Journal of Symbolic Computation 115 (2023) 478–491

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Machine learning invariants of arithmetic curves

Yang-Hui He [a,b,c], Kyu-Hwan Lee [d], Thomas Oliver [e]

[a] London Institute, Royal Institution, 21 Albemarle St, London W1S 4BS, UK
[b] Merton College, University of Oxford, OX14JD, UK
[c] School of Physics, NanKai University, Tianjin, 300071, PR China
[d] Department of Mathematics, University of Connecticut, Storrs, CT, 06269-1009, USA
[e] SCEDT, Teesside University, Middlesbrough, TS1 3BX, UK

ARTICLE INFO

Article history:
Received 7 December 2021
Received in revised form 29 June 2022
Accepted 15 August 2022
Available online 22 August 2022

Keywords:
Machine-learning
Arithmetic geometry
Elliptic curves
Hyper-elliptic curves
Birch-Swinnerton-Dyer conjecture

ABSTRACT

We show that standard machine learning algorithms may be trained to predict certain invariants of low genus arithmetic curves. Using datasets of size around 10^5, we demonstrate the utility of machine learning in classification problems pertaining to the BSD invariants of an elliptic curve (including its rank and torsion subgroup), and the analogous invariants of a genus 2 curve. Our results show that a trained machine can efficiently classify curves according to these invariants with high accuracies (> 0.97). For problems such as distinguishing between torsion orders, and the recognition of integral points, the accuracies can reach 0.998.

Contents

E-mail addresses: hey@maths.ox.ac.uk (Y.-H. He), khlee@math.uconn.edu (K.-H. Lee), T.Oliver@tees.ac.uk (T. Oliver).

https://doi.org/10.1016/j.jsc.2022.08.017

1. Introduction

In this article we build on recent work by the present authors, namely: He et al. (2022a,b). In the latter, we presented experiments demonstrating the capacity of machine learning to predict basic invariants of algebraic number fields. Many of the invariants studied there appear together in the analytic class number formula:

$$\lim_{s\to 1}(s-1)\zeta_F(s) = \frac{2^{r_1}(2\pi)^{r_2}\,\mathrm{Reg}_F\,h_F}{w_F\sqrt{|\Delta_F|}}\,, \tag{1.1}$$

in which $\zeta_F(s)$ is the Dedekind zeta function of a number field F, and, (r_1, r_2) is the signature, Reg_F is the regulator, h_F is the class number, Δ_F is the discriminant, and w_F is the number of roots of unity in $\mathcal{O}_F$.

Recall that the set $E(\mathbb{Q})$ of rational points on an elliptic curve E defined over $\mathbb{Q}$ defines a finitely generated abelian group. We will denote the rank of this group by r, and the torsion subgroup by $E(\mathbb{Q})_{\mathrm{tors}}$. Associated to an elliptic curve E over $\mathbb{Q}$, one has the L-function $L(E,s)$ which conjecturally vanishes to order r at $s=1$ with the leading Taylor coefficient given by the following equation analogous to (1.1):

$$\frac{L^{(r)}(E,1)}{r!} \stackrel{?}{=} \frac{\#Ш(E/\mathbb{Q})\Omega(E/\mathbb{Q})\mathrm{Reg}(E/\mathbb{Q})\prod_p c_p}{(\#E(\mathbb{Q})_{\mathrm{tors}})^2}, \tag{1.2}$$

in which the symbol $\stackrel{?}{=}$ indicates that the equality is conjectural, $Ш(E/\mathbb{Q})$ is the Tate–Shafarevich group, $\Omega(E/\mathbb{Q})$ is the global period, $\mathrm{Reg}(E/\mathbb{Q})$ is the regulator, c_p is the Tamagawa number at a prime p (cf. Silverman, 1992, Appendix C, Conjecture 16.5 for details).

The Hasse–Weil zeta function of a curve was introduced in the 1950's, and is initially only defined for $\mathrm{Re}(s)$ sufficiently large (Hasse, 1954). Therefore, to be careful, equation (1.2) requires analytic continuation of $L(E,s)$ to a larger domain incorporating the value $s=1$. In fact, much more was established by Wiles and Breuil–Conrad–Diamond–Taylor in the 1990's as a consequence of the Modularity Theorem, which, roughly speaking, states that the Dirichlet coefficients coincide with the Fourier coefficients of a holomorphic modular form on the upper half-plane. Famously, the modularity theorem also enabled Wiles' proof of Fermat's Last Theorem in 1995.

Equation (1.2) is of course the renowned conjecture of Birch and Swinnerton-Dyer (BSD), which emerged in the 1960's and is an open Millennium prize problem. Aside from the BSD conjecture, there are many interesting open questions regarding the numbers appearing in equation (1.2). For example, it is unknown whether or not the Tate–Shafarevich group $Ш(E/\mathbb{Q})$ appearing on the right-hand side is finite, cf. Silverman (1992, Chapter X, Conjecture 4.13), and there is no rigorous method known for its computation. Nevertheless, when $Ш(E/\mathbb{Q})$ is finite, its order is known to be a square (Silverman, 1992, Chapter X, Theorem 4.14). As for the left-hand side, it is not yet known whether or not, as E

varies, the set of ranks r is bounded (a heuristic model suggesting that this set might be bounded was presented in Park et al. (2019)). Furthermore, in a suitable asymptotic sense, it is conjectured that 50% of elliptic curves have rank 0, 50% have rank 1, and 0% have rank ≥ 2 (Goldfeld, 1979, Conjecture B; Katz and Sarnak, 1999a,b). More fundamental problems in the theory of arithmetic curves will be discussed in Section 3.1.

In this paper, we will apply supervised machine learning techniques to predict—with varying levels of success—the invariants appearing in equation (1.2) and its generalisations. The generic supervised learning strategy is reviewed in Section 3.2, a crucial step of which is training the classifier, which typically involves random sampling from our database. The accuracy of the resulting classifier depends on this sample, and so its exact numerical value cannot be replicated. On the other hand, the experimental set-up can be easily reconstructed, and repeated implementation yields classifiers of similar precision.

The classifiers developed are, inevitably, less than 100% accurate. Furthermore, even a perfect classifier on a finite database does not constitute a theorem in the infinite set of elliptic curves. Whilst it is theorems that are the intention of mathematics, we can use a classifier as a heuristic tool and gather insight into new structures through working towards interpreting the classifier. One example of this is documented in Section 4.6 and another is explored in He et al. (under review). As we explore in a future project, insight might also be gained by understanding which curves are consistently misclassified.

With appropriate modifications, it is possible to replace $E/\mathbb{Q}$ in equation (1.2) by the Jacobian of a smooth, projective, geometrically integral curve defined over a global field. In this paper, we also study genus 2 curves over $\mathbb{Q}$. There are some important differences between elliptic curves and those of genus 2. For example, there are known to be only finitely many rational points on a genus 2 curve. In contrast, an elliptic curve with positive rank has infinitely many rational points. On the other hand, the rational points on the Jacobian of a genus 2 curve form a finitely generated abelian group which could be infinite. By the rank of a genus 2 curve, we mean the rank of its Jacobian. For elliptic curves, classification by rank is essentially binary as per the conjecture mentioned above. On the other hand, there is a significant proportion of genus 2 curves with rank 2—thus allowing for a ternary classification.

For a broad introduction to machine learning, see Goodfellow et al. (2016); Hastie et al. (2001). In this paper we will utilise logistic regression, naive Bayes, and random forest classifiers, which are reviewed in Hastie et al. (2001, Sections 4.4, 6.6.3, 15). Machine learning algorithms require the training of classifiers using large sets of data, and we obtain our data sets from The LMFDB Collaboration (2020). Previous elliptic curve machine learning experiments were documented in Alessandretti et al. (in press), as part of a recent programme of machine learning various structures in mathematics (He, 2017, 2019, 2021). The key difference in the present article is that, whilst Alessandretti et al. (in press) utilised Weierstrass coefficients as training data (which had enormous variation in magnitude), we will here lean more heavily on the Euler factors of L-functions. We observe this to be much more successful, allowing even for extrapolation to elliptic curves with conductors in ranges beyond those in the training dataset.

We remark that there is some comparison to be made with He et al. (2022a), in which we studied machine learning of a particular classification problem arising from the Sato–Tate conjecture for hyperelliptic curves. There, we found that naive Bayes classifiers could distinguish, with accuracies into $0.99 \sim 1.00$ range, the Sato–Tate groups of hyperelliptic curves with genus 1 or 2, using a small number of Euler-factors. The experimental results of this paper show that the same method is just as powerful for other invariants. It is interesting that a method as simple as naive Bayes could do so well for various invariants in number theory. It might be suggesting that mathematics is more workable with machine learning than the real world where data sets could be dimmed or distorted by various noises.

We conclude this introduction with an overview of what is to come. In Section 2 notation is fixed and complemented by a brief explanation of some concepts. In Section 3, the generation of training data and the experimental set-up are explained. In Section 4 we document the experimental outcomes for elliptic curves. In Section 5, we do the same for genus 2 curves. Finally, in Section 6, we offer some concluding remarks and tentative directions for further research.

Acknowledgements

We thank Álvaro Lozano-Robledo, Andrew Sutherland and Chris Wuthrich for helpful discussions. YHH is indebted to STFC UK, for grant ST/J00037X/1, KHL is partially supported by a grant from the Simons Foundation (#712100), and TO acknowledges support from the EPSRC through research grant EP/S032460/1.

2. Notation

We use the following notation throughout:

Elliptic curve defined over $\mathbb{Q}$ is denoted by E. The set $E(\mathbb{Q})$ of rational points defines a finitely generated abelian group;

Genus 2 curve defined over $\mathbb{Q}$ is denoted by C. The curve C is assumed to be smooth, projective, and geometrically integral. The set $C(\mathbb{Q})$ of rational points is finite;

Jacobian of C is denoted by J. The Jacobian is a two-dimensional Abelian variety defined over $\mathbb{Q}$. The set $J(\mathbb{Q})$ of rational points defines a finitely generated Abelian group;

Rank of E (resp. C) denoted by r_E (resp. r_C) is the rank of the finitely generated abelian group $E(\mathbb{Q})$ (resp. $J(\mathbb{Q})$). If the rank is 0 (resp. positive), then there are finitely many (resp. infinitely many) rational points;

Torsion subgroup of $E(\mathbb{Q})$ (resp. $J(\mathbb{Q})$) is denoted by $E(\mathbb{Q})_{\text{tors}}$ (resp. $J(\mathbb{Q})_{\text{tors}}$);

Cyclic Group of order n is denoted C_n. The torsion subgroup of $E(\mathbb{Q})$ is a product of cyclic groups;

Good primes of a variety X defined over $\mathbb{Q}$ are those primes $p \in \mathbb{Z}$ such that X has an integral model whose reduction modulo p defines a smooth variety of the same dimension. A good prime for C is a good prime for J, but the converse is not necessarily true;

Bad primes of a variety X defined over $\mathbb{Q}$ are those primes $p \in \mathbb{Z}$ which are not good. The bad reduction types of an elliptic curve are reviewed in Silverman (1992, Section VII.5);

Conductor of E (resp. C) denoted by Q_E (resp. Q_C) is a positive integer of the form $\prod p^{e_p}$ in which p varies over the bad primes for E (resp. J). The power e_p to which a bad prime p appears depends on the reduction type, cf. Silverman (1994, Section IV.10);

Tate–Shafarevich group of E (resp. J) denoted by $Ш(E/\mathbb{Q})$ (resp. $Ш(J/\mathbb{Q})$) is a torsion Abelian group and measures the extent to which the Hasse principle fails to hold, cf. Silverman (1992, Section X.4).

3. Methodology

In this section we explain our experimental set-up. In particular, we construct the training and validation sets from appropriate data and overview the machine learning strategies used.

3.1. Euler factors

Let X be a smooth, projective, geometrically connected curve of genus $g \in \{1, 2\}$, defined over $\mathbb{Q}$. For each good prime p of X, we define the local zeta function to be:

$$Z(X/\mathbb{F}_p; T) = \exp\left(\sum_{k=1}^{\infty} \frac{\#X\left(\mathbb{F}_{p^k}\right) T^k}{k}\right). \tag{3.3}$$

It is well-known that the local zeta function can be written in the form

$$Z(X/\mathbb{F}_p; T) = \frac{L_p(X, T)}{(1-T)(1-pT)}, \tag{3.4}$$

where $L_p(T) \in \mathbb{Z}[T]$ is a polynomial of degree $2g$ with constant term 1. Given a finite set $\mathcal{F}$ of curves, and an invariant $\text{inv}(X)$ for each X, we will associate a labelled dataset of the form:

$$\mathcal{D} = \{v_L(X) \to \text{inv}(X) : X \in \mathcal{F}\}, \tag{3.5}$$

where $v_L(X) \in \mathbb{R}^N$ is a vector to be constructed in the examples below. The co-ordinates of the vectors $v_L(X)$ are (conjecturally, at least) distributed like eigenvalues of random matrices in $\mathrm{USp}(2g)$. These co-ordinates are indexed by primes, and we expect them to behave as independent random variables.

The invariants $\text{inv}(X)$ will generally be integers. For example, the rank or torsion order of the associated Mordell–Weil groups, or the Tate–Shafarevich group order. In Section 1, we mentioned some open problems connected to the rank and Tate–Shafarevich group. In particular, there is no known algorithm for their computation.

Example 1. Say $X = E$ is an elliptic curve defined over $\mathbb{Q}$. If p is a good prime for E, then:

$$L_p(E, T) = 1 - a_p T + pT^2, \quad (p \text{ good}), \tag{3.6}$$

where

$$a_p = p + 1 - \#E\left(\mathbb{F}_p\right). \tag{3.7}$$

For a bad prime p, we also define a_p as in equation (3.7), and put

$$L_p(E, T) = 1 - a_p T, \quad (p \text{ bad}).$$

At a bad prime p for an elliptic curve written as a minimal Weierstrass equation, the integer a_p can be seen as encoding the reduction type of E mod p. If p_i denotes the ith prime number then, out of the infinite sequence, $(a_{p_1}, a_{p_2}, \dots)$, we define the L-function of an elliptic curve E by

$$L(E, s) = \prod_{p\text{: prime}} L_p(E, p^{-s})^{-1}.$$

The main idea of this paper is to take a finite sequence $(a_{p_1}, a_{p_2}, \dots, a_{p_N})$ as a feature vector to which we apply machine learning tools. Using SageMath (The Sage Development Team, 2020), we may compute a large amount of a_p quickly. For $i \in \mathbb{Z}_{>0}$, let p_i denote the ith prime. For a positive integer N, we introduce the vector:

$$v_L(E) = (a_{p_1}, \dots, a_{p_N}) \in \mathbb{Z}^N, \quad N \in \mathbb{Z}_{\geq 1}. \tag{3.8}$$

In practice, we will take N to be 100 (resp. 200, 300, 500), so that $p_N = 541$ (resp. 1223, 1987, 3571).

A vector as in equation (3.8) does not determine a unique elliptic curve. Indeed, elliptic curves in the same isogeny class will have the same L-function. Therefore, our datasets use representative curves of isogeny classes. We will discuss this again at points throughout Section 4. Furthermore, if N is small, it can occur that non-isogenous curves have the same first N coefficients. By choosing N as outlined above, we typically circumvent this problem.

Example 2. If $X = C$ is a smooth projective geometrically connected genus 2 curve defined over $\mathbb{Q}$ and p is a good prime for C, then:

$$L_p(C, T) = 1 + a_{1,p}T + a_{2,p}T^2 + a_{1,p}pT^3 + p^2T^4, \quad a_{1,p}, a_{2,p} \in \mathbb{Z}, \quad (p \text{ good}). \tag{3.9}$$

If p is a bad prime for C, then $L_p(C, T)$ is a polynomial of degree < 4. There are only a few bad primes, and since the average values of $a_{1,p}$ and $a_{2,p}$ of a generic genus 2 curve are 0 and p, respectively, we will simply use the following convention:

$$(a_{1,p}, a_{2,p}) = (0, p), \quad (p \text{ bad}). \tag{3.10}$$

Using SageMath (The Sage Development Team, 2020), we may compute $(a_{1,p}, a_{2,p})$. For a positive integer N, we introduce the vector:

$$v_L(C) = \left((a_{1,p_2}, a_{2,p_2}), \ldots, (a_{1,p_{N+1}}, a_{2,p_{N+1}})\right) \in \left(\mathbb{Z}^2\right)^N, \quad N \in \mathbb{Z}_{\geq 1}, \tag{3.11}$$

where we do not include $p_1 = 2$ as it is always bad. In practice, we will take $N = 200$. As with equation (3.8), we can construct our databases to insure that $v_L(C_1) = v_L(C_2)$ only occurs when $C_1 = C_2$. We will discuss our choices further in Section 5.

3.2. Generic experimental strategy

1. Let $\mathcal{F}$ be a finite set of elliptic curves (resp. smooth projective geometrically connected genus 2 curves). The choice of $\mathcal{F}$ depends on the experiment. For example, $\mathcal{F}$ could be the set of elliptic curves (resp. genus 2 curves) over $\mathbb{Q}$ with conductor less than some bound and rank in the set $\{0, 1\}$ (resp. $\{0, 1, 2\}$).
2. For an elliptic curve E (resp. genus 2 curve C) in $\mathcal{F}$, let $\mathrm{inv}(E)$ (resp. $\mathrm{inv}(C)$) denote an invariant of interest. For example, $\mathrm{inv}(E)$ (resp. $\mathrm{inv}(C)$) could be the rank of $E(\mathbb{Q})$ (resp. $J(\mathbb{Q})$).
3. Generate datasets of the form $\mathcal{D} = \{v_L(X) \to \mathrm{inv}(X) : X \in \mathcal{F}\}$, where $\mathcal{D}$ is as in (3.5).[1] We will take N to be one of: $100, 200, 300, 500$. We stress at this point that N is an absolute constant, and does not vary with the curves in $\mathcal{D}$.
4. Divide the values of $\mathrm{inv}(X)$ into k groups according to the classification problem under consideration, and decompose $\mathcal{D}$ into a disjoint union

   $$\mathcal{D} = \mathcal{D}_1 \sqcup \mathcal{D}_2 \sqcup \cdots \sqcup D_k$$

   by assigning the corresponding category label $s \in \{1, 2, \ldots, k\}$. Almost always in this paper, we have $\#\{\mathrm{inv}(X) : X \in \mathcal{F}\} = k \in \{2, 3\}$, i.e., inv(X) naturally becomes a label and we consider a binary or ternary classification problem.
5. Choose a subset $\mathcal{T} \subset \mathcal{D}$ and denote its complement by $\mathcal{V} = \mathcal{D} - \mathcal{T}$. We will refer to $\mathcal{T}$ as the training dataset, and $\mathcal{V}$ as the validation dataset. It is important that the training set and validation set have no intersection so as not to over-fit the machine learning. We will not typically specify $\mathcal{T}$, or its size relative to $\mathcal{D}$, as the choice will not impact significantly on the results (see also step 8). Nonetheless, our most common choice is to use 70% of the dataset for $\mathcal{T}$ and 30% for $\mathcal{V}$.
6. Train a classifier on the set $\mathcal{T}$ with a standard supervised-learning algorithm. In this paper we will use naive Bayes, random forests, and logistic regression - see Hastie et al. (2001, Sections 4.4, 6.6.3, 15). We implement the algorithms using MATHEMATICA (Wolfram Research, Inc., 2020).
7. For all curves X in $\mathcal{V}$, ask the classifier to determine the category label $s \in \{1, 2, \ldots, k\}$ (or frequently $\mathrm{inv}(X)$ itself). When $k = 2$, we record the precision and confidence, which together constitute a good measure of accuracy and performance of the machine. The precision and confidence are real numbers in the interval $[0, 1]$, and the aspiration is that both are close to 1. By precision, we mean the proportion of predictions in agreement with The LMFDB Collaboration (2020), the validity of which is discussed in The LMFDB Collaboration (2020, Reliability of elliptic curve data over $\mathbb{Q}$, Reliability of genus 2 curve data over $\mathbb{Q}$). By confidence, we mean the Matthew's correlation coefficient (Matthews, 1975). The confidence value is an extra check intended to minimize false positives and false negatives. For $k > 2$, we record the precision and the confusion matrix, which performs a similar role to the confidence value and is more transparent.
8. Repeat steps 5 to 7 for different choices of $\mathcal{T}$. The precision and confidence values recorded below are representative of several repetitions.

[1] In exceptional circumstances, we will in fact construct different datasets in place of $\mathcal{D}$. We will do this, for example, in the investigation of particularly accurate classifiers as in Section 4.6, and in an attempt to improve on a poorly performing classifier as in Section 4.5. Such a digression from convention will always be clearly indicated.

3.3. Further specifications

In this section, we give brief descriptions of the machine learning methods adopted in this paper to explain how specific classifiers predict a label for $v \in \mathcal{V}$ based on the labels for $v \in \mathcal{T}$.

3.3.1. Logistic regression

For binary classification problems, the logistic regression classifier predicts the label

$$[\sigma(w \cdot v_L(E) + b)] \in \{0, 1\},$$

where $w \in \mathbb{R}^N$, $b \in \mathbb{R}$, σ is the logistic sigmoid function, and $[\sigma(x)]$ is the nearest integer to $\sigma(x)$.

The parameters w and b are estimated by using numerical methods to solve a non-linear equation derived from the labels for vectors in $\mathcal{T}$. Note that the parameters w, b yield a heuristic formula to predict $\mathrm{inv}(X)$. For n-ary classification problems, the function σ is replaced by the softmax function.

3.3.2. Naive Bayes

Given a curve X corresponding to the unlabelled vector $v = v_L(X) \in \mathcal{V}$, the naive Bayes classifier predicts the label given by

$$\underset{s \in \{1,2,\ldots,k\}}{\operatorname{argmax}} \left(\mathrm{Prob}(\mathrm{inv}(X) \equiv s | v_L(X) = v)\right), \tag{3.12}$$

where $\equiv$ is a shorthand for the correspondence between invariant values and label values (often this is an equality). The conditional probability in equation (3.12) is estimated by applying Bayes theorem to the set $\mathcal{T}$, under a naive independence assumption.

3.3.3. Random forests

The random forest classifier divides several copies of $\mathbb{R}^N$ into labelled regions, and each copy of $\mathbb{R}^N$ tentatively predicts the label for $v_L(X)$ based on the region into which it falls. Each partition of $\mathbb{R}^N$ is determined by randomly choosing certain features, and splitting the corresponding axes according to the labels of vectors in $\mathcal{T}$. Each partition of $\mathbb{R}^N$ is therefore determined by a decision tree, and the random forest classifier predicts the label given by a majority of trees.

3.3.4. Replicability

In the following sections, we will record the precision of the classifiers above when applied to particular invariants from arithmetic geometry. The experimental set-up can be reproduced to yield similar results. That said, without specifying $\mathcal{T}$, the exact precision values are unlikely to be replicated.

4. Elliptic curves

In this section we describe our experimental results for elliptic curves defined over $\mathbb{Q}$. For standard algorithms used in the computation of the invariants discussed below, the reader is referred to Cremona (1997). To perform the experiments in this section, we downloaded data from The LMFDB Collaboration (2020, Elliptic curves over $\mathbb{Q}$) the completeness of which is discussed in The LMFDB Collaboration (2020, Completeness of elliptic curve data over $\mathbb{Q}$). For implementations of principal component analysis with this data, the reader is referred to He et al. (under review).

We note that the Hasse–Weil L-function of an elliptic curve E is an invariant of its isogeny class, and so we in fact downloaded a representative curve for each isogeny class. In LMFDB, an isogeny class is represented by an optimal curve, and hence our data sets are generated from optimal curves only. In general, the torsion order, torsion structure and the number of integral points, considered in Sections 4.2 - 4.4, are not uniquely determined by an isogeny class.

Table 1
The above table shows the precision and confidence of a logistic regression classifier when asked to distinguish elliptic curves over $\mathbb{Q}$ with rank 0 from those with rank 1. The classifier is trained on E with conductor Q_E in the ranges specified by the first column, using the number of Euler factors given in the second column. The classifier is verified on E with conductor in the ranges specified by the third column.

Q_E training range	N	Q_E validation range	$\#\{E\}$	Precision	Confidence
$[1, 1 \times 10^4]$	100	$[1, 1 \times 10^4]$	1.6×10^4 $(\times 2)$	0.977	0.955
"	300	"	"	0.991	0.982
$[2 \times 10^4 + 1, 3 \times 10^4]$	300	$[2 \times 10^4 + 1, 3 \times 10^4]$	1.7×10^4 $(\times 2)$	0.964	0.922
"	500	"	"	0.971	0.941
$[1, 1 \times 10^4]$	300	$[2 \times 10^4 + 1, 3 \times 10^4]$	"	0.924	0.848

Table 2
The above table shows the precision and confidence of a naive Bayes classifier when asked to distinguish elliptic curves over $\mathbb{Q}$ with torsion order 1 from those with torsion order 2. The classifier is trained on a random sample of curves with conductor in the range specified by the first column, and verified on those which remain.

Q_E range	$\#\{E\}$	N	Precision	Confidence
$[1, 3 \times 10^4]$	3.57×10^4 $(\times 2)$	500	0.9997	0.9995

4.1. Rank

Recall that we denote by r_E the rank of an elliptic curve E.

It is conjectured that, in a rigorous sense, 50% of elliptic curves over $\mathbb{Q}$ have rank 0, 50% have rank 1, and 0% have higher rank, cf. Goldfeld (1979, Conjecture B); Katz and Sarnak (1999a,b). Furthermore, it is known that if $r_E \leq 1$ then r_E is equal to the order of vanishing of $L(E, s)$ at $s = 1$. It is therefore expedient to consider this as a binary classification problem using the vectors v_L defined by Euler factors as in (3.8). For different ranges of conductor Q_E, we established a balanced dataset of size $\sim 2 \times 10^4$ $(\times 2)$ for rank 0 and rank 1.

More precisely, our dataset is formed in the following way. The LMFDB has *all* the elliptic curves over $\mathbb{Q}$ with conductor up to 500,000 The LMFDB Collaboration (2020, Completeness of elliptic curve data over $\mathbb{Q}$). We consider curves with conductor at most 30,000 in this subsection. For the first experiment with conductor range $[1, 1 \times 10^4]$, there are 16,450 curves of rank 0 and 19,622 curves of rank 1. We choose 16,000 curves uniformly at random from each rank to form our dataset. A similar process is taken to form datasets for the other experiments.

Trying several standard classifiers, we find that logistic regression worked best and the results are summarized in Table 1. We see that the accuracies are in the high 0.90 s, which is reassuring that a machine learns ranks of elliptic curves. What is of particular interest is the last line in the table, where we trained on 300 Euler factors for conductors in the range from 1 to 10^4 but validated on those in the range from $2 \times 10^4 + 1$ to 3×10^4, and still achieved a 0.92 precision.

The results show that the number of Euler factors needed for high precision is about $3\sqrt{\max\{Q_E\}}$ in the range of Q_E we considered. We also note that a logistic regression classifier also performed best in distinguishing the ranks of algebraic number fields (He et al., 2022b) when number fields were presented through defining polynomials. On the other hand, when trained on Weierstrass coefficients as in Alessandretti et al. (in press), no classifier was able to accurately predict the rank of an elliptic curve.

4.2. Torsion order

Unlike the rank, for which there is no established algorithm for its computation, the torsion subgroup of an elliptic curve can be computed using rigorous algorithms based on theorems of Nagell and Lutz. Furthermore, the possibilities for the torsion subgroup were completely classified by Mazur. In fact, the torsion group of an elliptic curve over $\mathbb{Q}$ has order at most 16 (Silverman, 1992, Chapter VII, Theorem 7.5).

Table 3
The above table shows the precision and confidence of a random forest classifier when asked to distinguish elliptic curves over $\mathbb{Q}$ such that $E(\mathbb{Q})_{\text{tors}} \cong \mathbb{C}_4$ from those such that $E(\mathbb{Q})_{\text{tors}} \cong \mathbb{C}_2 \times \mathbb{C}_2$. The classifier is trained on a random sample of curves with conductor in the range specified by the first column, and verified on those which remain.

Q_E range	$\#\{E\}$	N	Precision	Confidence
$[1, 1 \times 10^6]$	5.4×10^3 $(\times 2)$	500	0.885	0.789

Table 4
The above table shows the precision and confidence of a naive Bayes classifier when asked to distinguish elliptic curves over $\mathbb{Q}$ with no integral points from those with a single integral point. The classifier is trained on a random sample of curves with conductor in the range specified by the first column, and verified on those which remain.

Q_E range	$\#\{E\}$	N	Precision	Confidence
$[1, 5 \times 10^4]$	3.2×10^4 $(\times 2)$	500	0.999	0.998

Nevertheless, for the sake of completeness, we investigate the torsion subgroup from the perspective of machine learning. Currently, there are too few data-points on LMFDB to experiment with torsion order > 2. Thus, we perform supervised machine learning of the form $\{v_L\} \to \#(E(\mathbb{Q})_{\text{tors}}) =$ 1 or 2, whereby predicting the torsion group being trivial or not, using the Euler factor coefficients alone. To be clear, we do not restrict the rank of a curve in this experiment. We established a balanced dataset of size $\sim 4 \times 10^4$ $(\times 2)$ for torsion order 1 and 2 together. A naive Bayes classifier was used and the results are summarized in Table 2. We see that the accuracies are extremely good, using 500 a_p coefficients. We note that the naive Bayes classifier appeared also in He et al. (2022a). We will revisit this experiment in Section 4.6.

4.3. Torsion structure

Continuing with the torsion group, let us see how well the actual torsion group can be distinguished. We established a balanced dataset of size $\sim 5 \times 10^3$ $(\times 2)$ for $\mathbb{C}_4$ and $\mathbb{C}_2 \times \mathbb{C}_2$ altogether. Using a random forest classifier, we found that $E(\mathbb{Q})_{\text{tors}}$ being $\mathbb{C}_4$ or $\mathbb{C}_2 \times \mathbb{C}_2$ can be separated using 500 a_p coefficients to fairly good accuracy. The results are summarized in Table 3. Note that the size of the dataset is relatively small compared to those of previous experiments. With a larger dataset, the precision might be improved.

4.4. Integral points

It is known that an elliptic curve has only finitely many integral points (Silverman, 1992, Chapter VIII, Chapter IX, Theorem 3.1). In contrast, it may have infinitely many rational points (this is the case when the rank $r_E > 0$), as addressed above. We set up a supervised ML to try to distinguish curves with no integral points from those with a single integral point, a total of around 60 thousand curves with conductor in the interval $[1, 5 \times 10^4]$. A balanced data-set of size $\sim 3.2 \times 10^4$ $(\times 2)$ for "single integral point" or "no integral point" was established and a naive Bayes classifier produced the results summarized in Table 4. One can see that the results are extremely good. We will revisit this experiment in Section 4.6.

4.5. Tate–Shafarevich group

Finally, we come to the Tate–Shafarevich group, one of the most subtle parts of BSD. A definition of the Tate–Shafarevich group is given in Silverman (1992, Section X.4). The Tate–Shafarevich group is not known to be finite, and there are no effective methods available for its computation. The LMFDB records the analytic order of the Tate–Shafarevich group, that is the real number implied by equation (1.2), which is equal to the order conditionally on the BSD conjecture. If the Tate–Shafarevich group is finite, then its order is a square integer.

Table 5
The above table shows the precision of a logistic regression classifier when asked to distinguish elliptic curves over $\mathbb{Q}$ with Tate–Shafarevich order 4 from those with order 9. The classifier is trained on a random sample of curves with conductor in the range specified by the first column, and verified on those which remain.

Q_E range	$\#\{E\}$	N	Precision
$[1, 10^6]$	2.8×10^4 ($\times 2$)	500	0.589

We could try the following binary classification problem: take 500 Euler a_p coefficients and see whether one could distinguish between a Tate–Shafarevich group of order 4 versus 9. We tried a variety of methods, such as Bayesian or logistic classifiers, as well as some forward-feeding neural-networks, but none performed especially well. The best result was attained by logistic regression with precision merely 0.589. This is in accordance with the difficulty in computing this group. The results are summarized in Table 5.

For this problem alone, we implemented Weierstrass coefficient training (as was done in Alessandretti et al., in press). This experimental variant did not do well with any of the standard classifiers or regressors, again yielding no better than < 0.6 precision. Nevertheless, we briefly review this approach for completeness. Every elliptic curve over $\mathbb{Q}$ has a unique *reduced* minimal Weierstrass equation of the form:

$$\begin{aligned} &y^2 + e_1xy + e_2y = x^3 + e_3x^2 + e_4x + e_5, \\ &e_1, e_3 \in \{0, 1\}, \quad e_2 \in \{-1, 0, 1\}, \quad e_4, e_5 \in \mathbb{Z}. \end{aligned} \tag{4.13}$$

Using the coefficients in (4.13), we define the vector:

$$v_W(E) = (e_1, e_2, e_3, e_4, e_5) \in \mathbb{Z}^5. \tag{4.14}$$

Let $\mathcal{F}$ denote a finite set of elliptic curves, and, for each $E \in \mathcal{F}$, let $\mathrm{inv}(E)$ be an invariant of interest. For example, $\mathcal{F}$ could be the set of all elliptic curves over $\mathbb{Q}$ with conductor less than one million and, for $E \in \mathcal{F}$, the invariant $\mathrm{inv}(E)$ could be the rank of E. We introduce the following labelled dataset:

$$\mathcal{D}_W = \{v_W(E) \to \mathrm{inv}(E) : E \in \mathcal{F}\}. \tag{4.15}$$

Such a labelled dataset was used in Alessandretti et al. (in press).

4.6. Interpretation of naive Bayesian models

Of the experimental results above, two instances with strikingly high accuracies are: the order of torsion subgroups in $E(\mathbb{Q})$ (Section 4.2), and, the existence of integral points on E (Section 4.4). The naive Bayes classifier was found to be optimal in both cases. Below we explore two explanations.

4.6.1. Parity of a_p

We first observe that these classification problems are related to one another. Indeed, it can be shown that[2]:

1. If $\#E(\mathbb{Z}) = 1$, then $\#E(\mathbb{Q})_{\mathrm{tors}} = 2$. Furthermore, the unique integral point is the torsion generator.
2. If $\#E(\mathbb{Z}) = 0$, then $\#E(\mathbb{Q})_{\mathrm{tors}} \leq 2$. Furthermore, in the LMFDB data, 99.99% of optimal elliptic curves over $\mathbb{Q}$ with $\#E(\mathbb{Z}) = 0$ have torsion order 1.

We might therefore expect that if a classifier can distinguish between $\#E(\mathbb{Q})_{\mathrm{tors}} \in \{1, 2\}$ then it can distinguish between $\#E(\mathbb{Z}) \in \{0, 1\}$.

[2] Álvaro Lozano-Robledo and Chris Wuthrich informed us that one can prove these statements using the Nagell–Lutz theorem and other facts about elliptic curves.

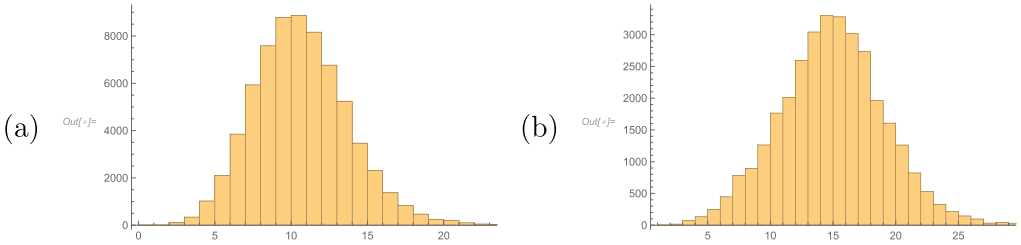


Fig. 1. The number of zeros in the list of 500 a_p coefficients for elliptic curves (a) without integral points, and (b) with a single integral point.

On the other hand, we observe the following "human" procedure for distinguishing between torsion order 1 and 2 using the vectors $v_L(E)$ as in equation (3.8). Recall from equation (3.7) that $a_p = p + 1 - \#E(\mathbb{F}_p)$. When p is an odd prime, it follows that a_p is even if and only if $\#E(\mathbb{F}_p)$ is even. If p is moreover a prime of good reduction, then a point of order 2 in $E(\mathbb{Q})$ maps to a point of order 2 mod p and so $\#E(\mathbb{F}_p)$ is even. We conclude that if $\#E(\mathbb{Q})_{\text{tors}} = 2$, then the vector $v_L(E)$ consists of even integers with a few possible exceptions coming from $p = 2$ and bad primes. (It is known that $a_p = -1, 0, 1$ for bad primes p, and the exceptional values of a_p are actually equal to ± 1). In the case $\#E(\mathbb{Q})_{\text{tors}} = 1$ we observe that a_p's are frequently odd as well as even. We are led to speculate that a naive Bayes classifier successfully distinguishes between vectors whose entries are all even from those whose entries are a mixture of even and odd numbers.

To test this, we perform the following experiment. We generate one set of 100-dimensional vectors with random integer coordinates in the range $[-10, 10]$, and another set of 100-dimensional vectors with coordinates equal to two times a random integer in the range $[-5, 5]$. A naive Bayes classifier is able to distinguish these vectors to 99.8% accuracy. By comparison, a random forest achieves around 74%. These accuracies are comparable to those observed in our experiments in Sections 4.2 and 4.4 and confirm the expectation that a Bayes classifier recognizes this difference.

4.6.2. Number of zeros in v_L

As a separate attempt, inspired by the Lang–Trotter (Lang and Trotter, 1976) conjecture, we study the distribution of the number of zeros in the vector v_L of 500 a_p values, for curves with a single integral point and without integral points (the case with the order of torsion group being 1 or 2 is similar to the ensuing discussions, mutatis mutandis). For this purpose, we separately consider the set $\mathcal{F}_0$ of curves with no integral points and the set $\mathcal{F}_1$ of curves with a single integral point, where $|\mathcal{F}_0| = 68,154$ and $|\mathcal{F}_1| = 32,816$. We calculate the number of zeros in v_L for each curve $E \in \mathcal{F}_i$, $i = 0, 1$, and draw the resulting histograms. This is shown in parts (a) and (b) respectively in Fig. 1. Clearly, the means of the two distributions are different. Precisely, $\mathcal{F}_0$ has mean 10.85 with standard deviation 13.21, while $\mathcal{F}_1$ has mean 15.26 with standard deviation 14.70.

To check whether a naive Bayes classifier detects this difference, we define the following binary vector for a positive integer N:

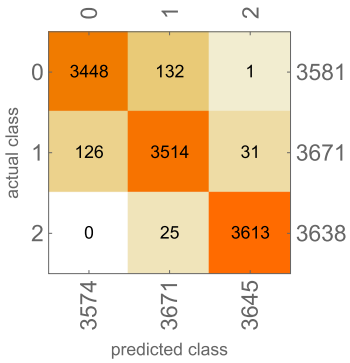
$$v_B(E) = (\delta_{p_1}, \ldots, \delta_{p_N}) \in \{0,1\}^N, \quad \delta_{p_i} = \begin{cases} 0, & a_p = 0, \\ 1, & a_p \neq 0. \end{cases} \tag{4.16}$$

The binary vectors in equation (4.16) are analogous to the binary vectors used in He et al. (2022b). Replacing $v_L(E)$ with $v_B(E)$ in Section 3.2 (Step 3) and performing the experiment in Section 4.4, we observe that the naive Bayes is accurate to around 0.8 precision. The result is similar if we instead use the following ternary vectors:

$$v_T(E) = (\epsilon_{p_1}, \ldots, \epsilon_{p_N}) \in \{-1,0,1\}^N, \quad \epsilon_{p_i} = \begin{cases} -1, & a_p < 0, \\ 0, & a_p = 0, \\ 1, & a_p > 0. \end{cases} \tag{4.17}$$

Table 6
The above table shows the precision and the confusion matrix of a logistic regression classifier with validation set of size $10,890 = 0.3 \times \#\{C\}$ when asked to distinguish between genus 2 curves over $\mathbb{Q}$ with rank in the set $\{0, 1, 2\}$.

$\#\{C\}$	Precision
1.21×10^4 $(\times 3)$	0.971



Therefore, we see that what the Bayes classifier is picking up to reach the near 100% predictions is based on *more than* merely the frequency of zeros/positives/negatives. If we do include the actual values of the a_p coefficients, it takes at least around 7 coefficients to get to more than 0.9 accuracy.

On the other hand, we should point out that the number of zeros to the a_p-coefficients is part of the Lang–Trotter (Lang and Trotter, 1976) conjecture which is a refinement of the Sato–Tate (Tate, 1965) conjecture. We are not aware of any claims in the literature that relate the distribution of zeros in the a_p-coefficients to the number of integral points on or the torsion order of an elliptic curve. Our experimental results suggest that such relations may exist.

5. Genus 2 curves

Having met with success for the genus 1 case, in this section we describe our experimental results for genus 2 curves defined over $\mathbb{Q}$. Throughout, we take the $N = 200$ pairs $(a_{p,1}, a_{p,2})$ of coefficients in the L-function and a conductor range from 1 to 1 million. This conductor range includes all the genus 2 curves available in LMFDB. See The LMFDB Collaboration (2020), Completeness of genus 2 curve data over $\mathbb{Q}$. Again the L-function depends only on the isogeny class, but this time we cannot specify one curve per isogeny class in the LMFDB data. Nevertheless, over 99% of the genus 2 isogeny classes in the database (which has 65534 classes and 66158 curves) have a unique representative. Accepting this slight redundancy, we simply use all the genus 2 curves available in LMFDB.

5.1. Rank

We performed an experiment analogous to that in Section 4.1. In the current context, a significant proportion of genus 2 curves have rank 2 and we consider the ternary classification problem of predicting whether the rank is 0, 1, or 2, from the Euler coefficients. A balanced dataset of size $\sim 1 \times 10^4$ $(\times 3)$ was thus established and a logistic regression classifier was found to perform well, with accuracies ~ 0.97. We emphasize that this is a 3-way classification and to obtain this level of accuracies in impressive. The results are summarized in Table 6.

5.2. Torsion order

As with the genus 1 case, we can try to distinguish the torsion group of order 1 versus 2 in a binary classification (cf. Section 4.2). A balanced data-set was established, with size $\sim 1.5 \times 10^4$ $(\times 2)$ and a naive Bayes classifier was found to perform best, with results presented in Table 7.

Table 7
The above table shows the precision and confidence of a naive Bayes classifier when asked to distinguish genus 2 curves over $\mathbb{Q}$ with torsion order 1 from those with torsion order 2.

#{C}	Precision	Confidence
1.46×10^4 $(\times 2)$	0.926	0.854

Table 8
The above table shows the precision of all classifiers when asked to distinguish between genus 2 curves over $\mathbb{Q}$ with number of rational points as in the first column.

# of rational points	#{C}	Precision
$\{0, 1, 2, 3, 4, 5, 6\}$	$5 \times 10^3 (\times 7)$	< 0.34
$\{2, 4\}$	$9.4 \times 10^3 (\times 2)$	< 0.73

Table 9
The above table shows the precision and confidence of a logistic regression classifier when asked to distinguish genus 2 curves over $\mathbb{Q}$ with trivial Tate–Shafarevich group from those with non-trivial Tate–Shafarevich group.

#{C}	Precision	Confidence
4.2×10^4 $(\times 2)$	0.78	0.562

5.3. Rational points

As mentioned in the Introduction, curves of genus > 1 have only a finite number of rational points. This allows for an experiment slightly different in nature to what was possible with elliptic curves. Indeed, one could ask for a multi-category classification using the number of rational points, being predicted from the L-function coefficients. We tried various classes, after balancing the data but no classifier performed especially well. The results are summarized in Table 8, where a 7-way classification is shown in the first row, and a binary, in the second. We suspect that training with a larger data set would result in a better performance.

5.4. Trivial Tate–Shafarevich group

Finally, we move to the Tate–Shafarevich group. Note that the order now needs not be a square for a genus 2 curve. We performed a binary-classification (having established a balanced data set of size $\sim 4 \times 10^4$ $(\times 2)$) of whether Tate–Shafarevich group is trivial or not. Again, no classifier was found to perform particularly well, though a logistic regression classifier performed best (see Table 9), and the accuracies are comparable to those of the genus 1 case, However, once again, the prediction is better than completely random.

6. Conclusions and outlook

The experiments in this paper show that an ML classifier can be trained to predict the rank and the torsion order of an elliptic curve or a genus 2 curve with high precision when the curve is represented by a few hundred coefficients of its L-function. In particular, for elliptic curves, the torsion order and the number of integral points are determined almost perfectly by ML classifiers. Among the discrete invariants appearing in the BSD conjecture, only the order of the Tate–Shafarevich group seems to be out of reach with our approach of using a finite number of L-function coefficients.

Along with our previous work (He et al., 2022a,b), this paper confirms that ML classifiers perform surprisingly well with various invariants in number theory. High accuracies attained in our experiments reflect that data sets arising from mathematics are actually “clean and clear” without any noise.

Tentatively, this opens up new opportunities of developing ML techniques for mathematics which exploit mathematical structures in data sets. In particular, we might expect high accuracy classifiers for invariants attached to any L-function through applying supervised learning strategies to their first coefficients. In forthcoming work, we investigate class numbers from this perspective, building on the results of He et al. (2022b).

With all these experimental results and evidence at hand, a compelling call to action is to understand what ML classifiers actually recognize in the data sets. Though the algorithms of standard classifiers are well-known, it does not seem straightforward to precisely analyse what a classifier does with data sets. As a starting point, in a future project, we will explore commonalities amongst misclassified curves and investigate the potential for extrapolation of classifiers beyond current computational limits.

In another direction, we are reminded that the influential Langlands program anticipates correspondences between two kinds of data sets: *arithmetic* data and *automorphic* data. We have been experimenting with arithmetic data. In accordance with Langlands program, we expect that a machine would learn automorphic data with high precision and efficiency. It would be very interesting to investigate whether this expectation is valid.

Declaration of competing interest

We, the authors, Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver declare that we have no conflict of interest in our submission to the special issue Algebraic Geometry and Machine Learning of the Journal of Symbolic Computation.

References

Alessandretti, L., Baronchelli, A., He, Y.H., in press. ML meets number theory: the data science of Birch–Swinnerton–Dyer. arXiv: 1911.02008 [math.NT].

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning - Adaptive Computation and Machine Learning. MIT Press.

Cremona, J., 1997. Algorithms for Modular Elliptic Curves. Cambridge University Press.

Goldfeld, D., 1979. Conjectures on Elliptic Curves over Quadratic Fields. Lecture Notes in Math., vol. 751. Springer, pp. 442–451.

Hasse, H., 1954. Zetafunktion und L-Funktionen zu einem arithmetischen Funktionenkörper vom Fermatschen Typus. Abh. Deutsch. Akad. Wiss. Berlin. Kl. Math. Nat. 1954 (4). 70 pp., 1955.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. NY Springer.

He, Y.H., 2017. Machine-learning the string landscape. Phys. Lett. B 774, 564–568.

He, Y.H., 2019. Deep-learning the landscape. Science 365 (6452).

He, Y.H., 2021. The Calabi-Yau Landscape: From Geometry, to Physics, to Machine Learning. Springer. arXiv:1812.02893.

He, Y.-H., Lee, K.-H., Oliver, T., 2022a. Machine-learning the Sato–Tate conjecture. J. Symb. Comput. 111, 61–72.

He, Y.-H., Lee, K.-H., Oliver, T., 2022b. Machine-Learning Number Fields. Mathematics, Computation and Geometry of Data. In press.

He, Y.-H., Lee, K.-H., Oliver, T., Pozdnyakov, A., under review. Murmurations of elliptic curves. arXiv:2011.08958.

Katz, N.M., Sarnak, P., 1999a. Random matrices, Frobenius eigenvalues, and monodromy. Colloq. Publ. – Am. Math. Soc. 45.

Katz, N.M., Sarnak, P., 1999b. Zeros of zeta functions and symmetry. Bull. Am. Math. Soc. 36 (1), 1–26.

Lang, S., Trotter, H., 1976. Frobenius Distributions in GL_2 Extensions. Springer Lecture Notes in Math., vol. 504.

The LMFDB Collaboration, 2020. The L-functions and modular forms database. http://www.lmfdb.org. (Accessed 1 September 2020). [Online].

Matthews, B.W., 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta, Protein Struct. 405 (2), 442–451.

Park, J., Poonen, B., Voight, J., Wood, M.M., 2019. A heuristic for boundedness of ranks of elliptic curves. J. Eur. Math.

The Sage Development Team, 2020. SageMath, the sage mathematics software system (version 9.1.0). http://www.sagemath.org.

Silverman, J.H., 1992. The Arithmetic of Elliptic Curves, second edition. Graduate Texts in Mathematics, vol. 106. Springer.

Silverman, J.H., 1994. Advanced Topics in the Arithmetic of Elliptic Curves. Graduate Texts in Mathematics, vol. 151. Springer.

Tate, J.T., 1965. Algebraic cycles and poles of zeta functions. In: Arithmetical Algebraic Geometry, Proc. Conf. Purdue Univ. 1963, pp. 93–110.

Wolfram Research, Inc., 2020. Mathematica 12.1. https://www.wolfram.com/mathematica. Champaign, Illinois.