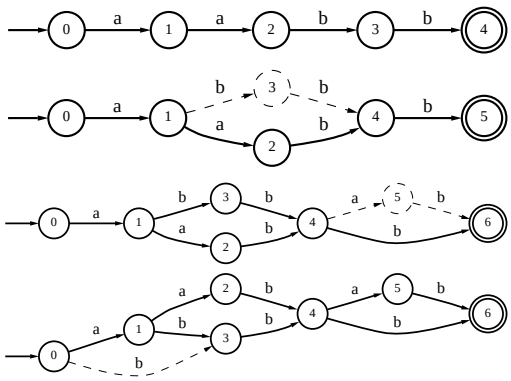


The ECGI algorithm [22, 21]

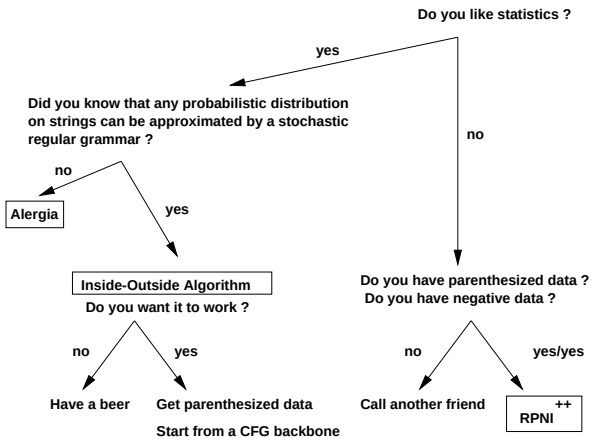
Another heuristic method:

- hypothesis: several training samples share common substrings
- method: error corrective parsing

Example: $\{aabb, abbb, abbab, bbb\}$



Context-free grammars [14, 19, 24]



I did not talk about...

- all inference methods
- NP-completeness of minimal DFA identification and other complexity issues [13, 1, 20, 7]
- other heuristic methods, Neural Nets [11], Genetic Algorithms [5]
- how to use a priori knowledge [8]
- how to reduce the alphabet size by statistical clustering [16, 3]
- how to parse natural languages without a grammar (decision trees) [15]
- how to smooth stochastic grammars
- how to extend the learning of automata to transducers [18]

A short list of references

References

[1] D. Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39:337–350, 1978.

[2] D. Angluin. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29(3):741–765, 1982.

[3] P. Brown, V. Della Pietra, P. de Souza, J. Lai, and R. Mercer. Class-based N-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

[4] R. Carrasco and J. Oncina. Learning stochastic regular grammars by means of a state merging method. In *Grammatical Inference and Applications, ICGI'94*, number 862 in Lecture Notes in Artificial Intelligence, pages 139–150. Springer Verlag, 1994.

[5] P. Dupont. Regular grammatical inference from positive and negative samples by genetic search : the GIG method. In *Grammatical Inference and Applications, ICGI'94*, number 862 in Lecture Notes in Artificial Intelligence, pages 236–245. Springer Verlag, 1994.

[6] P. Dupont. Incremental regular inference. In *Grammatical Inference : Learning Syntax from Sentences, ICGI'96*, number 1147 in Lecture Notes in Artificial Intelligence, pages 222–237. Springer Verlag, 1996.

[7] P. Dupont, L. Miclet, and E. Vidal. What is the search space of the regular inference ? In *Grammatical Inference and Applications, ICGI'94*, number 862 in Lecture Notes in Artificial Intelligence, pages 25–37. Springer Verlag, 1994.

[8] P. García, E. Segarra, E. Vidal, and I. Galiano. On the use of the Morphic Generator Grammatical Inference (mggi) methodology in automatic speech recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 4(4):667–685, 1990.

[9] P. García and E. Vidal. Inference of K-testable languages in the strict sense and applications to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):920–925, 1990.

[10] P. García, E. Vidal, and F. Casacuberta. Local languages, the Successor method, and a step towards a general methodology for the inference of regular grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):841–845, 1987.

[11] C.L. Giles, G.Z. Sun, H.H. Chen, Y.C. Lee, and D. Chen. High order recurrent networks and grammatical inference. In D.S. Touretzky, editor, *Advances in Neural Information Processing System*, pages 380–387. Morgan Kaufman, San Mateo, 1990.

[12] E.M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[13] E.M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37:302–320, 1978.

[14] K. Lari and S.J. Young. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

[15] D.M. Magerman. Learning grammatical structure using statistical decision trees. In *Grammatical Inference : Learning Syntax from Sentences, ICGI'96*, number 1147 in Lecture Notes in Artificial Intelligence, pages 1–21. Springer Verlag, 1996.

[16] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

[17] J. Oncina and P. García. Inferring regular languages in polynomial update time. In N. Pérez de la Blanca, A. Sanfeliu, and E.Vidal, editors, *Pattern Recognition and Image Analysis*, volume 1 of *Series in Machine Perception and Artificial Intelligence*, pages 49–61. World Scientific, 1992.

[18] J. Oncina, P. Garcia, and E. Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458, 1993.

[19] F. Pereira and Y. Schabes. Inside-outside reestimation from partially bracketed corpora. In *30th Annual Meeting of the ACL*, pages 128–135, 1992.

[20] L. Pitt. Inductive inference, dfa's, and computational complexity. In K.P. Jantke, editor, *Analogical and Inductive Inference*, number 397 in Lecture Notes in Artificial Intelligence, pages 18–44. Springer-Verlag, Berlin, 1989.

[21] H. Rulot. *ECGI : Un Algoritmo de Inferencia Gramatical Mediante Corrección de Errores*. Ph. D. dissertation, Universität de València, 1992.

[22] H. Rulot and E. Vidal. An efficient algorithm for the inference of circuit-free automata. In G. Ferratè, T. Pavlidis, A. Sanfeliu, and H. Bunke, editors, *Advances in Structural and Syntactic Pattern Recognition*, pages 173–184. NATO ASI, Springer-Verlag, 1988.

[23] L.G. Valiant. A theory of the learnable. *Communications of the Association for Computing Machinery*, 27(11):1134–1142, 1984.

[24] S.J. Young and H.-H. Shih. Computer assisted grammar construction. In *Grammatical Inference and Applications, ICGI'94*, number 862 in Lecture Notes in Artificial Intelligence, pages 282–290. Springer Verlag, 1994.