Inductive-Deductive Systems: A mathematical logic and statistical learning perspective

Nicolas Baskiotis, Michèle Sebag et Olivier Teytaud

Université Paris-Sud – CNRS – INRIA

Progic 2007

- God (=expert, client, scientist, engineer...) says
  - Zermelo-Fraenkel holds
  - Does the theorem "blabla" hold ?
- What do you reply ?

- God (=expert, client, scientist, engineer...) says
  - Zermelo-Fraenkel holds
  - Does the theorem "blabla" hold ?
- It depends on blabla (and on your computational power):
  - If ZF proves "blabla", then you reply "yes".
  - If ZF proves "not blabla", then you reply "no".

- God (=expert, client, scientist, engineer. . .) says
  - Zermelo-Fraenkel holds
  - Does the theorem "blabla" hold ?
- It depends on blabla (and on your computational power):
  - If ZF proves "blabla", then you reply "yes".
  - If ZF proves "not blabla", then you reply "no".
  - Otherwise ? . . .

Unfortunately, Godel's theorem ensures that God can find a question in the bad case.

- God (=expert, client, scientist, engineer...) says
  - Zermelo-Fraenkel holds
  - Does the theorem "blabla" hold ?
- It depends on blabla (and on your computational power):
  - If ZF proves "blabla", then you reply "yes".
  - If ZF proves "not blabla", then you reply "no".
  - Otherwise ? ...
- God is not happy, but he is gentle. He tells you that "blabla" holds.
- God aks: is "blabla_2" true ?

Essentially undecidable systems (e.g. ZF!): God can find a "blabla_2" for which you have no answer!

- Ok, we can not win if God is an adversarial, by essential undecidability.
- What if God is a distribution of probability ?
  $\Rightarrow (blabla_n)_{n \in N}$ independently identically distributed sequence of statements.
- At which rate can you reach a probability 1 ?

We show that if you *induce+deduce* axioms, you will converge faster than if you *deduce*.

Inductive/deductive systems

- Learning to Reason [Khardon and Roth 97]
- Induction vs Deduction:
  - deduction: $\forall n,\ C(n)$ leads to $C(1),\ C(2),\ \ldots$
  - induction: $C(1), C(3), C(7), \neg C(4), \neg C(6)$, leads to $C(2n+1) \wedge \neg C(2n)$
- Inductive/deductive system:
  - $e_1, \ldots, e_n$ available statements
  - the algorithm reads $e_1, \ldots, e_n$ and outputs a theory T
  - if T does not prove e_{n+1}, then T is not satisfactory.
  - if T is not consistent, then T is not satisfactory.

Why a probabilistic analysis ?

- Question: Does T prove e_{n+1} ?

- Worst case: incompleteness theorem (Gödel)
  there exists a statement e_{n+1} such that:
  - T does not prove e_{n+1}
  - and T does not prove $\neg e_{n+1}$

  $\Rightarrow$ for this criterion, all algorithms are equivalent:
  they always have a counter-example!

- Questions:
  is a statistical analysis possible ?
  can one estimate $proba(trouble)$?

Outline

Context

Theorems

Implications for Turing-computable approximations

Conclusion

PAC learning [Vapnik et al. 74]

- Analysis of the inductive power
- Z a domain of examples and $F \subset P(Z)$
  - $X \subset Z$ is « shattered » by F
    $\iff \forall X' \subset X \; \exists f \in F$ such that $X' = f \cap X$
  - VC-dim of F : maximal cardinal of a set shattered by F
- If $e_1, \dots, e_n, \dots$ are independent and identically distributed, generated in some unknown "target" theory,
and if F :
  - has finite VC-dim $\Rightarrow$ probability of "trouble" $O(V/n)$
  - does not shattered an infinite set
    $\Rightarrow$ in some cases $O(V(target)/n)$
  - shatters an infinite set
    $\Rightarrow$ arbitrarily slow convergence

Formalization, 1

- Modelization
  - consider ζ an essentially undecidable set of axioms
  - consider a set of axioms $E_n = \{e_1, \ldots, e_n\}$ (independent identically distributed according to M, consistent with ζ)
  - the algorithm reads E_n and outputs A_n such that $A_n,\ \zeta \vdash E_n = \{e_1, \ldots, e_n\}$
- We study:
  - uncompleteness: $L_n = M(\{e \mid A_n, \zeta \nvdash e\})$
  - compactness $DL(A_n)$: description length of A_n

Formalization, 2

- Three families of algorithms:
  - deduction: $A_n = E_n$
  - deduction with pruning: $A_n \subset E_n$, minimal
  - induction+deduction: A_n as "small" as possible
    A_n not necessarily included in E_n
- Particular cases:
  - ζ complete, then $L_n = 0$ ($A_n = \emptyset$)
  - ζ ess. undecidable, worst case on e_n, then $\forall n, L_n = 1$
    (Gödel's theorem)
- What happens if (i) ess. undecidable (ii) probability instead of worst case ?

Outline

Context

Theorems

Implications for Turing-computable approximations

Conclusion

Fundamental theorem

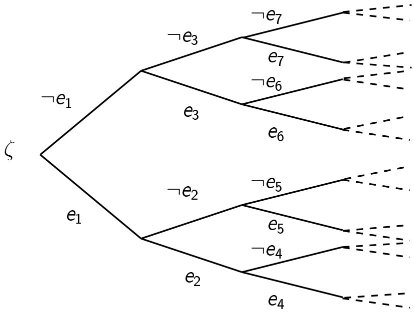
- Consider ζ an essentially undecidable set of axioms.

- Consider T' the set of consistent theories including ζ,
  $\Rightarrow$ T' shatters an infinite set $\Rightarrow$ disaster.

- Consider $T \subset T'$ the set of theories generated by an axiom set with finite description length,
  $\Rightarrow$ T has an infinite VC-dimension $\Rightarrow$ depends on the algorithm.

Sketch of the proof

- Build an infinite sequence of statements $\{l_i\}$
  shattered by T

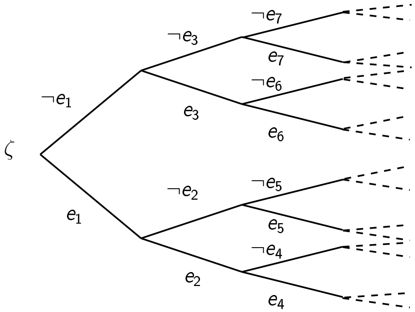


- not the same statements on the left and on the right!

$\Rightarrow$ Modify the tree

Sketch of the proof

- $l_1 = e_1,\ r_1 = \neg e_1, \qquad l_2 = e_1 e_2 \vee \neg e_1 e_3,\ r_2 = e_1 \neg e_2 \vee \neg e_1 \neg e_3,$
$l_3 = e_1 e_2 e_4 \vee \neg e_1 \neg e_2 e_5 \vee \neg e_1 e_3 e_6 \vee \neg e_1 \neg e_3 e_7, \ldots$



- $\zeta, l_1, r_2, l_3 \vdash l_1, \vdash l_3, \not\vdash l_2$
- $\{l_1, \ldots, l_n\}$ is shattered by T

Outline

Context

Theorems

Implications for Turing computable approximations

Conclusion

Implications

- If induction+deduction *and* finite description length:
  - target asymptotically reached
  - fast convergence ($O(log(n)/n)$)
  - description length of output bounded and converging to the MDL of the target
- Otherwise
  - arbitrarily slow convergence rate
  - description length might run to infinity

Implications for Turing computable machines

• Question : We have used an oracle (in 0') for solving MDL problems. Is this necessary ?
• Approximation of the "idealize" MDL principle (in 0') by finite-length deduction (in 0, i.e. computable)
  $\Rightarrow$ huge complexity of the algorithm, but

$\Rightarrow$ Same results as in 0'

- convergence of L_n as $O(log(n)/n)$
- target theory almost surely reached
- description length converges to the optimal possible one

Proof in the paper

Conclusion

induction+deduction > deduction
induction+deduction > deduction+pruning
on the set of theories with finite description length

- Probabilistic framework for the analysis of ess. undecidable Inductive Logic Programming
- Induction + finite description length → convergence
- Turing-computable (but very expensive)
- Shorter axiom sets are better
- Making a difference between facts, which are definitely true, and induced facts, which are unstable.

Possible applications:

- Merging ontologies in ess. undecidable (i.e., natural!) settings
- A principled way for expert systems in ess. undecidable settings: approximate MDL + deduction
- Epistemology of mathematics: induction is precisely what mathematicians do.