Zariski-Dense Dataset Generation for Learning to Compute Gröbner Bases

Yuta Kambe (Mitsubishi Electric)

07/2026

SCML 2026

Today

- Introduction
  - **Approximation of Input-Output Map**
- General Question on Dataset Generating
  - **How general is the generated dataset?**
- Open Problem: Correctness of the Heuristic

Introduction

The Rise of AI for Math

AI for Math

AI for Proof

- Solving Erdős Problems
- Solving Open Problems
- LEAN × LLM：
  - Formal verification,
  - Automated proof search

AI for Computation

- Algorithm Search
- Learned Numerical Computation
- **Learned Symbolic Co**
- **Computing Gröbner bases**

Symbolic Computation × Machine Learning

Approaches in SCML

SCML

Accelerating Implementation

- Optimization of Selecting
  - Monomial Order
  - S-pair
- Making Oracles

Approximate model of Algorithms

- **Approximation of Input-Output Map**

My Interests

Example of Approximation of I/O map

Problem: for given a function f, determine $F(x) = \int f(x)dx$

f **Relatively Difficult** $\longrightarrow$ F

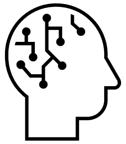
Given Function

Primitive Function

$\Longleftarrow$ **Relatively Easy** to Comp. $f = \frac{d}{dx}F$

$\blacktriangledown$

$\left\{\left(f_1, \int f_1\, dx\right), \left(f_2, \int f_2\, dx\right), \dots, \left(f_N, \int f_N\, dx\right)\right\}, \boldsymbol{N = 4 \times 10^7}$ $\longrightarrow$ Learning



$\longrightarrow$ Model of Integration

Example of Approximation of I/O map

Problem: for given a function f, determine $F(x) = \int f(x)dx$

$$f \xrightarrow{\text{Relatively Difficult}} L$$

Given Function

Primitive Function

Relatively Easy to

Comp. $f = \dfrac{d}{dx}F$

$$\left\{\left(f_1, \int f_1\, dx\right), \left(f_2, \int f_2\, dx\right), \dots, \left(f_N, \int f_N\, dx\right)\right\}, \boldsymbol{N = 4 \times 10^7} \xrightarrow{\text{Learning}} \longrightarrow \text{Model of Integration}$$

Lample-Charton(2019):

Guillaume Lample and Francois Charton, "Deep Learning for Symbolic Mathematics", ICLR2020

By learning transformer from 4×10^7 problem/answer pairs, they found instances that **SymPy cannot solve but a Neural Network Model can**

AI for Unsolved Problems

(Unsolved) Given a dynamical system F, does there exist

a **Lyapunov function** $\boldsymbol{L}$ for F? If so, how can it be constructed?

F

Dynamical System

? Still unsolved but...

We can construct **reverse algorithms** $\boldsymbol{L} \longmapsto \boldsymbol{F}$

L

Lyapunov Function

Alfarano-Charton-Hayat (2024): Alberto Alfarano, François Charton, Amaury Hayat, "*Global Lyapunov functions: a long-standing open problem in mathematics, with symbolic transformers*", NeurIPS2024

By making a reverse algorithm, **they constructed a training dataset** $(\boldsymbol{F_\lambda}, \boldsymbol{L_\lambda})$**,**

a NN model of $F \mapsto L$, and **<u>new Lyapunov functions for untrained systems</u>**

AI for fast computing Gröbner bases

$$F = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} \longrightarrow G = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

exponential time

Deterministic Algorithm
(Buchberger, F4, F5, ...)

Polynomial system

Polynomial time

Gröbner basis

Kera-Vaccon-Ishihara-**K**.-Yokoyama (2024): Hiroshi Kera, Tristan Vaccon, Yuki Ishihara, Yuta Kambe, Kazuhiro Yokoyama, "Learning to Compute Gröbner Bases", NeurIPS2024

By making a reverse algorithm, they constructed a training dataset (F_λ, G_λ), a NN model of $F \mapsto G$, and **find examples that the model is over 100 times faster than Buchberger's algorithm**

General Question
on Dataset Generating

Question on Dataset Generating

$$f \underset{\gamma\text{: Relatively Easy}}{\overset{\sigma\text{: Relatively Difficult}}{\rightleftarrows}} F$$

Input — Output

$$\begin{aligned}\mathcal{D} &= \{(\gamma(F_1), F_1), (\gamma(F_2), F_2), \dots, (\gamma(F_N), F_N)\} \\ &= \{(f_1, \sigma(f_1)), (f_2, \sigma(f_2)), \dots, (f_N, \sigma(f_N))\} \subset \mathrm{Graph}(\sigma)\end{aligned}$$

Question: How general is this dataset $\mathcal{D}$?

My Main Problem

$$\underset{\text{Input}}{f} \quad \underset{\gamma\text{:Relatively Easy}}{\overset{\sigma\text{:Relatively Difficult}}{\rightleftarrows}} \quad \underset{\text{Output}}{F}$$

$$\mathcal{D} = \{(\gamma(F_1), F_1), (\gamma(F_2), F_2), \dots, (\gamma(F_N), F_N)\}$$
$$= \{(f_1, \sigma(f_1)), (f_2, \sigma(f_2)), \dots, (f_N, \sigma(f_N))\} \subset \mathrm{Graph}(\sigma)$$

Question: How general is this dataset $\mathcal{D}$?

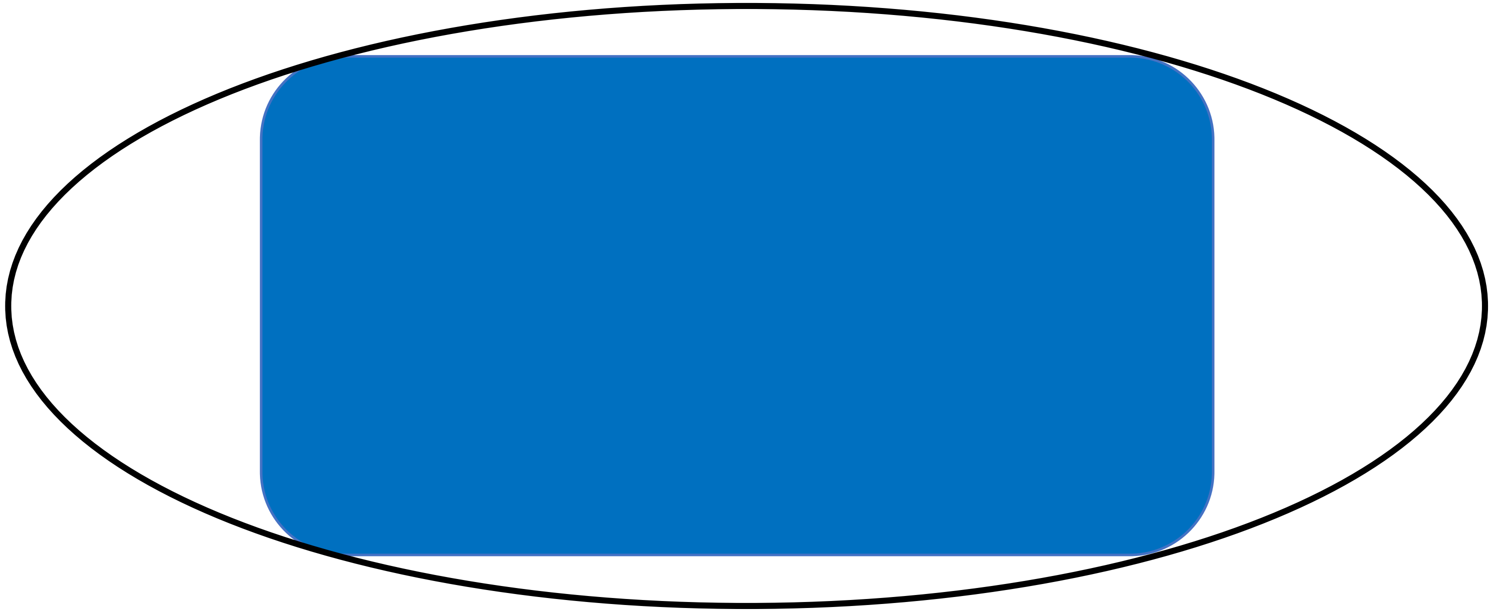
Problem：For given γ,

is the image of γ Zariski dense in each fiber of σ?

Recent Results on Gröbner bases Dataset

Theorem(Kambe-Maeda-Vaccon, 2025): Assume $\boldsymbol{K} = \mathbb{Q}$ + **Heuristic**

$$\{F \in R^{m\times 1} \mid \langle F\rangle = \langle G\rangle\}$$



The image of the reverse algorithm
is **dense** in $\{F \in R^{m\times 1} \mid \langle F\rangle = \langle G\rangle\}$

Yuta Kambe, Yota Maeda, Tristan Vaccon, "*Geometric Generality of Transformer-Based Gröbner Basis Computation*", to appear in *Artificial Intelligence and Mathematical Research*, Contemporary Mathematics, AMS

Accuracy of G.B./Accuracy of terms. of G.B.

Experiments by Kera et al.: Table of accuracies of **(Coeffs+Terms)/Only Terms**

- Accuracies over $\mathbb{Q}$ are **relatively high**, but
- accuracies over $\mathbb{F}_p$ are **relatively low**

Ring	$\prec_{\text{lex}}$				$\prec_{\text{grvlex}}$
	$n = 2, \sigma = 1.0$	$n = 3, \sigma = 0.6$	$n = 4, \sigma = 0.3$	$n = 5, \sigma = 0.2$	$n = 2, \sigma = 1.0$
$\mathbb{Q}[x_1, \ldots, x_n]$	92.6 / 96.3	94.7 / 97.4	92.7 / 97.8	89.3 / 94.9	–
$\mathbb{F}_7[x_1, \ldots, x_n]$	68.2 / 77.3	65.1 / 86.6	76.4 / 87.2	80.9 / 90.5	3.1 / 14.7
$\mathbb{F}_{31}[x_1, \ldots, x_n]$	45.5 / 79.7	60.1 / 89.9	72.2 / 88.2	78.5 / 92.3	5.8 / 45.3

Correctness of the Heuristic

Two sets of generators of an ideal

Notation

- K is a field
- $R = K[x_1, x_2, \dots, x_r]$ is the polynomial ring
- Let us consider poly. sys. F as a vector in $R^{m\times 1}$:

$$F = (f_1, f_2, \dots, f_m)^T = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} \in R^{m\times 1}$$

Proposition Let $F \in R^{m\times 1}, G \in R^{n\times 1}$.

$$\langle F\rangle = \langle G\rangle \Rightarrow \exists\ A \in R^{m\times n} \text{ s.t. } F = AG$$

Proposition Let $A \in R^{m\times n}, G \in R^{n\times 1}$.

$$\exists\ B \in R^{n\times m} \text{ s.t. } BA = E_n \Rightarrow \langle AG\rangle = \langle G\rangle$$

Remark $\langle AG\rangle = \langle G\rangle \Leftrightarrow \exists B \in R^{n\times m}\ s.t.(BA - I_n)G = O$

Left regular matrix

Proposition Let $A \in R^{m\times n}, G \in R^{n\times 1}$.

$$\exists\ B \in R^{n\times m} \text{ s.t. } BA = E_n \Rightarrow \langle AG\rangle = \langle G\rangle$$

We say A is a ***left regular matrix*** over R if $\exists\ B \in R^{n\times m}$ s.t. $BA = E_n$

Proposition (Kambe-Maeda-Vaccon, 2025)
Assume that $\boldsymbol{m \geq n \geq 3}$,

$$A \in R^{m\times n} \text{ is } \textbf{left regular} \Leftrightarrow \exists U \in \boldsymbol{E(m)} \text{ such that } \boldsymbol{A = U\begin{pmatrix} I_n \\ O_{(m-n)\times n}\end{pmatrix}},$$

where

$$I_n \coloneqq \text{identity matrix of size } n,$$
$$O_{(m-n)\times n} \coloneqq \text{zero matrix of size } (m-n)\times n$$
$$\boldsymbol{E(m) \coloneqq \langle GL(K^m), SL(R^m)\rangle}$$

The reverse algorithm for G.B.

$$F = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} \xleftarrow{\quad \textbf{Reverse Algorithm} \quad} G = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

Polynomial system **Gröbner basis**

<u>**Reverse Algorithm**</u>

1. Pick a random G.B. G (shape position, Cauchy module, etc….)
2. Pick a random left regular matrix $A \in R^{m \times n}$
3. Return $F = AG$

Evaluating generality of outputs of the algo.

Theorem(Kambe-Maeda-Vaccon, 2025): Fix m, n with $m > n$. Assume $K = \mathbb{Q}$.

Let $G \in R^{n\times 1}$. Let

$$\mathcal{F} = \{F \in R^{m\times 1} \mid \langle F \rangle = \langle G \rangle\} = \{AG \mid \exists B \in R^{n\times m}\ (BA - E_n)G = O\}$$

$$\boldsymbol{\mathcal{F}_0 = \{ AG \mid A \in R^{m\times n}: \textbf{left regular} \}} \subset \mathcal{F}.$$

We equip a Zariski topology on $\mathcal{F}$ from identification

$$R = \bigcup_{D=0}^{\infty} R_{\leq D} = \bigcup_{D=0}^{\infty} K^{N_D},$$

Kera & Ishihara's team also have some result in the case of $\boldsymbol{m \leq n}$

where $R_{\leq D} = \{ f \in R \mid \deg f = D \}, N_D = \sum_{i=0}^{D} \binom{r+i}{i}$(parameterization on coeffs).

If $m \geq 2n$ and some heuristic is true, then **$\mathcal{F}_0$ is dense in $\mathcal{F}$**.

Our Heuristic

Notation and Definition

- $A = \begin{pmatrix} A_1 \\ \hline A_2 \end{pmatrix} \in R^{m\times n}, A_1 \in R^{n\times n}, A_2 \in R^{(m-n)\times n}$
- $B = (B_1|B_2) \in R^{n\times m}, B_1 \in R^{n\times n}, B_2 \in R^{n\times(m-n)}$
- $\boldsymbol{\mathcal{X}_{\leq D} = \{\,(B, A) \in R_{\leq D}^{n\times m} \times R_{\leq D}^{m\times n} \mid (BA - E_n)G = O, B_1 A_1 \in R_{\leq D}^{n\times n}\,\}}$

Key Theorem (K.-Maeda-Vaccon)

If $m \geq 2n$ and $\boldsymbol{\mathcal{X}_{\leq D}\ (D \gg 0)}$ is irreducible, then $\mathcal{F}_0 \cap \mathcal{F}_{\leq D}$ is dense in $\mathcal{F}_{\leq D}$

Heuristic

Is $\mathcal{X}_{\le D}$ irreducible?

Notation and Definition

- $A = \begin{pmatrix} A_1 \\ \hline A_2 \end{pmatrix} \in R^{m\times n}, A_1 \in R^{n\times n}, A_2 \in R^{(m-n)\times n}$
- $B = (B_1 | B_2) \in R^{n\times m}, B_1 \in R^{n\times n}, B_2 \in R^{n\times(m-n)}$

$$\boldsymbol{\mathcal{X}_{\le D} = \{\,(B, A) \in R_{\le D}^{n\times m} \times R_{\le D}^{m\times n} \mid (BA - E_n)G = O, B_1 A_1 \in R_{\le D}^{n\times n}\,\}}$$

Our Heuristic

$\boldsymbol{\mathcal{X}_{\le D}}$ $(\boldsymbol{D \gg 0})$ is irreducible

- **Question: Is $\mathcal{X}_{\le D}$ irreducible for generic G and $D \gg 0$?**
- We **do not have any logical guarantee** of this heuristic
- Intuitively,

$$G \text{ is generic} \Rightarrow \mathcal{X}_{\le D} \text{ is a generic variety} \Rightarrow \mathcal{X}_{\le D} \text{ is irreducible}$$

- **Idea: Using syzygy of $\boldsymbol{G}$, reseach the generality/irreducibility of $\boldsymbol{\mathcal{X}_{\le D}}$**