Computational methods-Lecture 4

Discrete Least squares approximation

Recall that for continuous functions, if we want to minimize

$$I(a_1, \cdots, a_n) = \int_a^b w(x)|f(x) - \sum_{i=1}^n a_i\phi_i(x)|^2\, dx,$$

we can take derivative on a_j:

$$\frac{\partial I}{\partial a_j} = \frac{\partial}{\partial a_j}\langle f - S, f - S\rangle_w = 2\langle f - S, -\phi_j\rangle = 0.$$

This implies that $f - S$ is perpendicular to ϕ_j and thus to the subspace spanned by V. This is a perfect geometric meaning. Using these normal equations, we can find a_j.

It is best to use orthogonal basis, like Legendre polynomials for $w = 1$, the Chebyshev polynomials for $w = \frac{1}{\sqrt{1-x^2}}$. For trigonometric polynomials, we obtain the Fourier series.

1 Discrete least squares approximation

In practice, the values are known at discrete points. In interpolation, we require the function values to match. However, we can relax to require the mean square error to be small for functions from a particular space.

For example, consider $V = span\{\phi_0, \cdots, \phi_n\}$. We know the data $(x_i, y_i)_{i=1}^m$. The function is then given by

$$S(x) = \sum_{j=0}^n a_j\phi_j(x).$$

Then, we aim to minimize

$$I(a_0, a_1, \cdots, a_n) := \sum_{i=1}^m w_i(y_i - S(x_i))^2. \tag{1}$$

Using $\frac{\partial I}{\partial a_j} = 0$, you can derive a system of equations. This is called the (discrete) least square approximation.

Introduce the inner product notation

$$\langle u, v\rangle_w = \sum_{i=0}^{m} w_i u_i v_i.$$

The the equations are given by

$$\langle y - S, \phi_j\rangle = 0,\ \ j = 0, \cdots, n.$$

Consequently, you can form this as the matrix system

$$Ga = d,$$

where

$$G_{ji} = \langle \phi_i, \phi_j\rangle_w,$$

and

$$d_j = \langle y, \phi_j\rangle_w.$$

1.1 Linear functions

Now, assume we want to use linear functions to fit the data. Hence,

$$S(x) = a_0 + a_1 x.$$

We pick the weight $w_i = 1$. Then,

$$I(a_0, a_1) = \sum_{i=1}^{m} (y_i - (a_0 + a_1 x_i))^2.$$

The two equations are then

$$m a_0 + a_1 \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i,$$

and

$$a_0 \sum_{i=1}^{m} x_i + a_1 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i.$$

Alternatively, you can use $\langle \phi_i, \phi_j\rangle_w$ to formulate these coefficients.

Hence,

$$a_0 = \frac{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} x_i y_i \sum_{i=1}^{m} x_i}{m(\sum_{i=1}^{m} x_i^2) - (\sum_{i=1}^{m} x_i)^2}.$$

$$a_1 = \frac{m \sum_{i=1}^{m} x_i y_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{m(\sum_{i=1}^{m} x_i^2) - (\sum_{i=1}^{m} x_i)^2}.$$

1.2 Matrix form for discrete least squares*(not required)

The function of a_0, a_1 in the linear problem can be written as

$$I(a_0, a_1) = \left\| \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdots & \cdots \\ 1 & x_m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{pmatrix} \right\|^2 .$$

The general **least squares problem** is similar (including the problem (1) where one should redefine $\tilde{y}_i = \sqrt{w_i} y_i$):

Given matrix A and vector b, find a vector x such that

$$I(x) := \|Ax - b\|^2$$

is minimized.

Taking the gradient of I, one has

$$\nabla I = 2A^T(Ax - b) = 0$$

Hence, the least squares solution solves

$$A^T A x = A^T b.$$

Using QR decomposition (discussed later), one can obtain some efficient solvers for this.

1.3 Discrete Fourier transform and FFT*(Not required)

We now consider the complex-valued function cases. For complex functions, the inner product is given by

$$\langle u, v \rangle = \sum_{j=1}^{N} u_i \bar{v}_j.$$

Let $\{\phi^j\}_{j=1}^N$ be orthogonal basis functions under the complex inner product. We first say for complex orthogonal basis functions, the optimal coefficient is still given by

$$c_j = \langle f, \phi^j \rangle / \|\phi^j\|^2 = \sum_{j=1}^{N} f_i \bar{\phi}_i^j / \|\phi^j\|^2.$$

In fact, Consider the approximation

$$S = \sum_{j=1}^{N} c_j \phi^j.$$

$$I(c_2, \cdots, c_N) = \|f - S\|^2 = \sum_{i=1}^{N} |f_i - \sum_{k=1}^{N} c_k \phi_{k,i}|^2.$$

We need to derive $\{c_j\}$. At the local minimum, can we do $\partial I / \partial c_j = 0$? No! The reason is that c_j is a complex number and this derivative does not make sense unless the function is analytic in c_j. Here, clearly, it is not.

To find the conditions for the minimum, we set $a_j = \mathrm{Re}(c_j)$ and $b_j = \mathrm{Im}(c_j)$. Then,

$$\frac{\partial I}{\partial a_j} = 0 \Rightarrow \langle f - S, -\phi_j \rangle + \langle -\phi^j, f - S \rangle = 0.$$

Hence, $2\mathrm{Re}(\langle f - S, \phi^j \rangle) = 0$. Similarly, taking the derivative on the imaginary part, you get the imaginary part equals zero. This means the conditions is still

$$\langle f - S, \phi^j \rangle = 0$$

This implies

$$c_j = \langle f, \phi^j \rangle / \|\phi^j\|^2.$$

We focus on periodic functions on $[0, 2\pi)$, with discrete points $x_j = \frac{2\pi}{N} j, j = 0, \cdots, N-1$

It is easily verified that for such domain the basis

$$\phi^k := \{e^{ikx_j}\}_{p=0}^{N-1}, \; k = -\frac{N}{2} + 1, \cdots, \frac{N}{2},$$

forms orthogonal basis, where assume N to be even for convenience.

Then, the best approximation coefficient is given by

$$c_k = \langle f, \phi^k \rangle / \|\phi^k\|^2 = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-ikx_j}$$

$\{c_k\}$ is called the Discrete Fourier Transform, or DFT for short.

In fact, the most often form of DFT is

$$\hat{f}_k := N c_k = \sum_{j=0}^{N-1} f_j e^{-ikx_j}.$$

Computing DFT directly costs $O(N^2)$. However, there is an algorithm that takes $O(N \log N)$ to compute. This algorithm is called the Fast Fourier Transform (FFT).

The most frequently used one is the Cooley-Tukey algorithm (the algorithm was independently discovered also by Gauss).

The idea is based on the simple fact. Let N be even, then

$$\sum_{n=1}^{N} u_n e^{-ikn\frac{2\pi}{N}} = \sum_{m=1}^{N/2} u_{2m} e^{-ikm\frac{2\pi}{(N/2)}} + e^{ik\frac{2\pi}{N}} \sum_{m=1}^{N/2} u_{2m-1} e^{-ikm\frac{2\pi}{(N/2)}}$$

The DFT of an array of size N is reduced to 2 DFT of arrays with size $N/2$ plus extra N operations. By this way, the whole complexity is $O(N \log N)$.

2 Rational approximation

We have seen that using polynomials to interpolate or approximate may cause oscilation. In particular, the high order polynomial interpolation often has Runge's phenomenon. Hence, we may seek rational functions to distribution the error more evenly on the approximating interval. Another advantage is that rational functions have larger ability: for example, for a function that may blow up near but outside the boundary of the approximating interval, the rational functions may often give better results since it can have poles.

One may use the rational functions of the following form

$$R_{nm}(x) = \frac{P_n(x)}{Q_m(x)} = \frac{\sum_{k=0}^{n} p_k x^k}{\sum_{k=0}^{m} q_k x^k}.$$

Given a function $f(\cdot)$, one can expand it around $x = a$ by Taylor expansion

$$f(x) = \sum_{k=0}^{N} \frac{f^{(k)}(a)}{k!}(x-a)^k + r_N(x).$$

Then, use R_{nm} to approximate. This will give the so-called Padé approximation. Below, we focus on $a = 0$.

Similarly, one can also use the following form, where T_k's are the Chebyshev polynomials.

$$\tilde{R}_{nm} = \frac{\sum_{k=0}^{m} p_k T_k(x)}{\sum_{k=0}^{m} q_k T_k(x)}$$

Then, one can also expand $f(x)$ in terms of $T_k(x)$

$$f(x) \sim \sum_{k=0}^{\infty} a_k T_k(x),$$

and then identify p_k, q_k. This gives us the Chebyshev approximation.

Even though R_{nm} and $\tilde{R}_{nm}$ are mathematically equivalent, the conditions to find the coefficients are different, so they will yield different rational approximations: the Chebyshev approximation tends to be more uniformly accurate.

Below, we only look at the Padé approximation briefly.

2.1 Padé approximation

The conditions for Padé approximation is that

$$R_{nm}^{(k)}(0) = f^{(k)}(0). \tag{2}$$

Without loss of generality, we can impose

$$q_0 = 1.$$

One can find that

$$f(x) - R_{nm}(x) = \frac{f(x)\sum_{k=0}^{m} q_k x^k - \sum_{k=0}^{n} p_k x^k}{\sum_{k=0}^{m} q_k x^k}$$

We do Taylor expansion of f and get

$$f(x) \sim \sum_{i=0}^{\infty} a_i x^i.$$

The conditions means that $f - R_{nm}$ has $(N+1)$th zeros. Hence, N coefficients of x^i must be zero.

The coefficients of $f(x)\sum_{k=0}^{m} q_k x^k$ is given by

$$\tilde{q}_k = \sum_{i=0}^{k} a_i q_{k-i}.$$

This is in fact called **convolution** in mathematics. Of course, when the index of q exceeds m, we set it to be zero. The FFT can be used to compute convolution in $O(N \log N)$ time. However, the issue is that we do not know

p_k as well, so the FFT seems not available to find the Padé approximation directly.

Hence, the conditions are

$$\sum_{i=0}^{k} a_i q_{k-i} = p_k,\ \ k = 0, \cdots, N := n + m.$$

Since $q_0 = 1$, we in fact have

$$a_k + \sum_{i=0}^{k-1} a_i q_{k-1} = p_k.$$

For $k \geq n+1$, $p_k = 0$. Hence, one has the following equations

$$a_k + \sum_{i=0}^{k-1} a_i q_{k-1} = 0,\ \ k \geq n+1.$$

There are m equations. There are m unknowns, so we can solve $q_j, j = 1, \cdots, m$ uniquely.

After these q are solved, one can use the first n equations to find p_k. Here, you may use FFT to speed up computation if n is large.

Example Find the Padé approximation $R_{2,2}$ for $f(x) = \ln(1+x)$ near $x_0 = 0$.

We need powers up to $N = 2 + 2$. By Taylor expansion,

$$\ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^2 - \frac{1}{4}x^4 + \cdots.$$

For $k \geq 3$, we have

$$\frac{1}{3} + (-\frac{1}{2}q_1 + q_2) = 0. -\frac{1}{4} + (\frac{1}{3}q_1 - \frac{1}{2}q_2) = 0$$

This solves $q_1 = 1$, $q_2 = \frac{1}{6}$ (we know $q_0 = 1$).

Then, for $k \leq 2$, we have

$$p_0 = a_0 = 0,$$

$$p_1 = a_1 + a_0 q_0 = 1,$$

$$p_2 = a_2 + a_1 q_0 + a_0 q_1 = \frac{1}{2}.$$

Hence,

$$R_{2,2} = \frac{x + \frac{1}{2}x^2}{1 + x + \frac{1}{6}x^2}.$$

2.2 Continued-fraction*(Not required)

The continued fraction was often used in the old days when computer resource was not enough. Nowadays, it is not so frequently used.