[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 $\{0, 1\}$.

PARI's 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:

- an Lmath is an 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 $(a, a^*, A, k, N, \Lambda$'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

Riemann zeta (1/2)

```
L = 1; \\Lmath for Riemann zeta function
lfunan(L, 100) \\= first 100 coefficients
lfun(L, 2)
lfunzeros(L,30)
\pb 32
ploth(t = 0, 100, lfunhardy(L,t))
L = lfuninit(L, [100]); \\on critical line, height ≤ 100
ploth(t = 0, 100, lfunhardy(L,t))
```

`lfuninit` domains:

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

Riemann zeta (2/2)

Known bug: near the poles of $\gamma_A(s)$, derivatives get very inaccurate as the order of derivation increases.

```
\pb 64
x0 = 1e-10; lfun(1, 1e-10, 4)
derivnum(x = x0, zeta(x), 4)
```

`\pb 640` and try again…

Dedekind zeta

```
L = lfuncreate('x^3-2); \\Q(2^{1/3})
lfun(L, 2)
lfunzeros(L,30)
\pb 32
L = lfuninit(L, [30]);
ploth(t = 0, 30, lfunhardy(L,t))
```

Hasse-Weil zeta functions

```
E = ellinit([0,0,1,-7,6]);
L = lfuncreate(E); \\L(E,s)

lfun(L, 1)
lfun(E, 1)
lfun(E, 1, 1) \\L'(1)
lfun(E, 1, 2) \\2nd derivative
lfun(E, 1, 3) \\3rd derivative
ellanalyticrank(E)
lfunzeros(E,10)
\pb 32
Lbad = lfuninit(E, [1/2, 0, 30]); \\MISTAKE !
ploth(t = 0, 30, lfunhardy(Lbad,t))
L = lfuninit(E, [1, 0, 30]); \\Better
L = lfuninit(L, [30]); \\Best: foolproof
ploth(t = 0, 30, lfunhardy(L,t))
```

Hasse-Weil zeta, genus 2

```
L=lfungenus2([x^2+x, x^3+x^2+1]);
lfunan(L,30)
L = lfuninit(L, [10]);
lfun(L,1)
lfunzeros(L,9)
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 $[x_1, \ldots, x_d]\tilde{}$ 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

Real characters have a simpler description: $(D/.)$ (Kronecker character) for a fundamental discriminant D. Then lfuncreate(D) is $L((D/.), s)$.

```
lfun(-23, 1)
K = bnfinit(x^2+23);
(2*Pi) * K.no / sqrt(abs(K.disc)) / K.tu[1]
```

General character:

```
G = idealstar(, 100); \\(Z/100Z)^*
G.cyc
chi = [2, 0]
znconreyconductor(G,[2,0]) \\not primitive
L = lfuncreate([G, chi]); \\attached to induced primitive char

lfun(L, 1)
L = lfuninit(L, [30]);
ploth(t = 0, 30, lfunhardy(L,t))
```

Hecke L-function

```
K = bnfinit(x^3-7);
G = bnrinit(K, [11, [1]]);
G.cyc
chi = [2]
bnrconductor(G, [2]) \\not primitive
L = lfuncreate([G, chi]);

lfun(L, 0) \\Slow !
L = lfuninit(L, [1/2,30]); \\critical strip
lfun(L, 0)
lfun(L, 1)
lfunzeros(L,29)
ploth(t = 0, 30, lfunhardy(L,t))
```

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)
```