A survey of p-adic point counting

Jan Tuitman

KU Leuven

March 21, 2016

The zeta function

Let $X/\mathbb{F}_q$ be an algebraic variety over a finite field. The zeta function of X is defined as

$$Z(X,T) = \exp \sum_{i=1}^{\infty} |X(\mathbb{F}_{q^i})| \frac{T^i}{i}.$$

Here $|X(\mathbb{F}_{q^i})|$ denotes the number of points on X with values in $\mathbb{F}_{q^i}$, i.e. the number of solutions over $\mathbb{F}_{q^i}$ of the equations that define X.

The zeta function

Let $X/\mathbb{F}_q$ be an algebraic variety over a finite field. The zeta function of X is defined as

$$Z(X,T) = \exp \sum_{i=1}^{\infty} |X(\mathbb{F}_{q^i})| \frac{T^i}{i}.$$

Here $|X(\mathbb{F}_{q^i})|$ denotes the number of points on X with values in $\mathbb{F}_{q^i}$, i.e. the number of solutions over $\mathbb{F}_{q^i}$ of the equations that define X.

Theorem (Weil conjectures)

Suppose that X is smooth and projective of dimension d. Then:

- $Z(X,T) = \frac{P_1 P_3 \ldots P_{(2d-1)}}{P_0 P_2 \ldots\ldots P_{2d}}$, *where* $P_i = \prod_j (1 - \alpha_{ij} T) \in \mathbb{Z}[T]$.
- *The transformation* $t \to q^d/t$ *maps the* $\alpha_{i,j}$ *to the* $\alpha_{2d-i,j}$.
- $|\alpha_{i,j}| = q^{i/2}$ *for all* j, *where* $|\cdot|$ *denotes the complex absolute value.*

Computing zeta functions

Problem

Compute $Z(X, T)$ efficiently.

Computing zeta functions

Problem

Compute $Z(X, T)$ efficiently.

- Naive algorithm: compute enough of the $X(\mathbb{F}_{q^i})$ by trying all values. Far too slow to be useful in practice!
- Schoof's ℓ-adic method: in theory for all curves (Pila), in practice only for elliptic curves (perhaps genus 2 with some extra structure).
- For curves: use the group structure on the Jacobian (e.g. Baby Step Giant Step). Better than naive counting, but still restricted to small fields and low genus.

We will not say anything more about these and focus on p-adic algorithms in the rest of this talk.

Some applications

Discrete Logarithm Problem in cryptography:

- $X/\mathbb{F}_q$ an algebraic curve.
- J its Jacobian variety, $J(\mathbb{F}_q)$ is a finite abelian group.
- DLP: given $P, Q \in J(\mathbb{F}_q)$ find $n \in \mathbb{Z}$ such that $n * P = Q$.
- weak if $|J(\mathbb{F}_q)|$ only has small prime factors.
- $|J(\mathbb{F}_q)|$ can be deduced easily from $Z(X, T)$.

Some applications

Discrete Logarithm Problem in cryptography:

- $X/\mathbb{F}_q$ an algebraic curve.
- J its Jacobian variety, $J(\mathbb{F}_q)$ is a finite abelian group.
- DLP: given $P, Q \in J(\mathbb{F}_q)$ find $n \in \mathbb{Z}$ such that $n * P = Q$.
- weak if $|J(\mathbb{F}_q)|$ only has small prime factors.
- $|J(\mathbb{F}_q)|$ can be deduced easily from $Z(X, T)$.

Collect data about conjectures in number theory, e.g.:

- Sato-Tate and generalisations (higher genus),
- Lang-Trotter and generalisations (modular forms),
- Birch and Swinnerton-Dyer (distribution of analytic rank).

p-adic cohomology

X algebraic variety of dimension d over $\mathbb{F}_q$ with $q = p^n$.

p-adic cohomology

X algebraic variety of dimension d over $\mathbb{F}_q$ with $q = p^n$.

$\mathbb{Q}_q$ the unique unramified extension of degree n of $\mathbb{Q}_p$, $\mathbb{Z}_q$ its ring of integers and $\sigma \in \mathrm{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ the unique lift of $x \mapsto x^p \in \mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$.

p-adic cohomology

X algebraic variety of dimension d over $\mathbb{F}_q$ with $q = p^n$.

$\mathbb{Q}_q$ the unique unramified extension of degree n of $\mathbb{Q}_p$, $\mathbb{Z}_q$ its ring of integers and $\sigma \in \mathrm{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ the unique lift of $x \mapsto x^p \in \mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$.

Fact

One can define p-adic cohomology groups $H^i_{rig}(X)$ and $H^i_{rig,c}(X)$ which are finite dimensional $\mathbb{Q}_q$ vector spaces with a σ-semilinear action of the p-th power Frobenius map F_p on X (sending every coordinate to its p-th power) such that the following Lefschetz formula holds:

$$Z(X,T) = \prod_{i=0}^{2d} \det(1 - TF_p^n | H^i_{rig,c}(X))^{(-1)^{i+1}}$$

Special case: smooth affine variety

Let:

- $X = \mathrm{Spec}(\overline{A})$ with $\overline{A}$ a smooth $\mathbb{F}_q$ algebra of finite type.
- smooth lift $A = \mathbb{Z}_q[x_1, \ldots, x_l]/(f_1, \ldots, f_m)$ such that $A \otimes_{\mathbb{Z}_q} \mathbb{F}_q = \overline{A}$.
- weak completion $A^\dagger = \mathbb{Z}_q\langle x_1, \ldots, x_l\rangle^\dagger/(f_1, \ldots, f_m)$, where

$$\mathbb{Z}_q\langle x_1, \ldots, x_l\rangle^\dagger := \{\sum_I a_I x^I : a_I \in \mathbb{Z}_q, \exists \rho > 1 \text{ s.t. } \lim_{|I|\to\infty} |a_I|\rho^{|I|} = 0\}.$$

- 1-forms $\Omega^1_{A^\dagger} = (A^\dagger dx_1 \oplus \ldots \oplus A^\dagger dx_l)/(A^\dagger df_1 + \ldots A^\dagger df_m)$.
- de Rham complex $\Omega^\bullet_{A^\dagger}$ with $\Omega^i_{A^\dagger} = \Lambda^i(\Omega^1_{A^\dagger})$.

Then $H^\bullet_{rig}(X)$ is the cohomology of the complex $\Omega^\bullet_{A^\dagger}$.

Hyperelliptic curves

Let:

- $\mathbb{F}_q$ a finite field of odd characteristic,
- $Q \in \mathbb{F}_q[x]$ monic of degree $2g+1$ without repeated roots,
- X the smooth projective curve defined by $y^2 = Q(x)$.

X is a hyperelliptic curve of genus g (with a rational Weierstrass point).

For characteristic 2 or curves without a rational Weierstrass point this has to be modified slightly.

Kedlaya (2001) proposed to compute $Z(X, T)$ using p-adic cohomology.

Kedlaya's algorithm

Sketch:

- $X : y^2 = Q(x)$ hyperelliptic of genus g over $\mathbb{F}_q$ with $q = p^n$ odd.
- U open in X defined by $y \notin \{0, \infty\}$.
- $\mathcal{Q} \in \mathbb{Z}_q[x]$ a monic lift of Q.
- $A^\dagger = \mathbb{Z}_q\langle x, y, 1/y\rangle^\dagger/(y^2 - \mathcal{Q})$.
- basis for $H^1_{rig}(X) \subset H^1_{rig}(U)$ given by $[\frac{dx}{y}, \ldots, x^{2g-1}\frac{dx}{y}]$.
- lift Frobenius to $A^\dagger$: $\mathrm{F}_p(x) := x^p$, find $\mathrm{F}_p(y) \equiv y^p \bmod p$ (Hensel).
- Apply F_p to basis for $H^1_{rig}(X)$ and reduce to find matrix F_p.
- Compute $\chi(T) = \det(1 - T\mathrm{F}_p^n | H^1_{rig}(X))$.
- $Z(X, T) = \chi(T)/((1 - T)(1 - qT))$.

Complexity

$X/\mathbb{F}_q$ hyperelliptic $q = p^n$ genus g.

Theorem

Complexity of Kedlaya's algorithm:

time $O((pg^4n^3)^{1+\epsilon})$ *space* $O((pg^3n^3)^{1+\epsilon})$

Only polynomial time for fixed p, so restricted to (somewhat) small characteristic p. All p-adic algorithms suffer from this, but the dependence on p can be improved.

Larger characteristic

Harvey (2006) improved the complexity of Kedlaya's algorithm in p.

Main idea: use Baby Step Giant Step to carry out the reductions in cohomology more efficiently.

Let ω denote an exponent for matrix multiplication.

Theorem

Complexity of Harvey's $\sqrt{p}$ algorithm:

time $O\left(\left(p^{1/2}g^{\omega+5/2}n^{7/2} + \log(p)g^8n^5\right)^{1+\epsilon}\right)$

This is a lot better in p and still polynomial in g, n (but larger exponents).

Average polynomial time

Harvey (2014) came up with an 'average polynomial time' version of Kedlaya's algorithm.

Main idea: for a hyperelliptic curve over $\mathbb{Z}$, i.e. such that $Q \in \mathbb{Z}[x]$, exploit the overlap between the computations for Q mod p for various p (of good reduction), to obtain an algorithm with polynomial runtime per prime.

$||Q||$ = maximum absolute values coefficients of Q, consider all primes with $p < N$.

Theorem

Complexity of Harvey's average polynomial time algorithm (per prime):

time $O\left(g^{8+\epsilon}\log^3(N)\log^{1+\epsilon}(||Q||N)\right)$

Implementation

Sage: only contains code for $q = p$ prime

- Basic Kedlaya for odd p (hidden).
- Harvey $\sqrt{p}$ for large enough p (hypellfrob).

Implementation

Sage: only contains code for $q = p$ prime

- Basic Kedlaya for odd p (hidden).
- Harvey $\sqrt{p}$ for large enough p (hypellfrob).

Magma:

- Basic Kedlaya for all q (Harrison: q odd, Vercauteren: q even).

Implementation

Sage: only contains code for $q = p$ prime

- Basic Kedlaya for odd p (hidden).
- Harvey $\sqrt{p}$ for large enough p (hypellfrob).

Magma:

- Basic Kedlaya for all q (Harrison: q odd, Vercauteren: q even).

Harvey and Sutherland (2014, 2015) have implemented a mod p version of the average polynomial time algorithm for $g = 2$, but the code does not seem to be (publicly) available.

Implementation

Sage: only contains code for $q = p$ prime

- Basic Kedlaya for odd p (hidden).
- Harvey $\sqrt{p}$ for large enough p (hypellfrob).

Magma:

- Basic Kedlaya for all q (Harrison: q odd, Vercauteren: q even).

Harvey and Sutherland (2014, 2015) have implemented a mod p version of the average polynomial time algorithm for $g = 2$, but the code does not seem to be (publicly) available.

Show some examples....

More general curves?

What about more general curves? Hyperelliptic curves (in odd characteristic) are very special.

More general curves?

What about more general curves? Hyperelliptic curves (in odd characteristic) are very special.

Some generalisations of Kedlaya's algorithm:

- Gaudry - Gurel (2001): superelliptic curves
- Denef - Vercauteren (2004): hyperelliptic curves in characteristic 2
- Denef - Vercauteren (2006): C_{ab} curves
- Castryck - Denef - Vercauteren (2006) : nondegenerate curves

More general curves?

What about more general curves? Hyperelliptic curves (in odd characteristic) are very special.

Some generalisations of Kedlaya's algorithm:

- Gaudry - Gurel (2001): superelliptic curves
- Denef - Vercauteren (2004): hyperelliptic curves in characteristic 2
- Denef - Vercauteren (2006): C_{ab} curves
- Castryck - Denef - Vercauteren (2006) : nondegenerate curves

In the last two cases the algorithms turned out to be quite unpractical, complete implementations still not (publicly) available.

This is more or less where things stood until 2014: only practical algorithms and implementations for (hyper/super)elliptic curves.

State of the art

We (2014) have developed a new extension of Kedlaya's algorithm which:

- works for (almost) all curves,
- is more practical than the algorithms for C_{ab} and nondegenerate curves.

State of the art

We (2014) have developed a new extension of Kedlaya's algorithm which:

- works for (almost) all curves,
- is more practical than the algorithms for C_{ab} and nondegenerate curves.

Why *almost* all curves? Input to the algorithm is a lift to characteristic 0 that has good reduction mod p (technical). For many curves easy to find such a lift, but in general can be hard.

State of the art

We (2014) have developed a new extension of Kedlaya's algorithm which:

- works for (almost) all curves,
- is more practical than the algorithms for C_{ab} and nondegenerate curves.

Why *almost* all curves? Input to the algorithm is a lift to characteristic 0 that has good reduction mod p (technical). For many curves easy to find such a lift, but in general can be hard.

Starting from any curve over a number field, algorithm works mod almost all p.

State of the art

We (2014) have developed a new extension of Kedlaya's algorithm which:

- works for (almost) all curves,
- is more practical than the algorithms for C_{ab} and nondegenerate curves.

Why *almost* all curves? Input to the algorithm is a lift to characteristic 0 that has good reduction mod p (technical). For many curves easy to find such a lift, but in general can be hard.

Starting from any curve over a number field, algorithm works mod almost all p.

With Castryck (2016), we have developed a method that (almost always) finds a good lift for curves of genus $g \leq 5$.

Complexity

Suppose that a plane model of a lift of the curve is defined by a polynomial $Q \in \mathbb{Z}_q[x, y]$ with $q = p^n$ which is

- irreducible mod p,
- monic in y,
- of degree d_x in x and degree d_y in y.

Theorem

Complexity of the algorithm:

time $O((pd_y^6 d_x^4 n^3)^{1+\epsilon})$ *space* $O((pd_y^4 d_x^3 n^3)^{1+\epsilon})$

Perhaps possible to combine with Harvey's ideas to obtain $\sqrt{p}$ and average polynomial time versions. Work in progress....

Implementation

Completely implemented in Magma.

On my website:

- pcc_p : computes zeta function over prime field $\mathbb{F}_p$,
- pcc_q : computes zeta function over non prime field $\mathbb{F}_q$,
- goodmodels : find a good lift to characteristic zero for $g \leq 5$.

Implementation

Completely implemented in Magma.

On my website:

- pcc_p : computes zeta function over prime field $\mathbb{F}_p$,
- pcc_q : computes zeta function over non prime field $\mathbb{F}_q$,
- goodmodels : find a good lift to characteristic zero for $g \leq 5$.

Not in the standard distribution of Magma yet, but should be by the end of the year.

I would like to do a Sage implementation, but some function field functionality (integral bases) missing in Sage.

Implementation

Completely implemented in Magma.

On my website:

- pcc_p : computes zeta function over prime field $\mathbb{F}_p$,
- pcc_q : computes zeta function over non prime field $\mathbb{F}_q$,
- goodmodels : find a good lift to characteristic zero for $g \leq 5$.

Not in the standard distribution of Magma yet, but should be by the end of the year.

I would like to do a Sage implementation, but some function field functionality (integral bases) missing in Sage.

Show some examples....

Direct method

Abbott, Kedlaya and Roe (AKR) (2006): Kedlaya's algorithm for smooth hypersurface $X \subset \mathbb{P}^{n+1}_{\mathbb{F}_q}$ of degree d with $q = p^a$.

Main idea: compute the cohomology $H^{n+1}_{rig}(U)$ of $U = \mathbb{P}^{n+1} - X$ with its action of F_p.

Running time should be *about* $O((p^n d^{n^2} a^n)^{1+\epsilon})$.

Direct method

Abbott, Kedlaya and Roe (AKR) (2006): Kedlaya's algorithm for smooth hypersurface $X \subset \mathbb{P}^{n+1}_{\mathbb{F}_q}$ of degree d with $q = p^a$.

Main idea: compute the cohomology $H^{n+1}_{rig}(U)$ of $U = \mathbb{P}^{n+1} - X$ with its action of F_p.

Running time should be *about* $O((p^n d^{n^2} a^n)^{1+\epsilon})$.

This is being improved by Costa, Harvey and Kedlaya (CHK) (201?):

Running time should be *about* $O((p d^{n^2} a^n)^{1+\epsilon})$ and there are also $\sqrt{p}$ and average polynomial time versions.

Can the dependence on the dimension be improved?

Deformation

Lauder (2004) proposes to put the hypersurface $X \subset \mathbb{P}^{n+1}_{\mathbb{F}_q}$ with $q = p^a$ into a (smooth) family Y/T over some open $T \subset \mathbb{P}^1$ with:

- Y_0 a diagonal hypersurface,
- $Y_1 = X$.

Deformation

Lauder (2004) proposes to put the hypersurface $X \subset \mathbb{P}^{n+1}_{\mathbb{F}_q}$ with $q = p^a$ into a (smooth) family Y/T over some open $T \subset \mathbb{P}^1$ with:

- Y_0 a diagonal hypersurface,
- $Y_1 = X$.

The relative cohomology $H^n_{rig}(Y/T)$ is an overconvergent F-isocrystal:

- a p-adic differential equation (the Gauss-Manin connection),
- a Frobenius structure with matrix $\Phi(t)$.

The Frobenius structure is horizontal w.r.t the connection, so $\Phi(t)$ satisfies a p-adic differential equation.

Deformation

Lauder (2004) proposes to put the hypersurface $X \subset \mathbb{P}^{n+1}_{\mathbb{F}_q}$ with $q = p^a$ into a (smooth) family Y/T over some open $T \subset \mathbb{P}^1$ with:

- Y_0 a diagonal hypersurface,
- $Y_1 = X$.

The relative cohomology $H^n_{rig}(Y/T)$ is an overconvergent F-isocrystal:

- a p-adic differential equation (the Gauss-Manin connection),
- a Frobenius structure with matrix $\Phi(t)$.

The Frobenius structure is horizontal w.r.t the connection, so $\Phi(t)$ satisfies a p-adic differential equation.

Main idea: compute $\Phi(0)$ (easy, Y_0 is diagonal) then solve the differential equation for $\Phi(t)$, finally find $\Phi(1)$ and deduce $Z(X, T)$.

Complexity

Pancratz and me (PT) (2013) improved this in terms of complexity (a bit) and in practice (a lot).

Let ω be an exponent for matrix multiplication and e the basis of the natural logarithm.

Theorem

The (simplified) complexity of the deformation method is:

time $O\left((pd^{n(\omega+4)}e^{n(\omega+1)}a^3)^{1+\epsilon}\right)$ *space* $O\left((pd^{5n}e^{3n}a^3)^{1+\epsilon}\right)$

Fibration

Let Y/T be smooth projective family, T open in $\mathbb{P}^1_{\mathbb{F}_q}$.

Lauder (2006): compute the cohomology of the total space Y using Leray spectral sequence.

$$E_{p,q}^2 = H_{rig}^q(T, H_{rig}^p(Y/T)) \Rightarrow H_{rig}^{p+q}(Y)$$

Degenerates at E^2, only get nonzero terms for $q = 0, 1$ etc..

Fibration

Let Y/T be smooth projective family, T open in $\mathbb{P}^1_{\mathbb{F}_q}$.

Lauder (2006): compute the cohomology of the total space Y using Leray spectral sequence.

$$E_{p,q}^2 = H_{rig}^q(T, H_{rig}^p(Y/T)) \Rightarrow H_{rig}^{p+q}(Y)$$

Degenerates at E^2, only get nonzero terms for $q = 0, 1$ etc..

The $H_{rig}^p(Y/T)$ can be computed as in the deformation method: start from a fibre in the family and solve a p-adic differential equation. Lauder shows how to compute in $H_{rig}^1(H_{rig}^p(Y/T))$.

Fibration

Let Y/T be smooth projective family, T open in $\mathbb{P}^1_{\mathbb{F}_q}$.

Lauder (2006): compute the cohomology of the total space Y using Leray spectral sequence.

$$E_{p,q}^2 = H_{rig}^q(T, H_{rig}^p(Y/T)) \Rightarrow H_{rig}^{p+q}(Y)$$

Degenerates at E^2, only get nonzero terms for $q = 0, 1$ etc..

The $H_{rig}^p(Y/T)$ can be computed as in the deformation method: start from a fibre in the family and solve a p-adic differential equation. Lauder shows how to compute in $H_{rig}^1(H_{rig}^p(Y/T))$.

Use this recursively to reduce the dimension of Y.

Has only been tried for a surface fibred in (hyperelliptic) curves.

Average polynomial time

Harvey (2014) gave $\sqrt{p}$ and average polynomial time algorithms for any scheme X of finite type over $\mathbb{Z}$ (no smoothness assumptions).

Average polynomial time

Harvey (2014) gave $\sqrt{p}$ and average polynomial time algorithms for any scheme X of finite type over $\mathbb{Z}$ (no smoothness assumptions).

Theorem

Let $X/\mathbb{Z}$ scheme of finite type and X_p the reduction of X modulo p.

- *$Z(X_p, T)$ can be computed in time $O(p \log^{1+\epsilon}(p))$ space $O(\log(p))$.*
- *$Z(X_p, T)$ can be computed in time $O(p^{1/2} \log^{2+\epsilon})$ space $O(p^{1/2} \log(p))$.*
- *$Z(X_p, T)$ can be computed for all $p < N$ in time $O(N \log^{3+\epsilon}(N))$ space $O(N \log^2(N))$.*

Average polynomial time

Harvey (2014) gave $\sqrt{p}$ and average polynomial time algorithms for any scheme X of finite type over $\mathbb{Z}$ (no smoothness assumptions).

Theorem

Let $X/\mathbb{Z}$ scheme of finite type and X_p the reduction of X modulo p.

- *$Z(X_p, T)$ can be computed in time $O(p \log^{1+\epsilon}(p))$ space $O(\log(p))$.*
- *$Z(X_p, T)$ can be computed in time $O(p^{1/2} \log^{2+\epsilon})$ space $O(p^{1/2} \log(p))$.*
- *$Z(X_p, T)$ can be computed for all $p < N$ in time $O(N \log^{3+\epsilon}(N))$ space $O(N \log^2(N))$.*

To be able to work in this generality, Harvey avoids the use of cohomology. This might also have some disadvantages, the matrices get very large!

Implementation

What has been implemented?

- AKR: Magma code for surfaces publicly available (webpage Kedlaya), C++ implementation of CHK in progress, not (publicly) available yet.
- Deformation PT: C implementation by Pancratz for prime fields $\mathbb{F}_p$ publicly available (my webpage).
- Fibration (in a few cases): Lauder has Magma code, not (publicly) available.
- General average polynomial time stuff: implementation not available yet, not clear how practical.

Lots of things remain to be done, if you're interested let me know!