Introduction to *Symbolic Regression*

Andrzej Odrzywołek

Zakład Teorii Względności i Astrofizyki, Instytut Fizyki UJ

27 stycznia 2016

$$\sqrt[3]{x} = x^{\frac{\log(x)}{\log\left(\frac{\log(((x^x)^x)^x)}{\log(x)}\right)}}$$

Czy potrafisz rozpoznać liczby?

$$2.718281828459045$$

$$3.141592555900037$$

$$1.4142135623730951$$

$$1.618033988749895$$

$$0.69314718055994 53$$

Zadanie wydaje się łatwe...

$$e \simeq 2.718281828459045$$

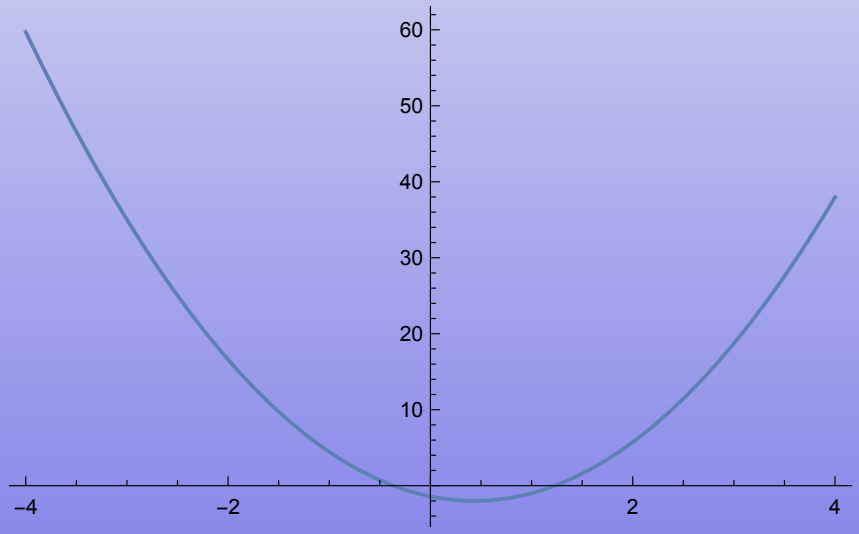
$$1 + e^{\frac{1}{1+\frac{1}{-1-\frac{1}{1+\frac{1}{e}}}}} \simeq 3.141592555900037$$

$$\sqrt{2} \simeq 1.414213562373095 1$$

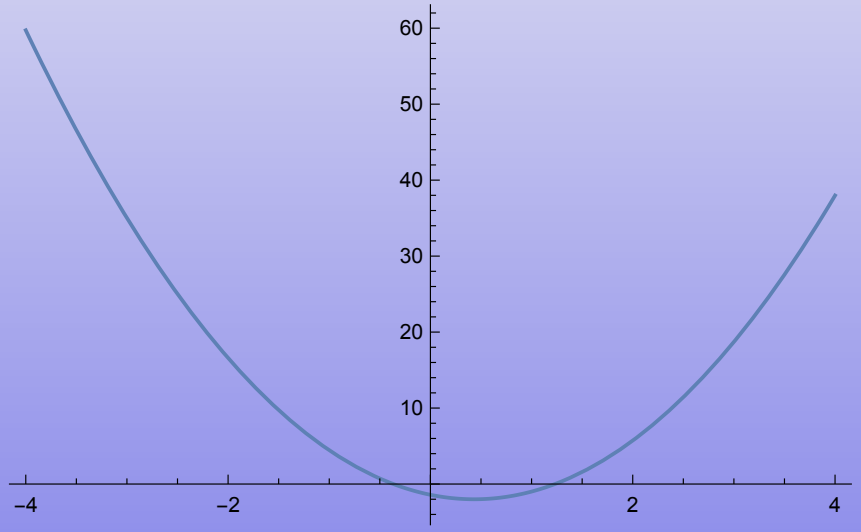
$$\frac{1+\sqrt{5}}{2} \simeq 1.618033988749895$$

$$\ln 2 \simeq 0.69314718055994 53$$

Czy można rozpoznać funkcję jednej zmiennej?

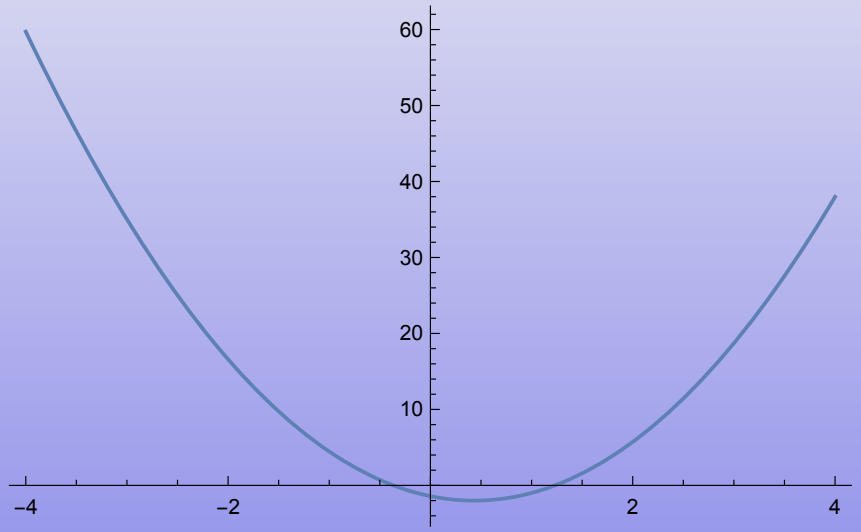


Czy można rozpoznać funkcję jednej zmiennej?



$$y = ax^2 + bx + c$$

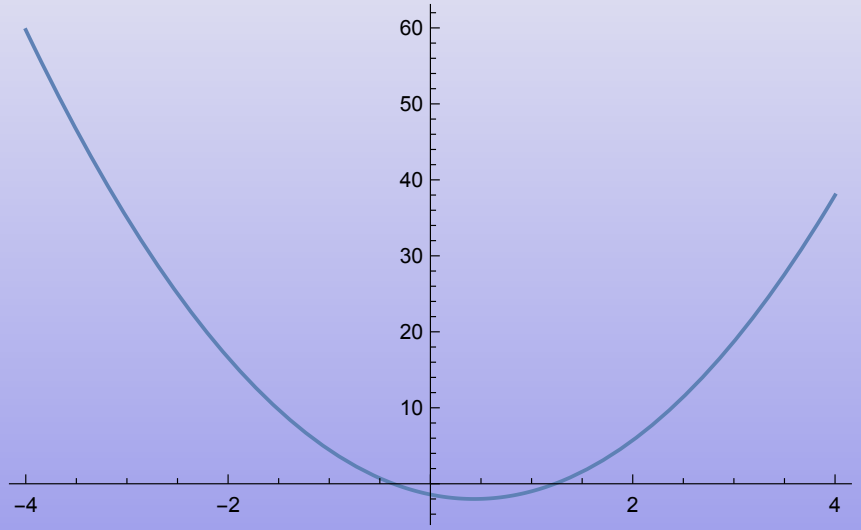
Czy można rozpoznać funkcję jednej zmiennej?



$$y = ax^2 + bx + c$$

$$y = 3.14x^2 - 2.71x - 1.41$$

Czy można rozpoznać funkcję jednej zmiennej?

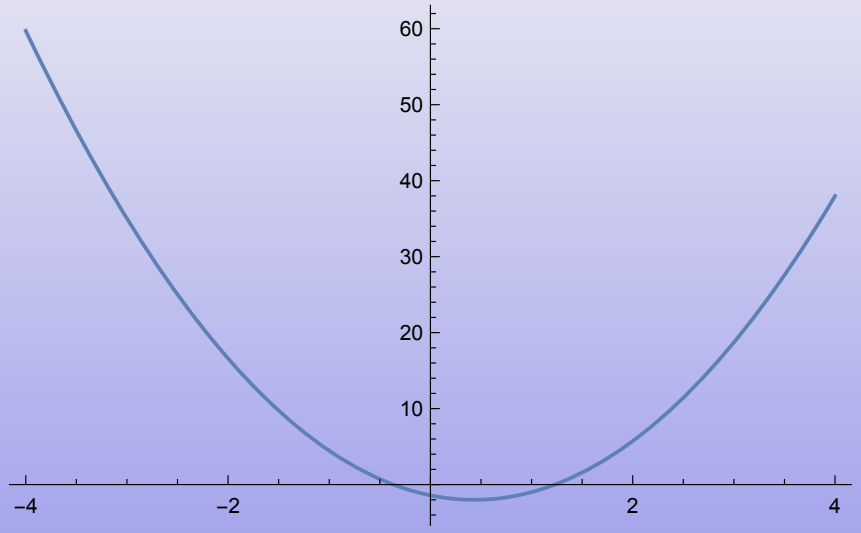


$$y = ax^2 + bx + c$$

$$y = 3.14x^2 - 2.71x - 1.41$$

$$y = \pi x^2 - ex - \sqrt{2}$$

Czy można rozpoznać funkcję jednej zmiennej?



$$y = ax^2 + bx + c$$

$$y = 3.14x^2 - 2.71x - 1.41$$

$$y = \pi x^2 - ex - \sqrt{2}$$

Powyższy proces, którego wynikiem jest **ścisły** matematycznie wynik, określamy mianem **regresji symbolicznej**.

Przykłady szukania rozwiązań metodą brute force

1. liczby naturalne (trywialny przykład)
2. liczby wymierne (rozwiązany nietrywialny przykład)
3. wielomiany o współczynnikach całkowitych (nieoczywisty przykład)
4. rozpoznawanie stałych (działający trudny przykład)
5. funkcje „elementarne” jednej zmiennej
6. funkcje wielu zmiennych i funkcje z parametrami

Zadanie

Dla jakiego naturalnego n spełnione jest równanie:

$$F_c(1/n!) \cdot fib(2n-2) = 1$$

gdzie F_C to schody Cantora a fib - ciąg Fibonacciego?

Oczywista metoda brute-force to wstawianie po kolei: 0,1,2,3,4, ...

UWAGA!

Gdyby chodziło o rozwiązania całkowite, wstawiamy po kolei:

$$0, 1, -1, 2, -2, 3, -3 \ldots$$

Idiotyczny błąd, formalnie możliwy do popełnienie, to wstawianie najpierw liczb dodatnich, potem ujemnych, lub wstawianie w kolejności, która pomija pewne liczby, np:

$$(-1)^k k \quad \text{zamiast} \quad \frac{1-(-1)^k(2k+1)}{4}.$$

O ile dla liczb całkowitych tego typu pomyłka byłaby absurdalna, w przypadku np: funkcji elementarnych podobny błąd łatwo popełnić i przeoczyć.

Liczby naturalne/całkowite : podejście funkcyjne

Zdefiniujmy trzy funkcje (zero, sukcesor, negacja):

$$Z(n) = 0, \qquad S(n) = n + 1, \qquad N(n) = -n.$$

Dowolne złożenie funkcji Z, S, N zawsze ma sens, i reprezentuje funkcję stałą, przyjmującą wartości całkowite.
Dowolny ciąg znaków ZSN reprezentuje liczbę całkowitą, np. cztery:

$$NSSSSZ.$$

Taka reprezentacja nie jest jednoznaczna, ale jest dobrą podstawą do metod genetycznych, gdyż po operacjach duplikacji, delecji, insercji czy odwrócenia nadal reprezentuje liczbę całkowitą. Pozwala także na wprowadzenie śmieciowego DNA, typu NNNNNNNNNN czy markerów np: ZZZ czy NSZZ pełniących rolę kodonu STOP/START.

Spostrzeżenia

- Zmiana znaku jest funkcją odwrotną do samej siebie. Funkcję $x + 1$ można zastąpić funkcją $-x + 1$, o tej samej własności.
- ciągi liter Z,S,N to liczby w systemie trójkowym
- można jednoznacznie powiązać liczbę całkowitą z numerem pierwszego pojawienia się (przy ustalonej kolejności „cyfr" Z, S, N)

Zadanie

Dla jakiego wymiernego r spełnione jest równanie:

$$?(r) \cdot F_c(r) = r - \frac{423}{2^{11}},$$

gdzie ? to „pytajnik Minkowskiego" a F_C schody Cantora?

Cantor podał metodę enumeracji liczb wymiernych $\frac{n}{m}$ o rosnących n, m. Liczby powtarzają się (np: $\frac{2}{4} = \frac{1}{2}$), ale redundancja jest niewielka, asymptotycznie dąży do złotego podziału.

Jak generować liczby wymierne bez powtórzeń?

Okazuje się, że od połowy XIX w. znane są sposoby enumeracji liczb wymiernych:

1. drzewo Sterna-Brocota (przeszukiwanie binarne „przedziału" $[0, \infty]$)
2. ciąg Fareya
3. ułamki łańcuchowe
4. składanie funkcji

Najciekawsze ponownie okazuje się podejście funkcyjne.

Funkcyjne generowanie liczb wymiernych

Zdefiniujmy cztery funkcje (one, minus sukcesor, negacja, odwrotność):

$$ONE(r) = 1, \qquad NSU(r) = -r + 1, \qquad NEG(r) = -r, \qquad INV(r) = 1/r.$$

Trzy ostanie są „samoodwrotne", i pozwalają poprzez składanie na generowanie wszystkich liczb wymiernych na zasadzie ułamka łańcuchowego.

Wszystko co powiedziano o liczbach całkowitych przenosi się na wymierne.

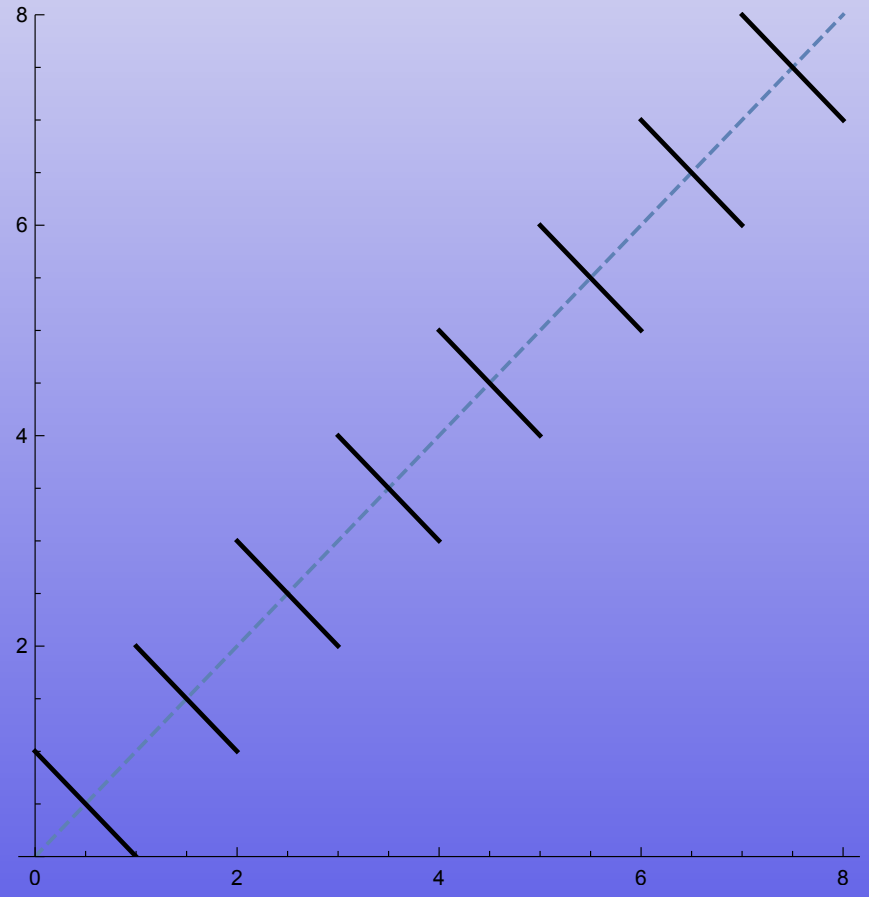
Niespodzianka

Istnieje funkcja, której składanie generuje po kolei wszystkie liczby wymierne bez powtórzeń. Jest ona złożeniem dwóch funkcji „self-inverse":

$$INV(r) = \frac{1}{r}, \qquad LAD(r) = Floor[x] + 1 - FractionalPart[x]$$

Wykres funkcji LAD(r)

$$LAD(x) = Floor[x] + 1 - FractionalPart[x]$$



Generowanie ułamków poprzez składanie funkcji

Okazuje się, że składając wielokrotnie funkcję:

$$f(x) = \frac{1}{Floor[x] + 1 - FractionalPart[x]}$$

otrzymujemy wszystkie liczby wymierne (w tym naturalne), w określonej kolejności, bez powtórzeń!

$$0, 1, \frac{1}{2}, 2, \frac{1}{3}, \frac{3}{2}, \frac{2}{3}, 3, \frac{1}{4}, \frac{4}{3}, \frac{3}{5}, \frac{5}{2}, \frac{2}{5}, \frac{5}{3}, \frac{3}{4}, 4, \frac{1}{5}, \frac{5}{4}, \frac{4}{7}, \frac{7}{3}, \frac{3}{8}, \frac{8}{5}, \frac{5}{7}, \frac{7}{2}, \frac{2}{7}, \frac{7}{5}, \frac{5}{8}, \frac{8}{3}, \frac{3}{7}, \frac{7}{4}, \frac{4}{5}, 5, \ldots$$

Jak metody dla liczb całkowitych i wymiernych można rozszerzyć na dowolne wyrażenia matematyczne?

PROJECT RAND

A DECISION METHOD FOR ELEMENTARY ALGEBRA AND GEOMETRY

ALFRED TARSKI

Prepared for Publication by J. C. C. McKinsey

This report, although published by the RAND Corporation, was written while the Project was a part of Douglas Aircraft Co., Inc.

August 1, 1948

(Revised May, 1951)

R-109

The RAND Corporation

1700 MAIN ST. • SANTA MONICA • CALIFORNIA

SECTION 1.

THE SYSTEM OF ELEMENTARY ALGEBRA

In this section we want to describe a formal system of elementary algebra — and in particular to define in a precise way the class of sentences of this system[7].

By a *variable* we shall mean any one of the following symbols:

$$x,\ x_1,\ x_2,\ldots;\ y,\ y_1,\ y_2,\ldots;\ z,\ z_1,\ z_2,\ldots\ .$$

We suppose that there are infinitely many variables and that they are arranged in a sequence, so that we can speak of the variable occupying the 1st, 2nd,..., nth place in the sequence. These variables are to be thought of intuitively as ranging over the set of real numbers.

By an *algebraic constant* we shall mean one of the following three symbols:

$$1,\ 0,\ -1\ .$$

By an *algebraic operation-sign* we shall mean one of the following two symbols:

$$+,\ \cdot\ .$$

The first is called the *addition sign*, and the second the *multiplication sign*.

By an *algebraic term* we understand any meaningful expression built up from variables and algebraic constants by means of the elementary operation-signs. Thus, for example,

$$x,\quad x_1 + y,\ -1\cdot x,\quad \left[\left(x_1 \cdot -1\right)\cdot x_1\right] + x_2$$

are algebraic terms. But

$$x + ,\quad \sqrt{2} + x$$

are not algebraic terms: the first, because it is meaningless; the second, because it involves the sign "$\sqrt{2}$", which is neither a variable nor an algebraic constant (in the restricted meaning we have given to the latter term).

If one wants a precise definition of algebraic terms, they can be defined recursively as follows: An algebraic term of first order is simply a variable or one of the three algebraic constants. If α and β are algebraic terms of order at most k, and if the maximum of the orders of α and β is k, then $(\alpha\cdot\beta)$ and $(\alpha + \beta)$ are algebraic terms of order $k + 1$. An expression is called an algebraic term if, for some k, it is an algebraic term of order k.

Elementarna algebra wg. Tarskiego

Oryginalnie, minimalny (startowy) zbiór symboli Tarskiego to:

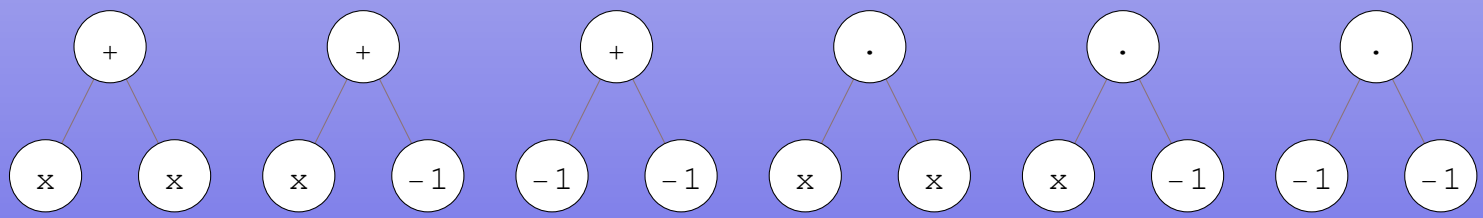
$$-1, 0, 1, x + y, x \cdot y, x, y, z, \ldots$$

Zero i 1 są redundantne:

$$1 = (-1) \cdot (-1), \qquad 0 = -1 + 1$$

Wygenerowanie wszystkich możliwych wyrażeń elementarnej algebry, polega na rekurencyjnym stosowaniu symboli i operacji bazowych. Tradycyjne symbole matematyczne traktujemy jako „skróty" np:

$$5 = 1 + 1 + 1 + 1 + 1, \qquad x^2 + 1 = x \cdot x + (-1) \cdot (-1), \ldots.$$

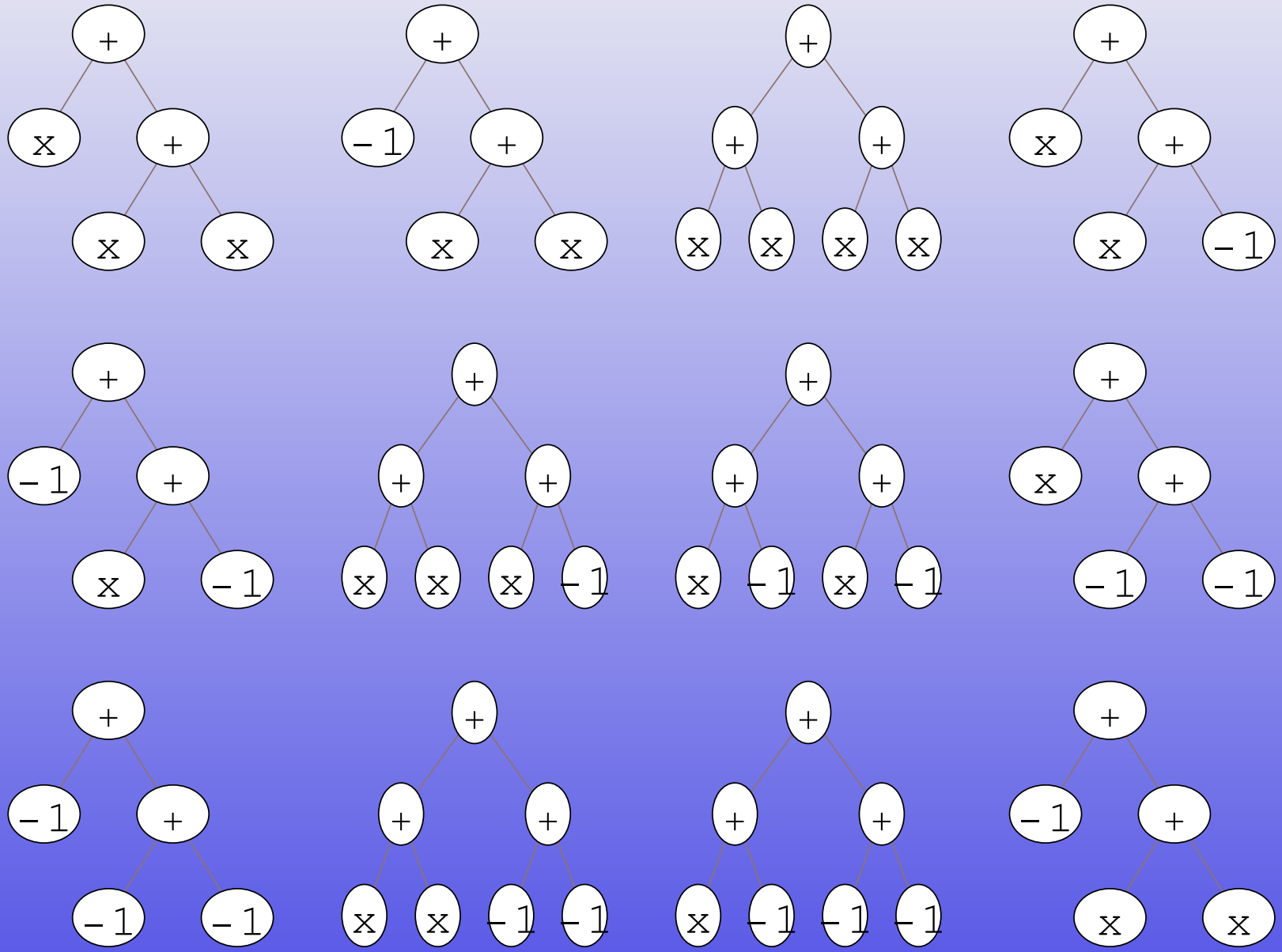


$x+x = 2x, x+(-1) = x-1, (-1)+(-1) = -2, x \cdot x = x^2, (-1) \cdot x = -x, (-1) \cdot (-1) = 1$

Lista pierwszych wyrażeń:

$x, -1, 2x, x-1, -2, x^2, -x, 1, 3x, 2x-1, 4x, 2x-1, x-2, 3x-1, 2(x-1), x-2, -3, 2(x-1), x-3,$

Elementarna algebra wg. Tarskiego



Rozszerzenie na inne wyrażenia matematyczne

- Zbiorem generowanym z $-1, x, +, \cdot$ jest zbiór wszystkich wielomianów o współczynnikach całkowitych.
- Z konstrucji wynika, że jest to zbiór przeliczalny.
- Algorytm rekurencyjny pozwala na enumerację wszystkich takich wielomianów (z powtórzeniami).
- Jak rozszerzyć pomysł na **„dowolne”** funkcje/wyrażenia matematyczne?

Najczęściej stosowany pomysł, to użycie **„wszystkich”** „istotnych” operacji arytmetycznych, algebraicznych i funkcji „elementarnych”, t.j:

$$\ldots -2, -1, 0, 1, \frac{1}{2}, 2, \frac{3}{2} \ldots, \pi, e, i, \phi, \sqrt{2}, \ln 2, \frac{1}{137}, \gamma \ldots,$$

$$+, -, \cdot, \frac{x}{y}, x^y, \sqrt{}, \exp, \ln, \sin, \cos, \mathrm{tg}, arcsin, \sinh, \cosh, \mathrm{tgh}, arsinh, \ldots$$

Pod pojęciem „istotne” można znaleźć m.in:

1. funkcje exp-log (operacje arytmetyczne $+$ $\exp, \ln$)
2. funkcje i operatory dostępne w pewnym języku programowania (C, Mathematica, ...)
3. dostępne liczby całkowite, wymierne, zespolone a nawet algebraiczne + podstawowe stałe przestępne ($\pi, \ldots$)
4. funkcje i liczby uważane za szczególnie „produktywne”: złoty podział $\phi = \frac{\sqrt{5}+1}{2}$, funkcja gamma, liczby pierwsze, wielomiany ortogonalne, itp.

Problemy generacji funkcji elementarnych

Praktyka pokazuje, że zbiorem symboli bazowych nie można manipulować w zupełnie dowolny sposób. Jest to główne źródło niepowodzeń i kiepskiej/losowej skuteczności istniejących metod regresji symbolicznej.

1. niekompletność generowanego zbioru z powodu złego wybrania symboli startowych
   Przykłady:
   - czy dysponując symbolami $+, \exp, \log, i$ wygenerujemy π ?
   - czy dysponując symbolami $+, \times, \wedge, \ln, e$ wygenerujemy -1 ?
2. duża redundancja np: $\frac{1+x}{1-x} = e^{2\,\mathrm{tgh}^{-1}\,x}$,$\cos(3 \arccos x) = 4x^3 - 3x$, $e^{ix} = \cos x + i \sin x$, itp. itd.
3. głębokość drzewa *vs* liczba „instrukcji" (2^{32} liczb całkowitych !?!?)

Konieczne jest wydzielenie spośród zbioru „wszystkich" symboli tych, których nie wolno usunąć, pod groźbą pominięcia niektórych z wyrażeń.

„Broken calculator problem"

Najlepszą metodą aby wyselekcjonować kluczowe symbole jest . . . regresja symboliczna.

Przykłady minimalnego zbioru funkcji/operatorów

Zbiór minimalny „botton-up"

1. dodawanie $+$, $x + y$
2. f. wykładnicza $\exp$, e^x
3. logarytm naturalny $\ln$, lnx
4. odwrotność (brak symbolu) $\frac{1}{x}$

Zbiór minimalny „up-bottom"

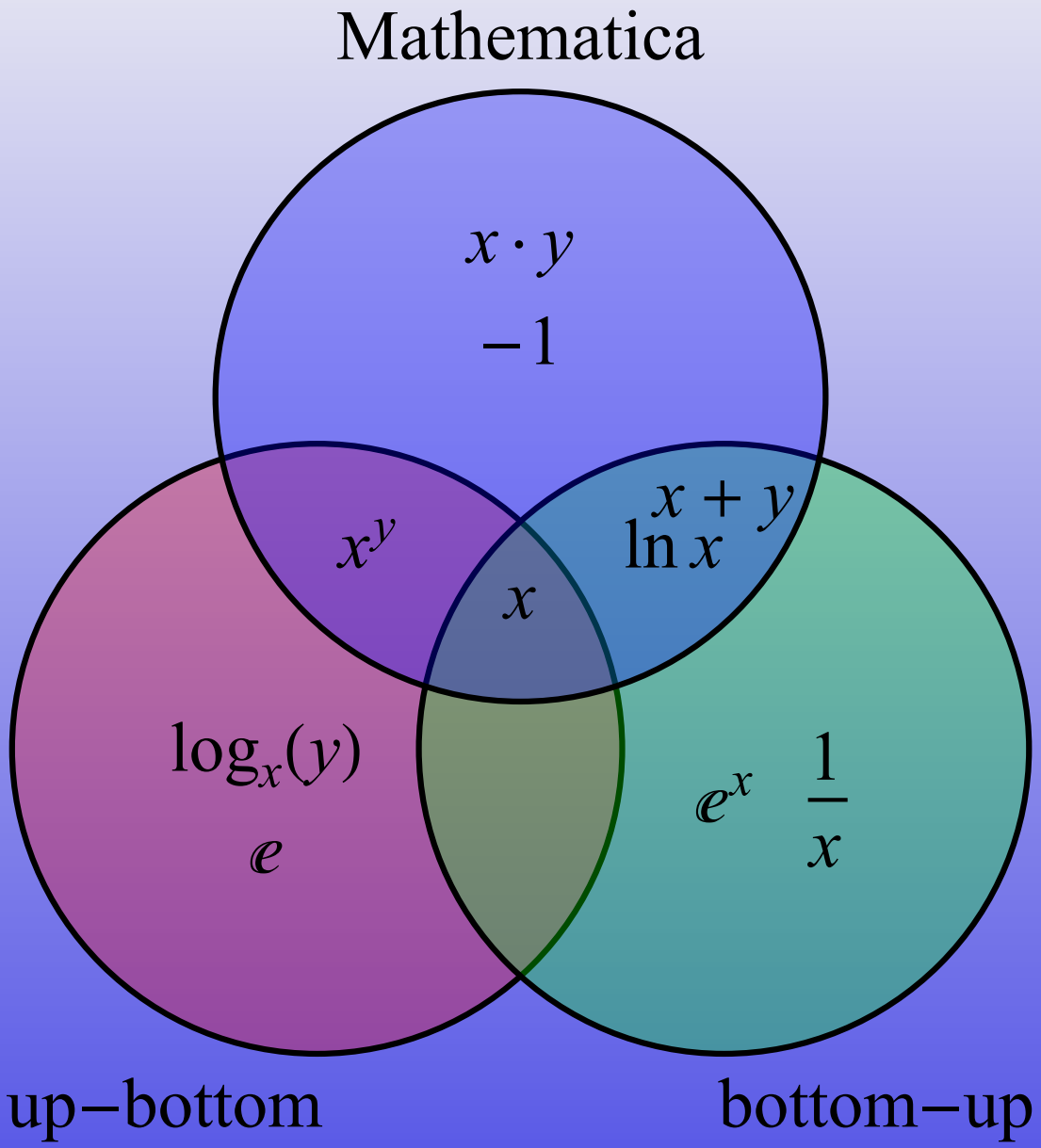
1. potęgowanie $**$, x^y
2. logarytm dwuargumentowy (o dowolnej podstawie) $\log_x y$
3. liczba E (podstawa logarytmu naturalnego) lub π

Dla porównania: *Mathematica*

1. potęgowanie **Power**, mnożenie **Times**, dodawanie **Plus** (wieloargumentowe!)
2. logarytm naturalny **Log**
3. liczba **-1** oraz spory zestaw redundantnych stałych przestępnych (**E**, **Pi**, ...)
4. liczby całkowite **Integer**, wymierne **Rational**, zespolone **Complex**, algebraiczne **Root**
5. „skróty" typu **Sin**, **ArcTan**, **LegendereP** ... rozwijane przez **FunctionExpand**, **TrigToExp**, ...

Powyższe pozwala generować liczby. Aby generować funkcje, trzeba dodać zmienne i parametry $x, y, z, \ldots a, b, c \ldots$. Ewentualnie dodać stałe, np: $\pi, 2, \sqrt{2}, \ln 2, \ldots$.

Minimalne zbiory funkcji i operatorów podstawowych



Pierwsze funkcje w bazie $+, \ln, \exp, 1/x, x$

Pierwsze funkcje

$$x, e^x, \frac{1}{x}, \ln x, e^{e^x}, e^{\frac{1}{x}}, e^{-x}, \frac{1}{\ln x}, -\ln x, \ln(\ln x), 2x, e^{e^{e^x}}, e^{e^{\frac{1}{x}}}, e^{e^{-x}}, e^{\frac{1}{\ln x}}, e^{2x}, e^{-e^x}, e^{-1/x}$$

Pierwsze funkcje stałe (liczby)

$$0, 1, -\ln 2, \ln 2, e^e, \ldots$$

Reprezentacja operacji arytmetycznych

$$x \cdot y = e^{\ln x + \ln y}, \qquad x^y = \exp\left(e^{\ln y + \ln \ln x}\right)$$

Pierwsze funkcje w bazie $x^y, \log_x y, e, x$

Pierwsze funkcje

$$x, e, 1, \frac{1}{\ln x}, \ln x, x^x, x^e, e^x, e^e, 0, \mp\frac{\ln(\ln x)}{\ln x}, \frac{x}{\ln x}, \frac{e}{\ln x}, \mp\ln(\ln x), x\ln x, e\ln x, \infty, \ldots$$

Pierwsze funkcje stałe (liczby)

$$e, 1, e^e, 0, \frac{1}{e}, e^{e^e}, e^{e^2}, -1, e^2, e^{-e}, \frac{1}{e^2}, e^{\frac{1}{e}}, e^{e^{e^e}}, e^{e^{e^2}}, e^{e^{1+e}}, e^{e^3}, 2, -e, -2, e^{1+e}, e^3, i\pi$$

$$i = e^{\log_a b}, \quad a = \ln (e^e)^e = e^2, b = \ln \log_{e^e} e = -1$$

Reprezentacja operacji arytmetycznych

$$x \cdot y = \log_x ((x^x)^y), \qquad x + y =$$

Hierarchia operatorów: top-bottom czy bottom-up ?

Pokazano, że zbiór symboli startowych i operacji jest w zasadzie dowolny, jeżeli zawiera zbiór minimalny.

1. bramka logiczna NAND
2. zeracja (sukcesor) $x + 1 + 1 + 1 + 1 + 1 + \ldots + 1 + 1 = x + n$
3. dodawanie/odejmowanie $x + x + x + x + \ldots + x + x + x = nx$
4. mnożenie/dzielenie $x \cdot x \cdot x \cdot \ldots \cdot x = x^n$
5. potęgowanie/logarytm $x^{x^{\cdot^{\cdot^{x}}}} =^n x$
6. tetracja/superlogarytm/superpierwiastek

Dla teoretyka informatyki cyfrowej operacje logiczne i dodawanie są pierwotne, dla fizyka czy cybernetyka: potęgowanie i logarytmy.

- Operatory wyższe niż potęgowanie są nadal słabo zbadane
- Nie ma zgody co do kontynuacji analitycznej na płaszczyźnie liczb zespolonych
- Nie wiadomo np: czy ${}^4\pi$ jest liczbą naturalną !

Potencjał generowania $\pi, e, \ldots$ lub/i funkcji specjalnych, liczb algebraicznych itp?

Rozpoznawanie stałych

Najpopularniejszym aktualnie zastosowaniem regresji symbolicznej jest dopasowanie wzoru analitycznego do zadanej liczby zmiennoprzecinkowej.

1. Maple (identify)
2. Mathematica (FindFormula, FindSequenceFunction, RootApproximant), WolframAlpha
3. Inverse Symbolic Calculator (https://isc.carma.newcastle.edu.au/), PSLQ
4. RIES (http://mrob.com/pub/ries/)
5. nsimplify (SymPy)
6. A.O. (float - double - arb. prec - symbolic)

Spóbujmy „obliczyć" kilka całek oznaczonych:

$$\int_0^{\pi/2} \ln \sin x \; dx, \qquad \int_0^1 \frac{arctg\,\sqrt{x^2+2}}{(x^2+1)\sqrt{x^2+2}} dx, \qquad \int_0^1 \left(\frac{1}{\ln x} + \frac{1}{1-x}\right)^2 dx,$$

$$\int_{-1}^{1} \frac{1}{x}\sqrt{\frac{1+x}{1-x}} \ln\left(\frac{2\,x^2+2\,x+1}{2\,x^2-2\,x+1}\right) \; \mathrm{d}x$$

	WolframAlpha	identify	ISC	RIES
$K = \int_0^{\pi/2} \ln \sin x \; dx$ $= -2.17758609$	none	$-\pi \ln(2)$	-ln(2)/ln(exp(1/Pi))	$-K/\pi = \ln 2$ $\ln(2^{-\pi})$
$K = \int_0^1 \frac{arctg\sqrt{x^2+2}}{(x^2+1)\sqrt{x^2+2}} dx$ $= 0.51404189589$	$\frac{5\zeta(2)}{16}$	$\frac{5}{96}\pi^2$	$\frac{5\pi^2}{96}$	$x = \left(\frac{\pi}{4}\right)^2 - \frac{x}{5}$
$K = \int_0^1 \left(\frac{1}{\ln x} + \frac{1}{1-x}\right)^2 dx$ $= 0.3378770664$	$\frac{857261070\ \pi}{7970843089}$	—	K+2 = 1/2+ln(2)+ln(Pi)	$K = \ln(2\pi) - \frac{3}{2}$
$\int_{-1}^1 \frac{1}{x}\sqrt{\frac{1+x}{1-x}} \ln\left(\frac{2x^2+2x+1}{2x^2-2x+1}\right)$ $= 8.37221162660127566162574 7121$	matem. bełkot	—	Wow, really found nothing	$\cos\left(\frac{K}{\pi}\right) + 1 = \frac{2}{\phi^6}$

Rozszerzenie algorytmu na funkcje jednej zmiennej

Niemal identycznie można „zgadywać" dowolne funkcje jednej (lub więcej) zmiennej możliwe do uzyskania w postaci numerycznej. Pochodzenie danych jest nieistotne, może to być:

- całka oznaczona z parametrem lub ruchomą granicą,
- rozwiązanie r.r. zwyczajnego
- rozwiązanie r. algebraicznego z parametrem
- dane eksperymentalne
- ...

Fitowanie z rozpoznawaniem stałych

Generowane funkcje mózg z łatwością klasyfikuje jako należące do oczywistych klas równoważności, np:

$$x, 2x, x+e, x^2, ex, x+1, x+e^e, 3x, 2x+e, 2x+1, 2x+e^e, 4x, x+2e, x+e+1, x+e^e+e, 3x+e,$$

to wielomiany drugiego stopnia postaci:

$$ax^2 + bx + c.$$

Naturalnym jest pomysł, aby generować funkcje jednej zmiennej x, kilku parametrów $a, b, c, \ldots$, znaleźć numeryczne wartości a, b, c standardową regresją nieliniową, a następnie użyć algorytmu identyfikacji stałych $a, b, c, \ldots$

Warunek niezależności parametrów $a, b, c, \ldots$

Wrońskian pochodnych funkcji $f(x; a, b, c, \ldots)$ względem parametrów $a, b, c, \ldots$ musi być różny od zera.

Permutacje parametrów $a, b, c, \ldots$

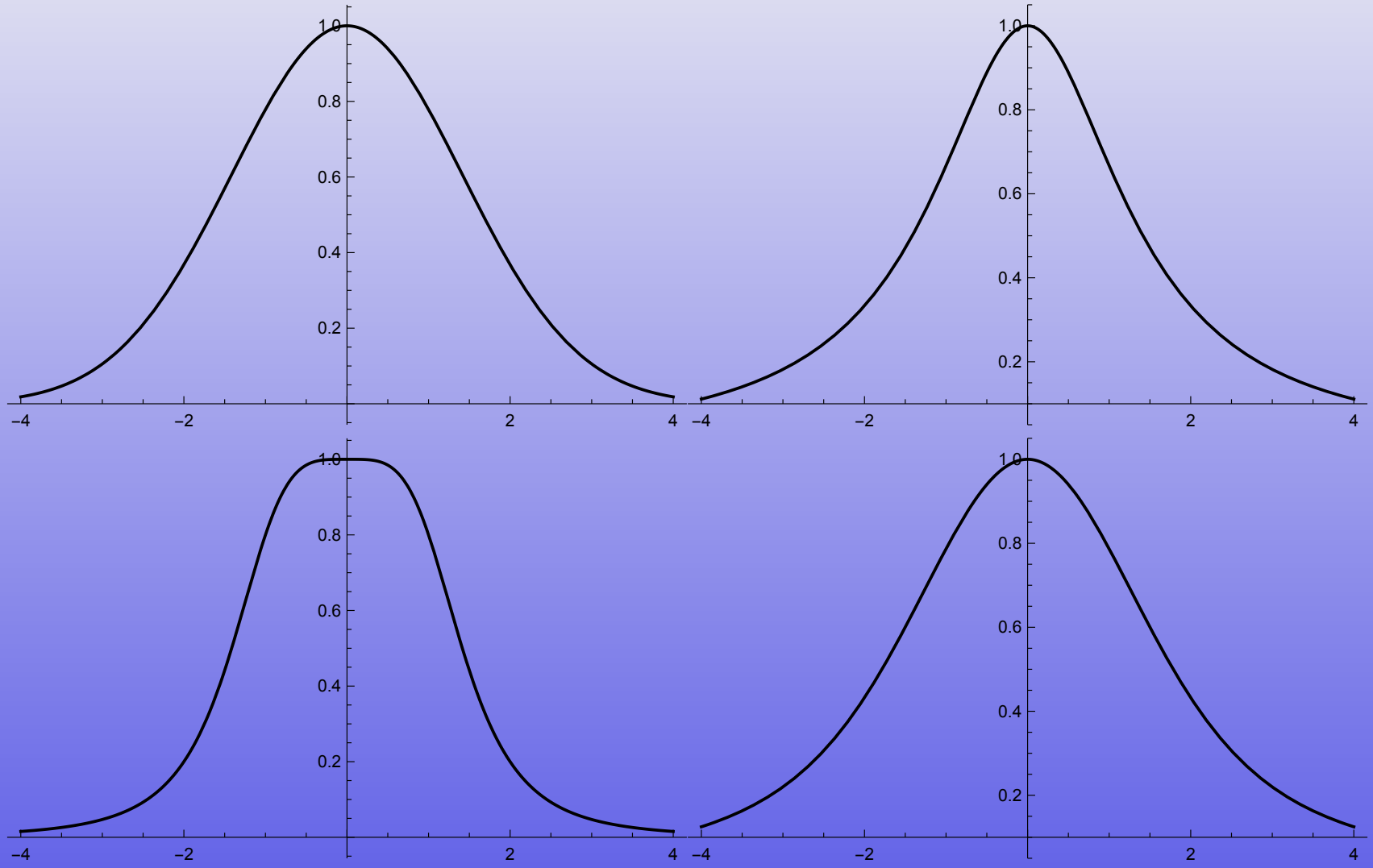
Funkcje różniące się wyłącznie nazwami parametrów należą do tej samej klasy, np:

$$ax^2 + bx + c, bx^2 + ax + c, cx^2 + bx + a, \ldots$$

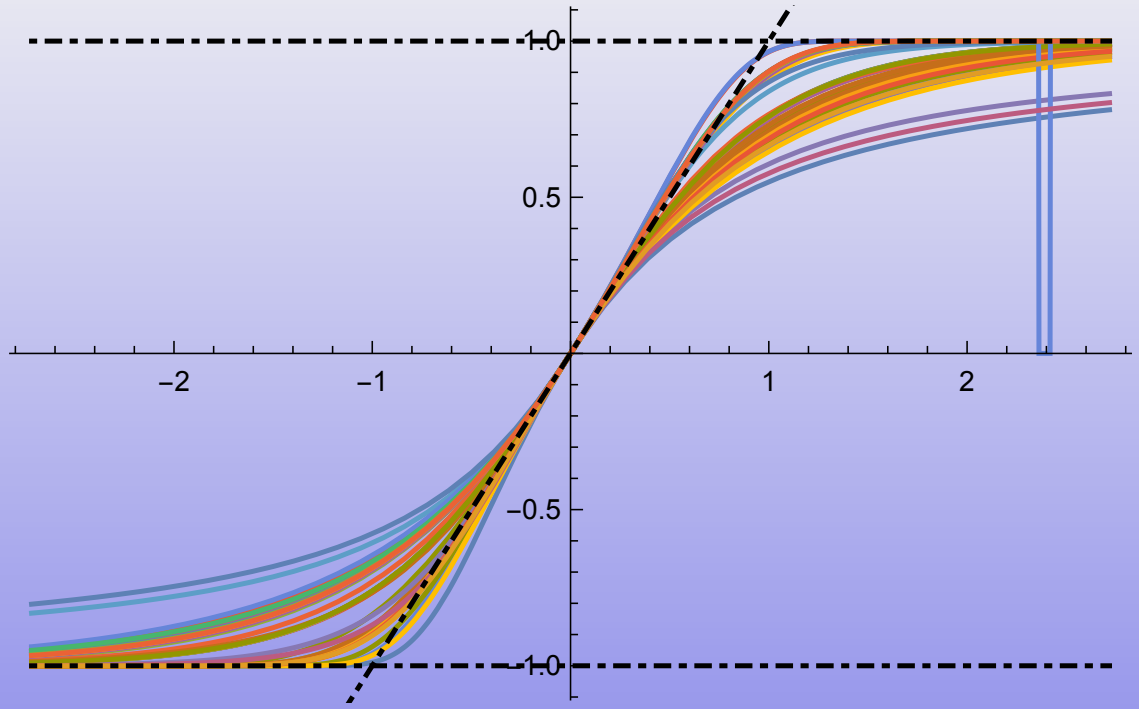
Zastosowania

1. szukanie rozwiązań symbolicznych o małej złożoności dla dowolnych sprawdzalnych problemów np: całkowanie oznaczone i nieoznaczone, r. r. zwyczajne i cząstkowe, r. funkcyjnych i opóźnionych
2. generowanie losowych funkcji, np: w celach dydaktycznych lub testowych
3. analiza kształtu impulsu w fizyce eksperymentalnej
4. liczbowe określenie skomplikowania funkcji
5. katalogowanie i wyszukiwanie funkcji, np: wzór nr. 1236378347 zamiast $\frac{\ln x}{x^2+1}$ (rozwinięcie idei baz ciągów OEIS)

Jak rozpoznać/rozróżnić skomplikowane krzywe?



Generacja funkcji sigmoidalnych



$$\mathrm{sgm}(x) = \mathrm{tgh}\, x, 2^{1-e^{-\frac{x}{\ln 2}}} - 1, 1 - 2^{1-e^{\frac{x}{\ln 2}}}, \frac{6}{4e^{-\frac{3x}{\ln 16}} + 2} - 1, \frac{4}{3e^{-\frac{4x}{\ln 27}} + 1} - 1$$

$$\frac{2}{\ln\left((e-1)e^{-\frac{2ex}{e-1}} + 1\right) + 1} - 1, \frac{\ln 4}{\ln\left(2^{1-4x} + 2\right)} - 1, 1 - \frac{\ln 4}{\ln\left(2\left(16^x + 1\right)\right)}, 1 - \frac{6}{4e^{\frac{3x}{\ln 16}} + 2}$$

Impuls „wzorcowy”

$$y_0 + h\ \mathrm{sgm}\left(\frac{t - t_0}{T}\right)$$

Uwagi końcowe: dlaczego brutalna siła nie działa lepiej niż umysł?

- bariera psychologiczna: oczekujemy „wyjaśnienia” nawet gdy wynik łatwo znaleźć metodą „zgadywania”
- nauka jako „superalgorytm” o własności typu P=NP
- komputery ciągle zbyt wolne

Otwarte pytania/hipotezy

- Jaka jest rola fukcji odwrotnych do siebie?
- Czy zbiór trzech operatorów: tetracja, superlog, superoot jest zbiorem minimalnym? Czy π lub/i e są ich kombinacjami?
- Czy istnieje sposób enumeracji wyrażeń matematycznych poprzez składanie funkcji, jak dla liczb wymiernych?
- Czy ilość klas równoważności funkcji z parametrami jest nieskończona? Jeżeli tak, to jak szybko rośnie ich liczba?
- Czy kiedykolwiek algorytm tego typu wyprzedzi naukowców?
- Czy kiedykolwiek uda mi się skończyć własny program do regresji symbolicznej!?

Ogura, kill the men that get past me.

Referencje

A. Tarski, A Decision Method for Elementary Algebra and Geometry (RAND, 1948)
http://www.rand.org/content/dam/rand/pubs/reports/2008/R109.pdf

Metody regresji symbolicznej

Enumeracja (brute-force search):

1. rekurencyjne generowanie drzew
2. sekwencyjne generowanie drzew
3. maszyna stosowa
4. składanie funkcji
5. concatenative programming language

Inne metody:

1. fitowanie z rozpoznawaniem stałych
2. metody genetyczne
3. gramatyka generatywna

Regresja tradycyjna versus symboliczna

Zakładamy, że są dostępne pewne dane liczbowe, pochodzenia numerycznego lub eksperymentalnego. Można o nich myśleć jako fukncji jednej zmiennej rzeczywistej zadanej w postaci tabelki.

W idealnej sytuacji dane są możliwe do uzyskania w dowolnej ilości i z dowolnie dużą dokładnością.

Regresja tradycyjna

- funkcja zadana z góry
- parametry szukane w postaci numerycznej
- złożoność modelu nie jest brana pod uwagę

Regresja symboliczna

- funkcja szukana równocześnie z parametrami
- parametry szukane docelowo w postaci symbolicznej (docelowo zero, „wchłonięte przez" definicję funkcji)
- złożoność funkcji jednym z kryteriów dopasowania