Organisation
des Nations Unies
pour l'éducation,
la science et la culture

Chaire UNESCO en technologies
pour la formation des enseignants
par les ressources éducatives libres,
Université de Nantes, France

UNIVERSITÉ DE NANTES

LABORATOIRE DES SCIENCES DU NUMÉRIQUE DE NANTES

Grammatical inference: where did all those good ideas go?

Colin de la Higuera

Outline

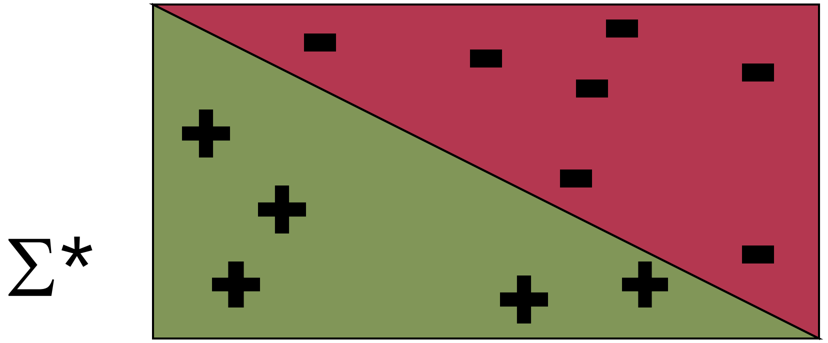
1. What is Grammatical Inference about?
2. A dateline (of the prehistory of GI)
3. Some keywords
    1. Identification
    2. Complexity
    3. Simplicity
    4. Approximation
    5. Interaction

1. What is grammatical inference?

1.0 The rules of the game

Motivation

- We are given a set of strings S_+ and a set of strings S_-
- Goal is to build a classifier
- This is a traditional (or typical) machine learning question
- How should we solve it?

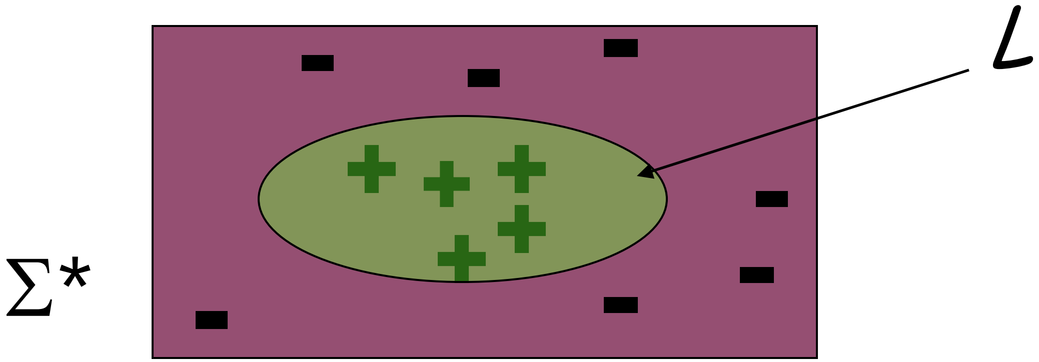


Ideas

- Use a distance between strings and try *k*-NN (nearest neighbours)
- Embed strings into vectors and use some off-the-shelf technique (decision trees, SVMs, other kernel methods)

Alternative

- Suppose the classifier is some grammatical formalism
- Thus we have *L* and Σ*\*L*



Some alternative definitions

- **Grammar induction** (or **grammatical inference**[1]) is the process in machine learning of learning a formal grammar (usually as a collection of *re-write rules* or *productions* or alternatively as a finite state machine or automaton of some kind) from a set of observations, thus constructing a model which accounts for the characteristics of the observed objects. More generally, grammatical inference is that branch of machine learning where the instance space consists of discrete combinatorial objects such as strings, trees and graphs. *[Wikipedia]*
- The problem of grammatical inference is, in its broadest sense, the problem of learning a description of a language from data drawn from (but not necessarily in) the language. *[Lee, Lillian. 1996. Learning of Context-Free Languages: A Survey of the Literature. Harvard Computer Science Group Technical Report TR-12-96.]*
- Grammatical inference is about learning a grammar given information about a language *[cdlh, 2010]*

Inference or induction? (Feldman & al. 1969)

Early studies of grammatical inference referred to it as a form of induction. The term "induction“ has been used as a description of generalization processes. Unfortunately, it has also been used in dozens of other ways and is threatening to become meaningless.

We favor restricting the term "induction“ to statistical modes of inference such as those of Solomonoff [64] as is done currently in Philosophy.

The particular model which we found most appropriate is the hypothetico-deductive- empirical (HDE) mode of inference. An HDE inference consists of forming hypotheses, deducing conclusions about the data and testing these conclusions for validity.

[…]

In our case, a hypothesis is a grammar rule, a deduction is a derivation, and the data are the sample strings.

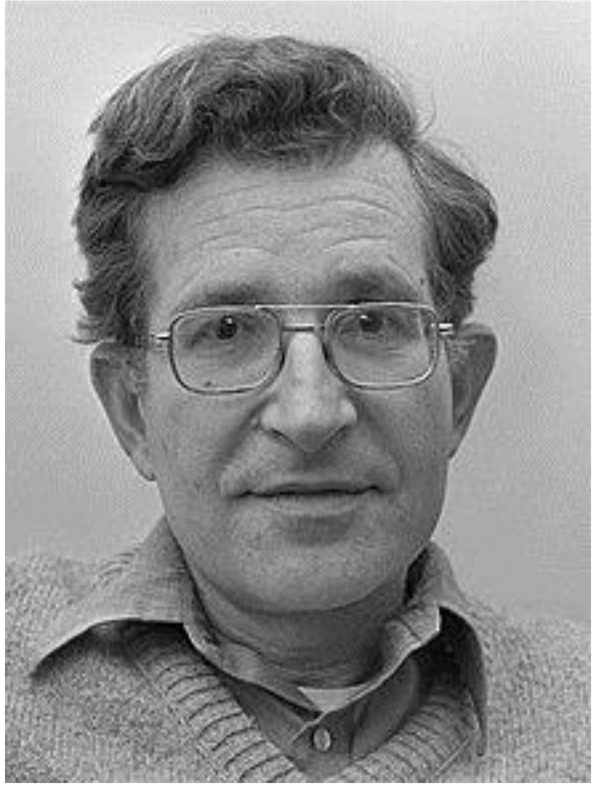
2. A timeline and some names

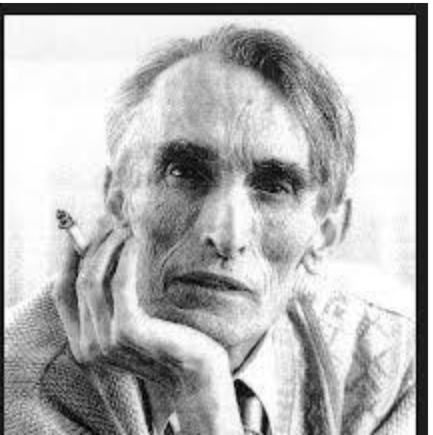
Where (how?) did things start?

- 1955 Chomsky
- 1959 Solomonoff
- 1965 Gold
- 1967-69 Feldman and Horning
- 1972 Fu
- 1980 Miclet
- 1980 Sakakibara, Yokomori and the Japanese school
- 1984 A theory of the learnable (Valiant)
- 1986 Angluin’s active learning setting
- 1992 RPNI
- **1993 the first ICGI workshop**

The beginning

- One may arrive at a grammar by intuition, guess-work, all sorts of partial methodological hints, reliance on past experience, etc.
- It is no doubt possible to give an organized account of many useful procedures of analysis, but it is questionable whether these can be formulated rigorously, exhaustively and simply enough to qualify as a practical and mechanical discovery algorithm [for grammars]. [Cho57]



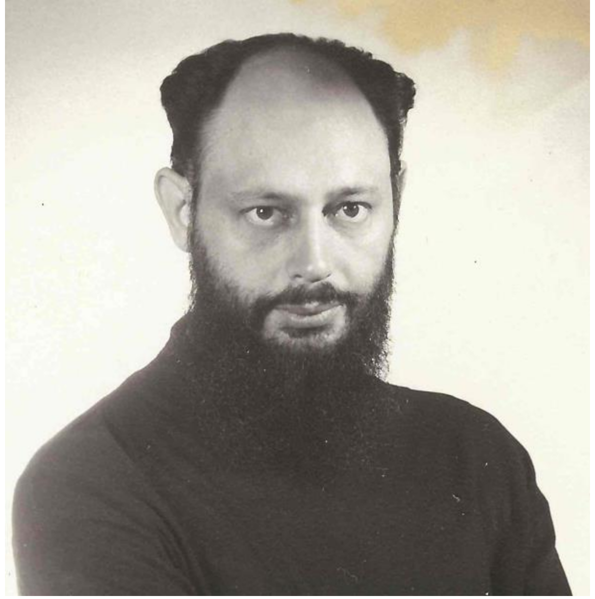


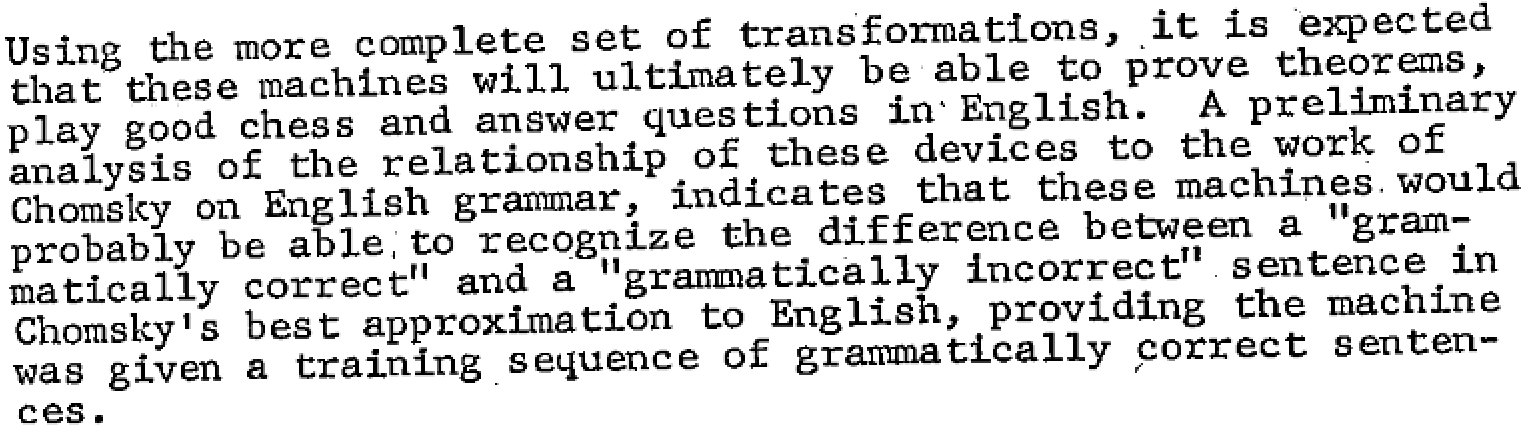
Too much faith should not be put in the powers of induction, even when aided by intelligent heuristics, to discover the right grammar. After all, stupid people learn to talk, but even the brightest apes do not. [Cho63]

On the other hand

- Miller & Chomsky 1957-1967
- LAD : Language Acquisition Device
- The goal is to find a procedure (an algorithm) which is able to build a grammar given utterances of a language

Ray Solomonoff





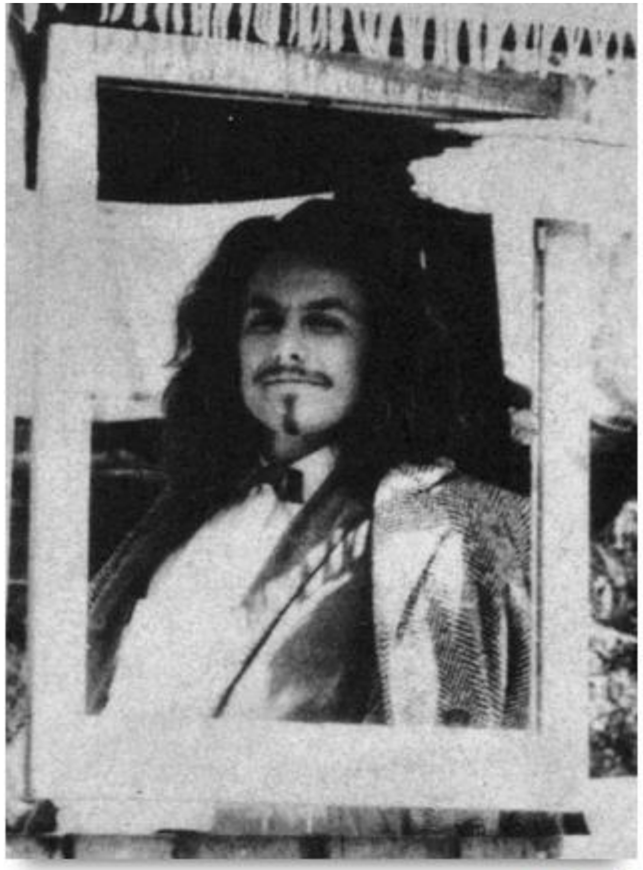
Using the more complete set of transformations, it is expected that these machines will ultimately be able to prove theorems, play good chess and answer questions in English. A preliminary analysis of the relationship of these devices to the work of Chomsky on English grammar, indicates that these machines would probably be able to recognize the difference between a "grammatically correct" and a "grammatically incorrect" sentence in Chomsky's best approximation to English, providing the machine was given a training sequence of grammatically correct sentences.

1956 "An Inductive Inference Machine," (pdf 1,400 k) Abstract Report circulated at the Dartmouth Summer Workshop on Artifical Intelligence, August 1956

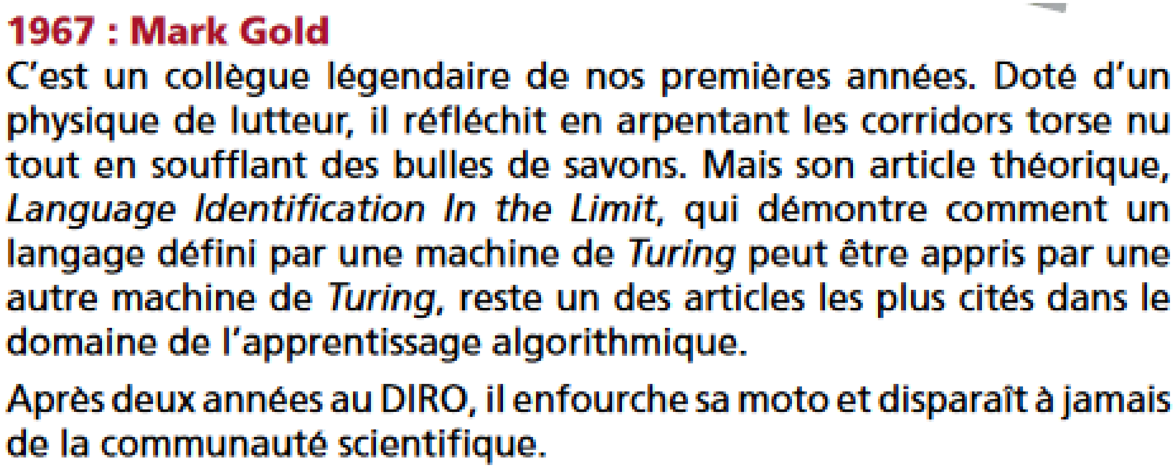
Solomonoff 1956

- **Abstract**
- **A machine is described which is designed to operate as human beings seem to. Inductive inferences are made by classifying events and their outcomes within categories that have been useful in the past, and by means of a small set of transformations, the system derives new categories that are likely to be useful in the future. These are tested empirically for usefulness in prediction, and useful ones combined with older useful categories to create new categories. These in turn are tested and the process is repeated again and again.**
- **A simplified system has been developed; it's attributes are described, and some future aspects, such as a system to improve itself are considered.**
- **A preliminary analysis of the relation of such systems to the work of Chomsky on English grammar is discussed.**

E. Mark Gold



E Mark Gold



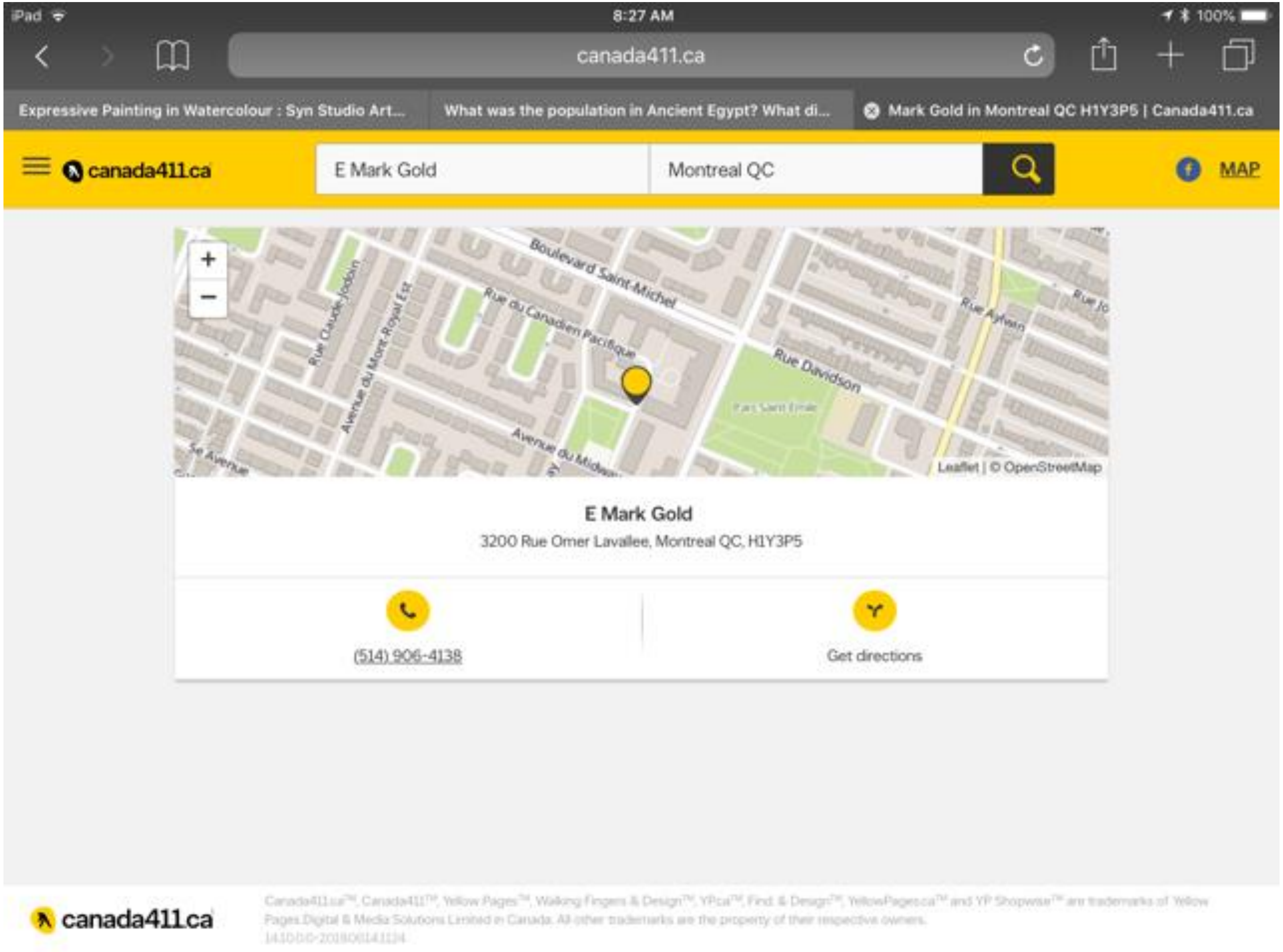
1967 : Mark Gold

C'est un collègue légendaire de nos premières années. Doté d'un physique de lutteur, il réfléchit en arpentant les corridors torse nu tout en soufflant des bulles de savons. Mais son article théorique, *Language Identification In the Limit*, qui démontre comment un langage défini par une machine de *Turing* peut être appris par une autre machine de *Turing*, reste un des articles les plus cités dans le domaine de l'apprentissage algorithmique.

Après deux années au DIRO, il enfourche sa moto et disparaît à jamais de la communauté scientifique.

https://www.iro.umontreal.ca/~echo/40e_Web/Posters/diro_12affiches_finales.pdf

Thanks to Jean Vaucher, Montreal



iPad
8:27 AM
100%
canada411.ca
Expressive Painting in Watercolour : Syn Studio Art...
What was the population in Ancient Egypt? What di...
Mark Gold in Montreal QC H1Y3P5 | Canada411.ca
canada411.ca
MAP
+
−
Boulevard Saint-Michel
Rue du Canadien Pacifique
Rue Claude-Jodoin
Avenue du Mont-Royal Est
Rue Davidson
Rue Aylwin
Avenue du Midway
Parc Saint-Émile
Leaflet | © OpenStreetMap

E Mark Gold

Montreal QC

E Mark Gold

3200 Rue Omer Lavallee, Montreal QC, H1Y3P5

☎ (514) 906-4138

Get directions

canada411.ca

Canada411.ca™, Canada411™, Yellow Pages™, Walking Fingers & Design™, YP.ca™, Find. & Design™, YellowPages.ca™ and YP Shopwise™ are trademarks of Yellow Pages Digital & Media Solutions Limited in Canada. All other trademarks are the property of their respective owners.
1410.0.0-201806141124

First thoughts on grammatical inference

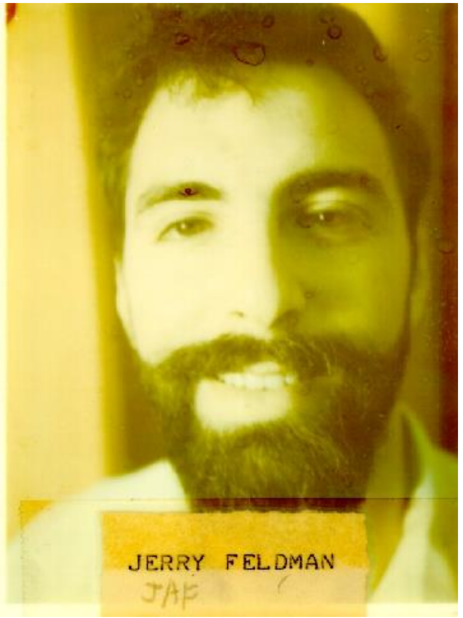
Authored By: J. A. Feldman

Paper Title: First thoughts on grammatical inference

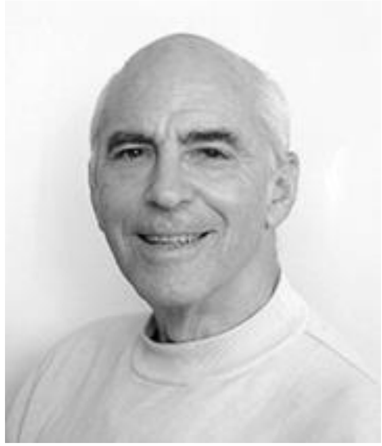
Publisher: Stanford University Artificial Intelligence Memo 55

Publication Date: 1967

GRAMMATICAL COMPLEXITY AND INFERENCE
BY
JEROME A. FELDMAN
JAMES G IPS
JAMES J. HORNING
STEPHEN REDER
SPONSORED BY
ADVANCED RESEARCH PROJECTS AGENCY
ARPA
ORDER NO. 457
TECHNICAL REPORT NO. CS 125
JUNE 1969



Jim Horning 1942 2013

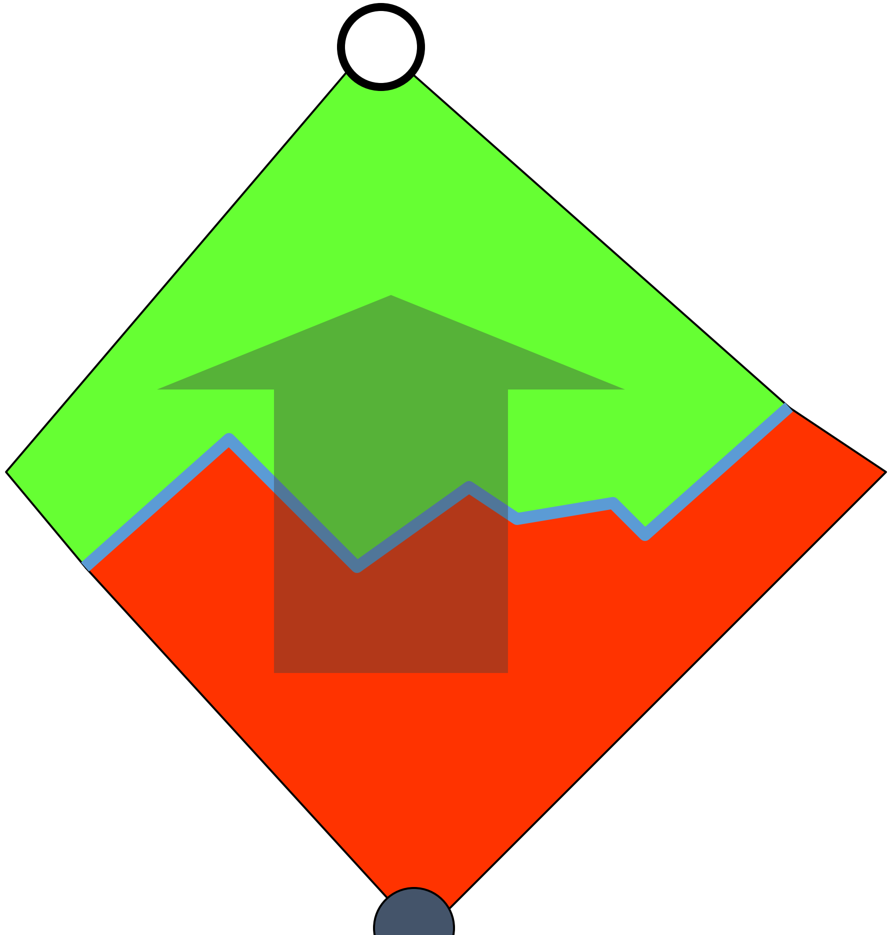


King-Su Fu

- Dr. **King-Sun Fu** (Chinese: 傅京孫; October 2, 1930—April 29, 1985)[4] was a Goss Distinguished Professor at Purdue University in West Lafayette, Indiana. He was instrumental in the founding of International Association for Pattern Recognition (IAPR), served as its first president, and is widely recognized for his extensive contributions to- and a pioneer in- the field of pattern recognition (within computer image analysis) and machine intelligence

Laurent Miclet

- 1980 : Regular Inference with a Tail-Clustering Method.

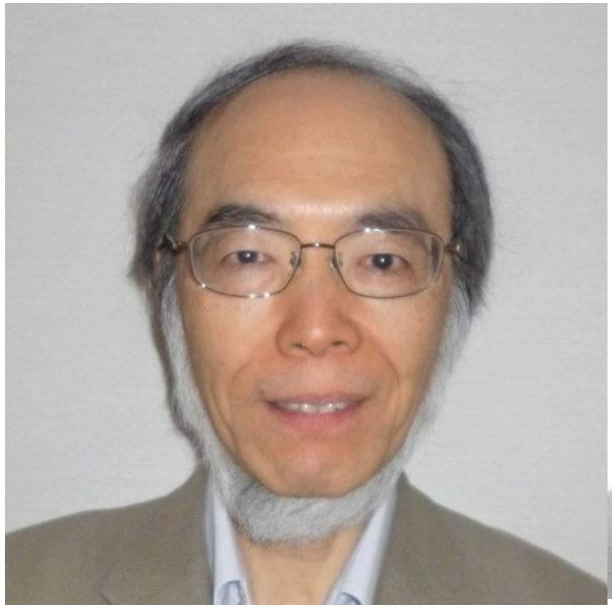


The American school

- Pitt, Warmuth, Schapire, Rivest, Kearns…
- And specially Dana Angluin

	λ	*a*
λ	1	0
a	0	0
b	1	0
aa	0	0
ab	1	0

The Japanese school



- Takashi Yokomori (1987…)
- Yasubumi Sakakibara (1988…)
- …

- YS: Grammatical Inference in Bioinformatics. IEEE Trans. Pattern Anal. Mach. Intell. 27(7): 1051-1062 (2005)

The Spanish school

- Enrique Vidal, Universidad Politecnica de Valencia
- Jose Oncina, Universidad de Alicante

The early days… the battle between empirical and theoretical GI

- The Tomita benchmark
- *M. Tomita. Learning Of Construction Of Finite Automata From Examples Using Hill-Climbing. Pittsburgh, Pennsylvania 15213, May 1982. Sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-81-K-1539.*

N5

Accept

- 11
- 00
- 1001
- 0101
- 1010
- 1000111101
- 1001100001111010
- 111111
- 0000

Reject

- 0
- 111
- 010
- 000000000
- 1000
- 01
- 10
- 1110010100
- 010111111110
- 0001
- 011

As a conclusion

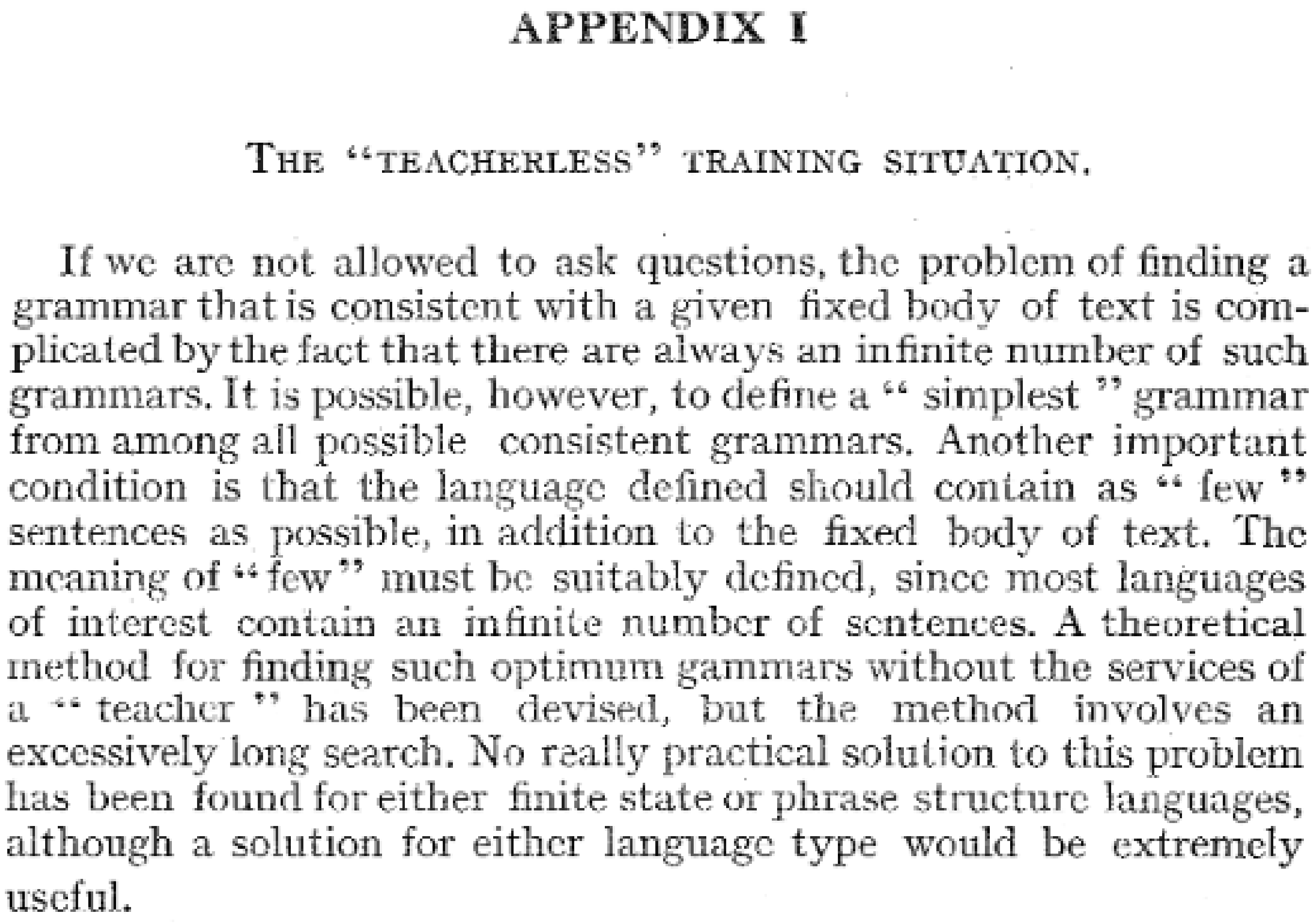
- Also, and not mentioned here
  - Genetic algorithms (Simon Lucas,…)
  - Neural networks (Jurgen Schmidhuber, Jordan Pollock)
  - Pattern recognition (SSPR)
- …

Yuji Takada, CEO, RunMyProcess

3. Some ideas

3.1 Simplicity

"The Mechanization of Linguistic Learning," (pdf 595 k) Second International Congress on Cybernetics, pp. 180-193, 1958 Solomonoff

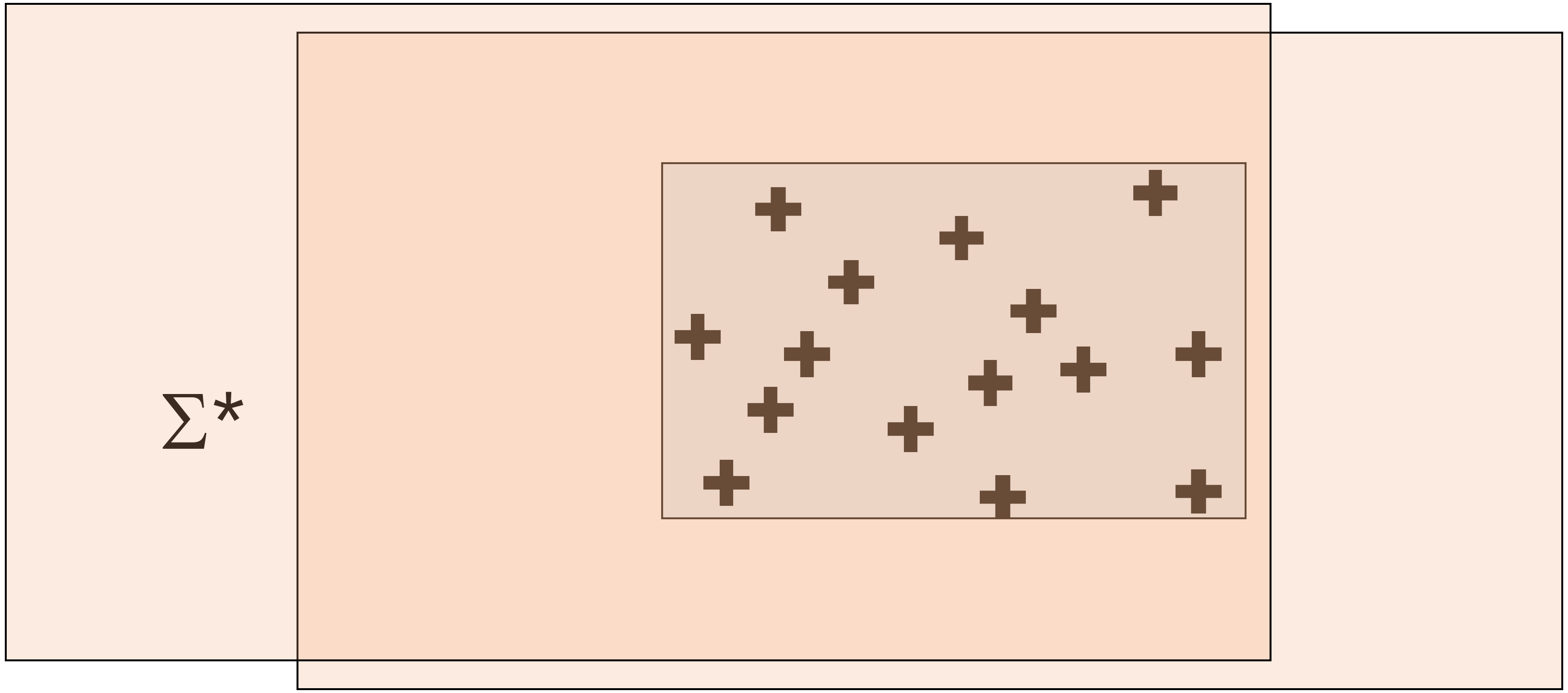


APPENDIX I

The "teacherless" training situation.

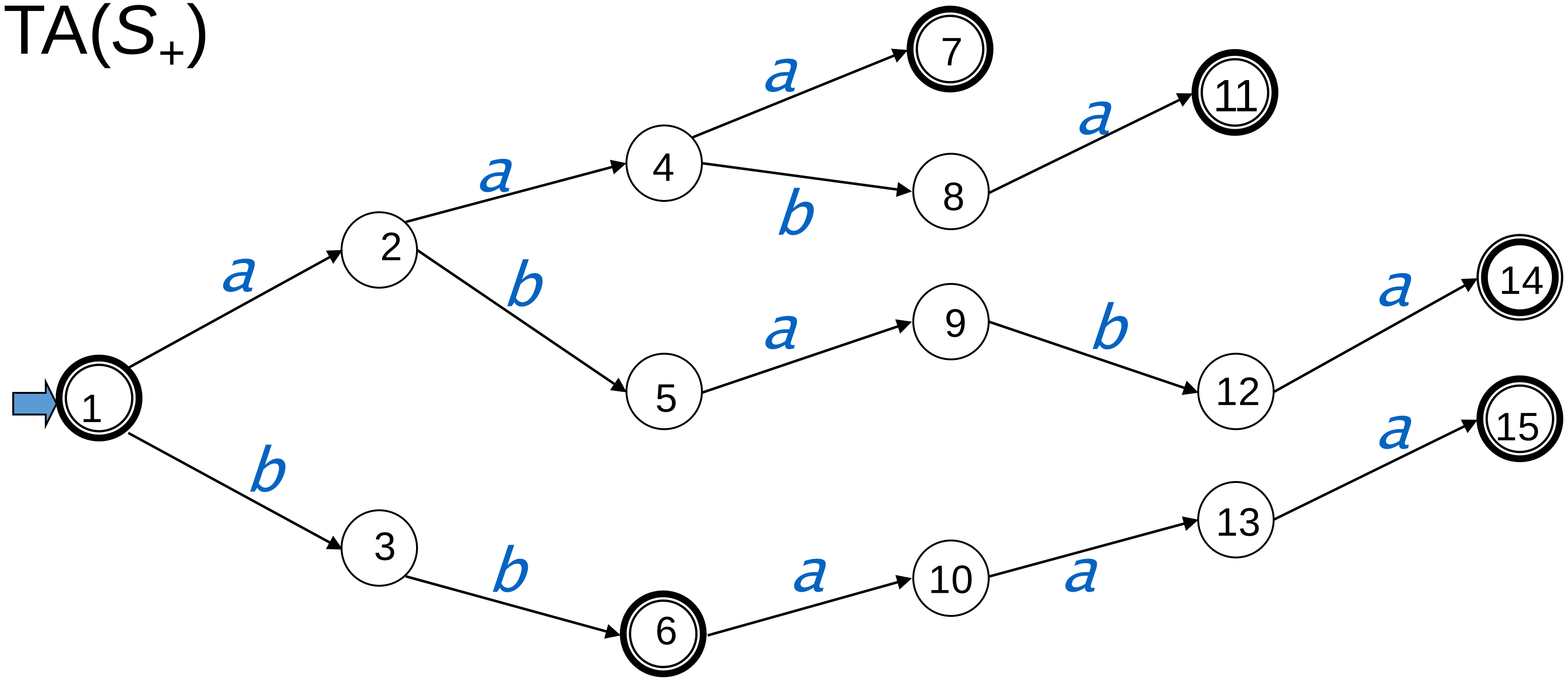
If we are not allowed to ask questions, the problem of finding a grammar that is consistent with a given fixed body of text is complicated by the fact that there are always an infinite number of such grammars. It is possible, however, to define a " simplest " grammar from among all possible consistent grammars. Another important condition is that the language defined should contain as " few " sentences as possible, in addition to the fixed body of text. The meaning of " few " must be suitably defined, since most languages of interest contain an infinite number of sentences. A theoretical method for finding such optimum gammars without the services of a " teacher " has been devised, but the method involves an excessively long search. No really practical solution to this problem has been found for either finite state or phrase structure languages, although a solution for either language type would be extremely useful.

Consider learning rectangles



A simple grammar as the tightest one? (the language containing least strings)

PTA(S_+)



S_+={λ, *aaa*, *aaba*, *ababa*, *bb*, *bbaaa*}

Why are we interested in simplicity?

- As a way around the poverty of the stimulus (Chater & Vitanyi 2006)
- As a way of obtaining an Occam algorithm (and thus a PAC algorithm)

Routes to simplicity

- Finding simpler languages (k-testable, k-reversible)
- Simple PAC and PACS
- MDL (minimum description length)

- We can also have simple distributions in which we can reason: « this string is simple and it is not in the learning sample; therefore, with high probability it is not in the language »

Simplicity today?

- Ample room for progress:
  - Simpler classes of languages
  - Make hypotheses about the distributions to allow use of what is present and what is not
- Puzzling:
  - Recurrent neural networks are not simple

3.2 Identification

Limiting Recursion
E. Mark Gold
The Journal of Symbolic Logic
Vol. 30, No. 1 (Mar., 1965), pp. 28-48

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

- […]Functions, sets, and functionals which are decidable by such infinite algorithms will be called limiting recursive. These, together with classes of objects which can be *identified in the limit*, are the subjects of this report.

Gold, E Mark, Language identification in the limit, RM-4136-PR, the RAND Corporation, 1964. *I can't find this reference*

The general idea

- Information is presented to the learner who updates its hypothesis after each piece of data
- At some point, always, the learner will have found the correct concept and not change from it

Example

2

{2}

3

{2, 3}

5

7

Fibonacci numbers

11

Prime numbers

103

23

31

A game: beating the box

A black box generates numbers from a sequence. We have to guess the next number. The black box indicates yes or no depending on if we have guessed the next element of the sequence (and gives us this next element)

Some questions

- Can we always beat the box?
  - Not if the box can change its rule on the fly after seeing your guess.
  - Not if the function is not computable.
- When do we stop?
  - When after a certain point we do not change our mind... But then we don’t know for sure we are correct!

Is identification in the limit a good learning model?

- In 1967 it was
  - At last a rule!
  - It did allow extensions for complexity
  - It represents a minimum: if a class is not identifiable in the limit there is a problem
- Today it only covers very limited settings:
  - We have to be sure there is something in the class to rediscover
  - We have to accept not to know anything about how well we are doing

Identification today?

- (sorry) Unless the setting justifies it (software verification-perhaps) identification is the limit should remain a side result.
- Challenge: can we do better? Can we also say something about how well the algorithm is doing?

3.3 Complexity

Some dates to take into account

- Hartmanis and Stearns, “On the Computational Complexity of Algorithms” 1965
- Edmonds 1965 (believed to invent classes P and NP)
- Cook 1971, Karp 1972, Garey-Johnson 1978.
- Most GI papers from the 60s and 70s are about decidability.

Complexity… in what?

- Obviously it is harder to learn English than ab^*a
- It is tempting to say that some languages are more complex than others. This was followed by Feldman & al 69
- But the same language $(a+b)^*a(a+b)^n$ is recognized by a 2^n state DFA and an n+1 state NFA.
- (this is in line with the early PAC results for Boolean formulae, 1977)

What do we count?

- We can try to count
  - global time
  - update time
  - errors before converging (IPE)
  - mind changes (MC)
  - queries
  - good examples needed

Main landmarks

- 1978: Gold proves that it is NP hard to find the smallest DFA consistent with a complete sample
- 1978 Angluin proves that it is NP hard to find the smallest regular expression consistent with a complete sample
- 1989 Pitt & Warmuth prove that even polynomial approximation is hard

Complexity in terms of data

- Cdlh87: DFA are learnable from polynomial time and data but NFA and CFGs are not
- Appealing because learning DFA is simpler
- Unconvincing because it still doesn’t tell us how good my current hypothesis is likely to be

What happens

- A nice result is one which says: if you are given so much data, so much time, then the result is expected to be good
- A typical PAC setting!
- What **we** **have** goes the other way round: If you want to learn this machine then you need so much data and time.

Complexity today?

- Given large amounts of data, having fast algorithms (linear time) matters
- Usage of solvers to find solutions

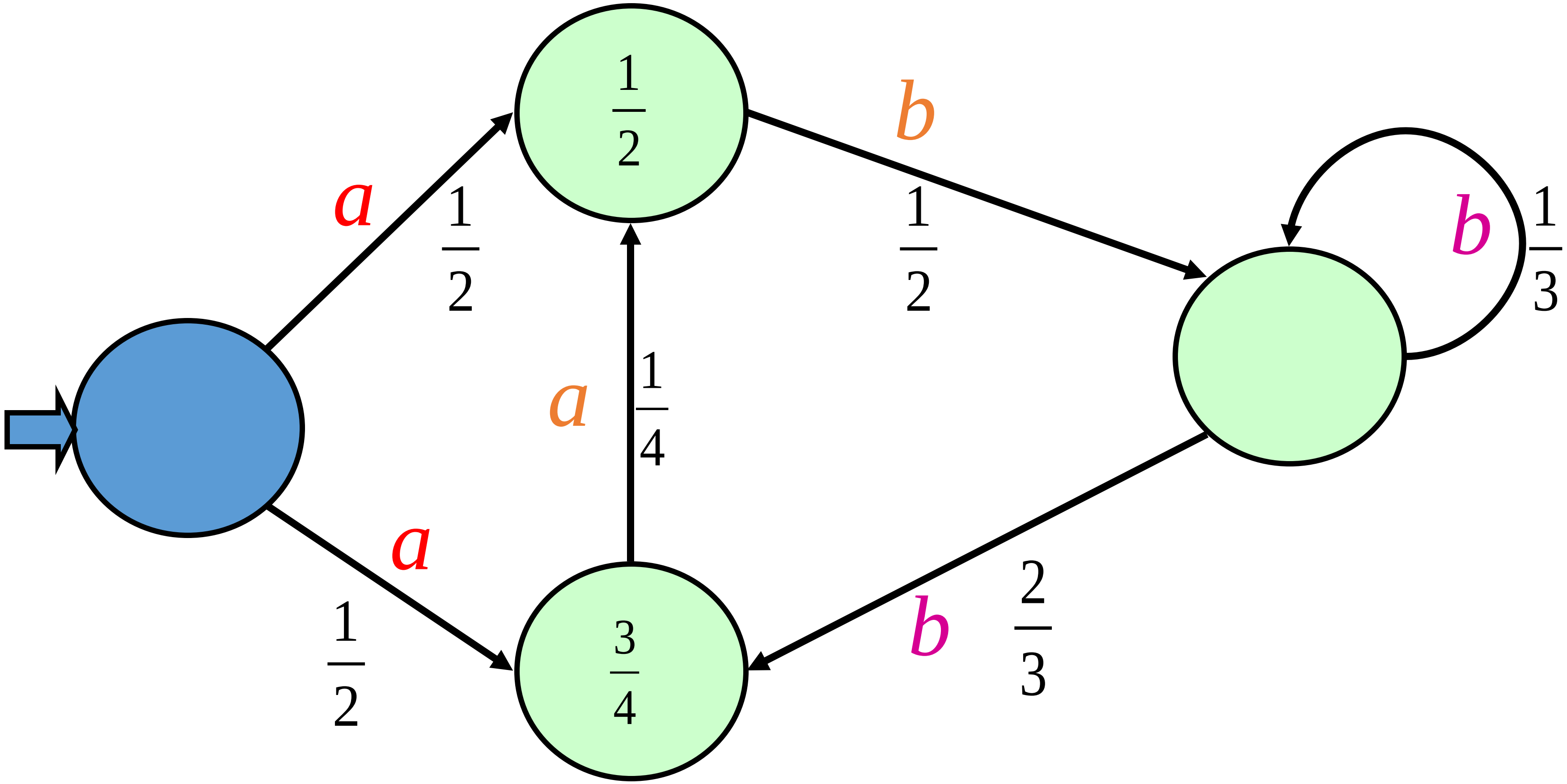
3.4 Approximation

The problems as they were encountered

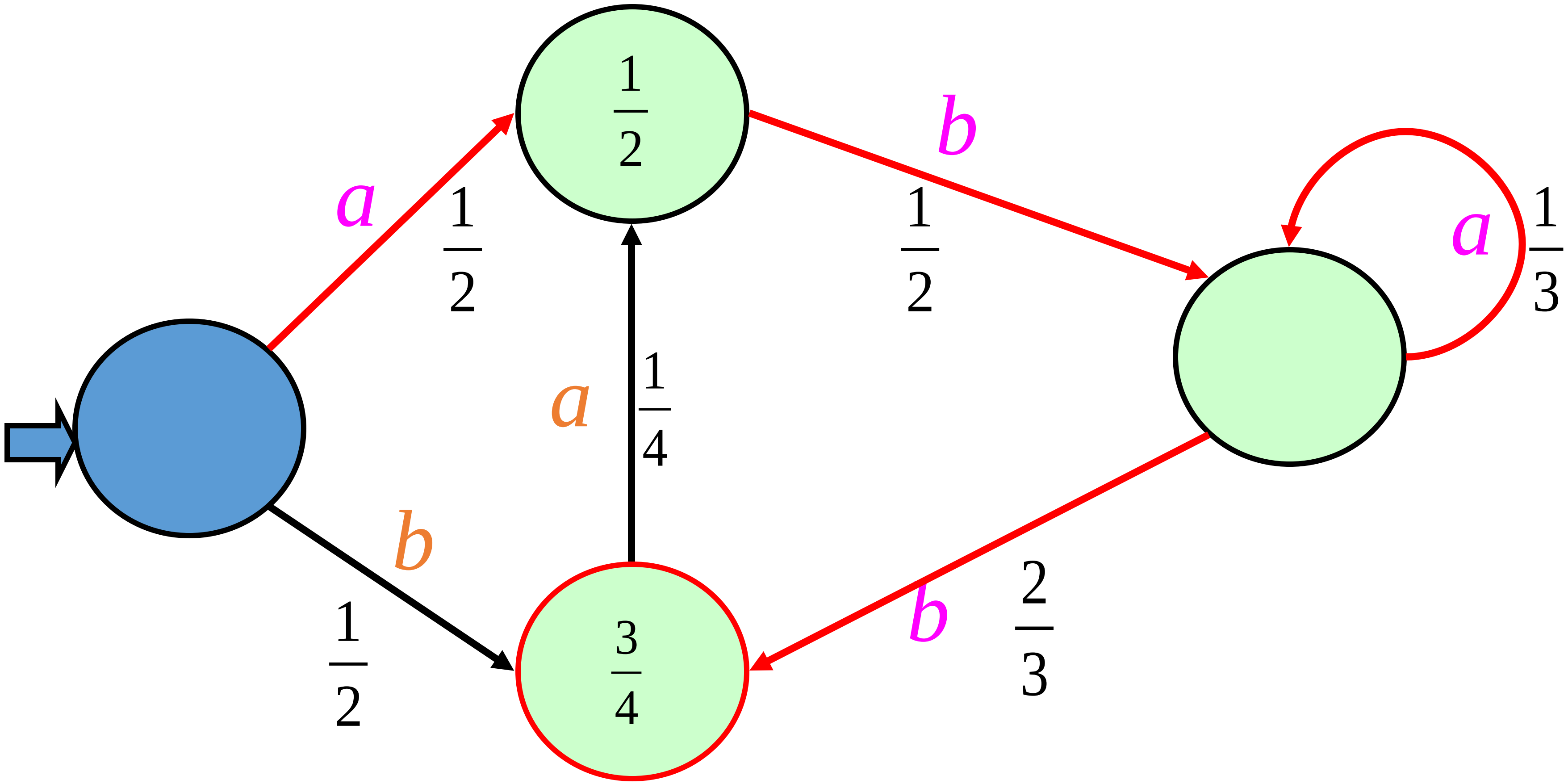
- Research on learning DFA and CFGs was receiving essentially negative results from the COLT community
- Solomonoff had advocated right from a start towards learning probabilistic grammars
- Distributions over strings were modelled through very simple artefacts: bigrams

Some common beliefs (in the 90s)

- We can talk about the support language $L=\{w\in\Sigma^*:\mathrm{P}(w)>0\}$ or $L=\{w\in\Sigma^*:\mathrm{P}(w)>t\}$ and attempt to learn this language.
- We can approximate any distribution over Σ by one represented by a DPFA
- We can approximate any distribution over Σ by one represented by a PFA
- We can approximate any distribution represented by a PFA by one represented by a DPFA



PFA: Probabilistic Finite (state) Automaton



$$\mathrm{P}_A(abab) = \frac{1}{2}\times\frac{1}{2}\times\frac{1}{3}\times\frac{2}{3}\times\frac{3}{4} = \frac{1}{24}$$

Approximating means distances

- Can we compute the distance between two automata? Between two grammars?
- Answer: yes but not very useful (there are exceptions!)
- Can we compute the distance between two PFA, two PCFGs?
- Answer: we have many possible distances. Each PFA or PCFG should be seen as an infinite dimension vector and most classic metrics will work
- Question: but can we compute them?

Related problems

- Are two PFA/PCFGs equivalent ?

- Find the consensus string (the most probable string)

For PFA

- $\langle d_{L1}, \mathcal{PFA}\rangle$, $\langle d_{V}, \mathcal{PFA}\rangle$ are NP-hard
- $\langle d_{L2}, \mathcal{PFA}\rangle$ is in P
- $\langle d_{KL}, \mathcal{DPFA}\rangle$ is in P

- $\langle EQ, \mathcal{PFA}\rangle$ is in P
- $\langle CS, \mathcal{PFA}\rangle$ is NP-hard but admits good parameterized algorithms

- **On computing the total variation distance of hidden Markov models**
- S Kiefer - arXiv preprint arXiv:1804.06170, 2018

For PCFGs

- The associated decision problems are **all** undecidable

(cdlh & Scicluna, unpublished)

Some **better** reasons for considering probabilistic grammars (in 1969)

There are many other motivations for using the frequencies of the strings in a positive information sequence (text presentation) to assist in grammatical inference:

(a) Since more information from the sequence is used, grammars may be discriminated earlier.

(b) The significance of "missing strings" can be evaluated.

(c) Inference can be conducted even in the presence of noise.

(d) Grammars for the same language may be discriminated on the basis of their agreement with observed frequencies.

(e) Complexity can be related to efficient encoding, and various results from information theory applied.

- http://infolab.stanford.edu/pub/cstr/reports/cs/tr/69/125/CS-TR-69-125.pdf

Approximation today?

- Clearly learning PFA (and PCFGs) is an important topic today
- Spectral methods, neural networks,…
- Importance of PAUTOMAC challenge

- Many open questions concerning PFA and PCFGs

3.5 Interaction

Interaction

- 1972: The active learning model is invented
- Key idea: if I can’t learn with interaction then I can’t learn without.
- To prove the validity of the model Angluin invents a purely theoretical algorithm to learn DFA from membership queries
- In 2010 Zulu

Interaction today?

- Please be here tomorrow at 9 for Frits’ talk!

Conclusion

Grammatical inference…

- … has been a topic studied by some great scientists
- … still needs a better fit theory-practice
- … still allows to look at many interesting research questions
- …has a great past and an even better future

Thank you!

Extras

Key dates

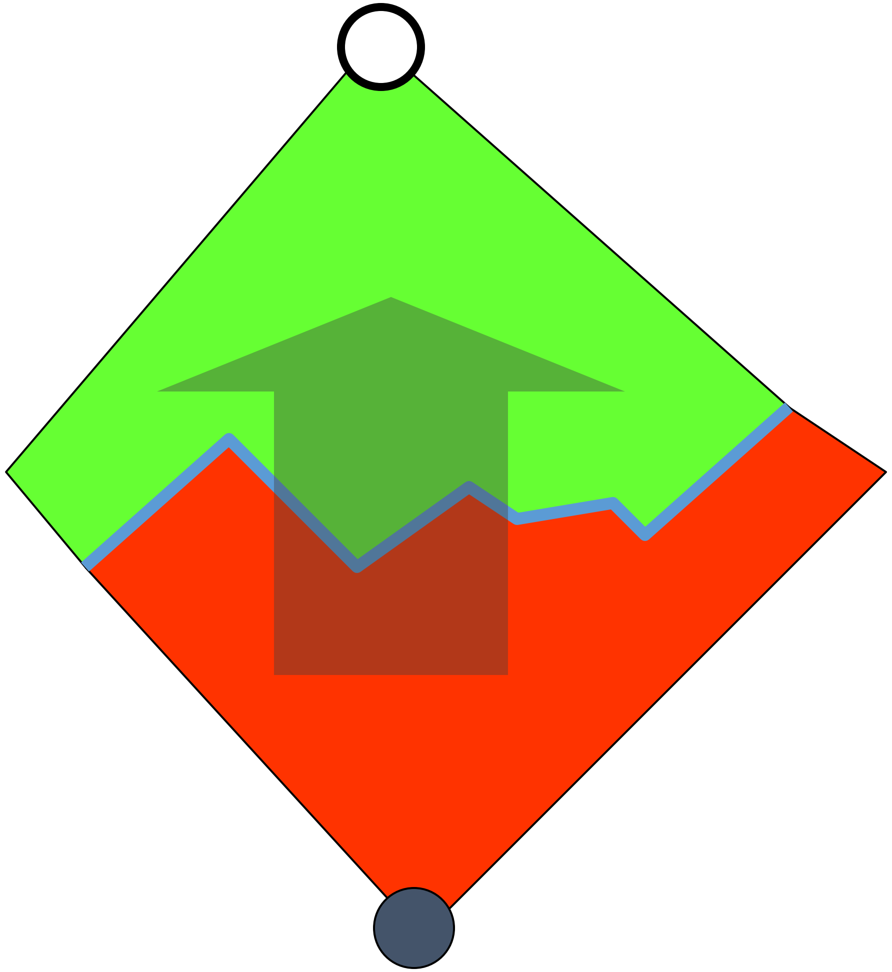
- The LAD: 1957
- PhD by Pieter Adriaans, linking with Kolmogorov complexity
- Work by François Denis
- Work about simple solutions and the MDL principle

- What is nice about simplicity? It attempts to solve the issue: if string « the » is absent from my dataset, then perhaps this isn’t english?

State splitting

Searching by splitting:

start from the one-state universal automaton, keep constructing *DFA* controlling the search with $<S_+, S_->$



PAC-learning

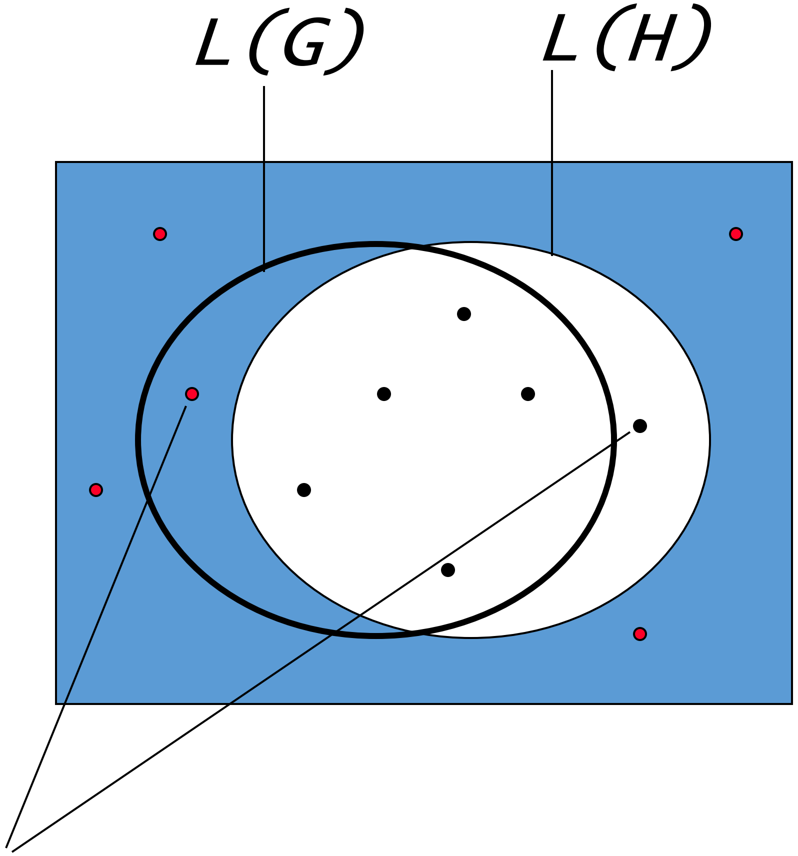
(Valiant 84, Pitt 89)

- $\mathcal{L}$ a class of languages
- $\mathcal{G}$ a class of grammars
- $\varepsilon > 0$ and $\delta > 0$
- *m* a maximal length over the strings
- *n* a maximal size of machines

H is ε - AC (approximately correct)*

if

$\Pr_{D}[H(x) \neq G(x)] < \varepsilon$



Errors: we want this < ε

(French radio)

- Unless there is a surprise there should be no surprise
- (after the elections, on 3rd of June 2008)

Results

- Using cryptographic assumptions, we cannot PAC-learn DFA
- Cannot PAC-learn NFA, CFGs with membership queries either

Proposal

- A grammar class is reasonable if it encodes sufficient different languages
- *Ie* with n bits you have 2^{n+1} encodings so optimally you should have 2^{n+1} different languages

But

- We should allow for redundancy and for some strings that do not encode grammars
- Therefore a grammar representation is reasonable if there exists a polynomial $p()$ and for any n the number of different languages encoded by grammars of size n is in $\theta(2^n)$