[Tutorial] L-functions

Karim Belabas

Institut de
Mathématiques de
Bordeaux

First part: Theory

L and Λ-functions (1/3)

Let $\Gamma_{\mathbb{R}}(s) = \pi^{-s/2}\Gamma(s/2)$, where $\Gamma(s) = \int_0^\infty e^{-t}t^{s-1}\,dt$ is Euler's gamma function; given a d-tuple $A = [\alpha_1, \ldots, \alpha_d] \in \mathbb{C}^d$, let $\gamma_A := \prod_{\alpha\in A} \Gamma_{\mathbb{R}}(s+\alpha)$

Given

- a sequence $a = (a_n)_{n\geqslant 1}$ of complex numbers such that $a_1 = 1$,
- a positive *conductor* $N \in \mathbb{Z}_{>0}$,
- a *gamma factor* γ_A as above,

we consider the Dirichlet series

$$L(a,s) = \sum_{n\geqslant 1} a_n n^{-s}$$

and the attached completed function

$$\Lambda_{N,A}(a,s) = N^{s/2}\cdot\gamma_A(s)\cdot L(a,s).$$

L and Λ-functions (2/3)

A *weak L-function* is a Dirichlet series $L(s) = \sum_{n \geqslant 1} a_n n^{-s}$ such that

- The coefficients $a_n = O_\varepsilon(n^{C+\varepsilon})$ have polynomial growth. Equivalently, $L(s)$ converges absolutely in some right half-plane $\mathrm{Re}(s) > C + 1$.
- The function $L(s)$ has a meromorphic continuation to the whole complex plane with finitely many poles.

This becomes an *L-function* if it satisfies a functional equation: there exist a "dual" sequence a^* defining a weak L-function $L(a^*, s)$, an integer k, and completed functions

$$\Lambda(a, s) = N^{s/2}\gamma_A(s) \cdot L(a, s),$$

$$\Lambda(a^*, s) = N^{s/2}\gamma_A(s) \cdot L(a^*, s),$$

such that $\Lambda(a, k - s) = \Lambda(a^*, s)$ for all regular points. The L-function package is able to compute $L^{(m)}(a, s)$ given the above data.

L and Λ-functions (3/3)

In number theory, additional constraints may arise

- $a^* = \varepsilon \cdot \overline{a}$ for some *root number* ε of modulus 1; often, $\varepsilon = \pm 1$;
- the complex coeffients a live in the ring of integer of some fixed number field, often in $\mathbb{Z}$ or a cyclotomic ring $\mathbb{Z}[\zeta]$;
- the growth exponent such that $a_n = O_\varepsilon(n^{C+\varepsilon})$ can be taken as $C = (k-1)/2$ if L is entire (Ramanujan-Petersson), and $C = k - 1$ otherwise;
- the L-function satisfies an Euler product $L(s) = \prod_{p \text{ prime}} L_p(s)$, where the local factor $L_p(s)$ is a rational function in p^{-s};
- the α_i are integers, often in $\{0, 1\}$.

The PARI implementation assumes none of these, although it takes advantage of them when they are true.

Data structures describing L functions

Three data structures are attached to L-functions, by increasing complexity:

- a high-level description of the underlying mathematical situation, to which e.g., we associate the a_p as traces of Frobenius elements; this is done via constructors to be described shortly.
- an Ldata is a low-level description, containing the complete datum (a, a^*, A, k, N, Λ's polar part). This is obtained via the function lfuncreate.
- an Linit contains an `Ldata` and everything needed for fast *numerical* computations in a certain *domain*: it specifies

  (1) the functions to be considered: $L^{(j)}(s)$ for derivatives of order $j \leqslant m$, where m is now fixed;

  (2) the range of the complex argument s, to a certain rectangular region;

  (3) the output bit accuracy.

  This is obtained via the functions lfuninit.

Any of them can be used as the first argument L of the functions we will now describe.

Second part: Practice

Constructors

Riemann ζ	`lfuncreate(1)`
Dirichlet for quadratic char. $(D/\cdot)$	`lfuncreate(D)`
Dirichlet L-series	`lfuncreate(Mod(m,N))`
Dedekind ζ_K, $K = \mathbb{Q}[X]/(T)$	`lfuncreate(T)`
Hecke L-series, χ mod $\mathfrak{f}$	`lfuncreate([bnr,`χ`])`
Artin L-function	`lfunartin`
$L(E, s)$, E elliptic curve	`lfuncreate(E)`
genus 2 curve, $y^2 = F(x)$	`lfungenus2(F)`
$\ldots$, $y^2 + Q(x)y = F(x)$	`lfungenus2([F,Q])`
Modular form	`mftolfun(mf)`
Lattice Θ function	`lfunqf(Q)`

Domains for `lfuninit`

`lfuninit(L, domain, {D = 0})`, where L is from a constructor as above, `D` is the maximal order of derivation and *domain* is symmetric about the real axis, of the form:

- $[c, w, h]$: rectangular box $|\mathrm{Re}(s) - c| \leqslant w$, $|\mathrm{Im}(s)| \leqslant h$;
- $[w, h]$: implicitly $c = k/2$, box centered on the critical line;
- $[h]$: implicitly $c = k/2$ and $w = 0$, a symmetric interval on the critical line.

For `realbitprecision` b, the running time is proportional to

$$(b + h + D \log D + \log(|c| + w))^{d/2+3} \cdot N^{1/2}.$$

Riemann zeta

```
?  L = 1;    \\ encodes the Riemann zeta function
?  lfunan(L, 100)    \\ = first 100 coefficients
%2 = [1, 1, 1, 1, ...]
?  lfun(L, 2)
%3 = 1.6449340668482264364724151666460251892
?  lfunzeros(L,30)
%4 = [14.1347..., 21.0220..., 25.0109...]
?  \pb 32
   realbitprecision = 64 significant bits (6 decimal digits displayed)
?  ploth(t = 0, 100, lfunhardy(L,t))
?  L = lfuninit(L, [100]);    \\ on critical line, height ⩽ 100
?  ploth(t = 0, 100, lfunhardy(L,t))
```

Dedekind zeta

```
?  L = lfuncreate('x^3-2);     \\ K = Q(2^(1/3))
?  lfun(L, 2)
%2 = 1.6026632619004405990075072278650989584
?  lfunzeros(L,30)
(...31 zeros...)
?  \pb 32
L = lfuninit(L, [30]);
ploth(t = 0, 30, lfunhardy(L,t))
```

Elliptic curves

```
?  E = ellinit([0,0,1,-7,6]); L = lfuncreate(E);    \\ L(E,s)
?  lfun(L, 1)
%3 = 0.E-58
?  lfun(E, 1, 1)     \\ L'(1) ≈ 0
%4 = 1.0280697501645834273549120167678691687 E-41
?  lfun(E, 1, 2)     \\ L^(2)(1) ≈ 0
%5 = 2.6769182259726016729068463995455020017 E-41
?  lfun(E, 1, 3)     \\ L^(3)(1) ≠ 0 !
%6 = 10.391099400715804138751850510360917070
?  ellanalyticrank(E)
%7 = [3, 10.391099400715804138751850510360917070]
?  lfunzeros(E,10)
%8 = [0, 0, 0, 2.052..., 3.262..., 4.470..., 4.754..., ...]
\pb 32
?  Lbad = lfuninit(E, [1/2,0,30]); lfunhardy(Lbad, 10)
\\ MISTAKE !
?  L = lfuninit(L, [30]); ploth(t = 0, 30, lfunhardy(L,t))
```

Hasse-Weil zeta, genus 2

```
?  L = lfungenus2([x^2+x, x^3+x^2+1]);
?  lfunan(L,15)
%2 = [1, -3, -2, 4, 0, 6, 0, -3, 3, 0, 0, -8, -5, 0, 0]
?  L = lfuninit(L, [10]); lfun(L,1)
%3 = 0.090490390832429629113589757258015412264
?  lfunzeros(L,9)
%4 = [5.068..., 6.570..., 7.544..., 8.455...]
ploth(t = 0, 10, lfunhardy(L,t))
```

Dirichlet characters

In PARI/GP, given a *finite* abelian group

$$G = (\mathbb{Z}/o_1\mathbb{Z})g_1 \oplus \cdots \oplus (\mathbb{Z}/o_d\mathbb{Z})g_d,$$

with fixed generators g_i of respective order o_i, then

- the *column* vector $[x_1, \ldots, x_d]$~ represents the element $g \cdot x := \sum_{i \leqslant d} x_i g_i \in G$;
- the *row* vector $[c_1, \ldots, c_d]$, represents the character mapping $g_i \mapsto e(c_i/o_i)$ for each i.

The group G is given by a GP structure, e.g. `bid`, `bnf`, `bnr`. We can choose $(g_i) := G$`.gen` (SNF generators), hence $(o_i) = G$`.cyc` and $o_d \mid \cdots \mid o_1$ (elementary divisors).

Dirichlet L-function (Quadratic character)

Primitive *real* characters have a simpler description: they are of the form $(D/.)$ (Kronecker character) for a fundamental discriminant D. Then `lfuncreate(D)` is $L((D/.), s)$.

```
?  lfun(-23, 1)
%1 = 1.9652020541078591659027670051223364151
?  K = bnfinit(x^2+23); [r1,r2] = K.sign
%2 = [0, 1]
?  2^r1*(2*Pi)^r2 * K.no * K.reg / sqrt(abs(K.disc)) / K.tu[1]
%3 = 1.9652020541078591659027670051223364151
```

Dirichlet L-function (General character)

```
?  G = znstar(100, 1);    \\ (ℤ/100ℤ)*
?  G.cyc
%2 = [20, 2]
?  chi = [2, 0];
?  zncharconductor(G,[2,0])    \\ not primitive
%4 = 25
?  L = lfuncreate([G, chi]);    \\ induced primitive char
?  lfun(L, 1)
%6 = 1.1371...+ 0.1227...* I
? L = lfuninit(L, [30]);
?  ploth(t = 0, 30, lfunhardy(L,t))
```

Hecke L-function

```
?  K = bnfinit(x^3-7);
?  G = bnrinit(K, [11, [1]]);    \\ Cl_f(K)
?  G.cyc
%3 = [30]
?  chi = [2]
?  bnrconductor(G, [2])    \\ not primitive
%5 = [[11, 0, 0; 0, 11, 0; 0, 0, 11], [0]]
?  L = lfuncreate([G, chi]);
?  lfun(L, 0)    \\ Slow !
%7 = 0
?  L = lfuninit(L, [1/2,30]);    \\ critical strip
?  lfun(L, 1)
?  lfunzeros(L,29)    \\ 72 zeros...
```

A custom L-function

```
?  a(N) = Vec(q*eta(q + O(q^N))^24);
?  L = lfuncreate([a, 0, [0,1], 12, 1, 1]);
?  lfuncheckfeq(L)
%3 = -127
?  a(8)
%4 = [1, -24, 252, -1472, 4830, -6048, -16744, 84480]
?  mfcoefs(mfDelta(), 8)
%5 = [0, 1, -24, 252, -1472, 4830, -6048, -16744, 84480]
```

Artin L-function

```
?  P = quadhilbert(-47);
?  N = nfinit(nfsplitting(P));
?  G = galoisinit(N);    \\ D_10
?  G.gen
?  G.orders
?  L1 = lfunartin(N,G, [[a,0;0,a^-1],[0,1;1,0]], 5);
?  L2 = lfunartin(N,G, [[a^2,0;0,a^-2],[0,1;1,0]], 5);
?  s = 1 + x + O(x^10);
?  lfun(1,s)*lfun(-47,s)*lfun(L1,s)^2*lfun(L2,s)^2 - lfun(N,s)
```