Note on Chebyshev Regression

Makoto Nakajima, UIUC
January 2006

1 Introduction

The family of Chebyshev polynomials is by far the most popular choice for the base functions for weighted residuals method. In particular, Chebyshev collocation or Chebyshev regression method is widely used for its convenience and a nice property regarding its error. In order to learn the method, we start with the properties of the Chebyshev polynomials, and learn the algorithm to implement Chebyshev collocation or regression.

2 The Definition

Chebyshev polynomials are defined over $[-1, 1]$, A Chebyshev polynomial of order i can be defined by the following closed form:

$$T_i(x) = cos(i\ cos^{-1}x)$$

However, the definition is not a friendly to compute. There is a recursive formulation as well, which enables a simple calculation of higher order Chebyshev polynomials:

$$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x)$$

Let's derive the Chebyshev polynomials of different order. The first two $i = 0$ and $i = 1$ can be easily obtained from the closed-form. From $i = 2$ on, we can use the recursive form to compute. We can obtain the followings:

$$\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
T_3(x) &= 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x \\
T_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1 \\
T_7(x) &= 64x^7 - 112x^5 + 56x^3 - 7x
\end{aligned}$$

In general, a Chebyshev polynomial of order i at x_0 ($T_i(x_0)$) can be evaluated using the following algorithm:

Algorithm 1 (Evaluating Chebyshev polynomial of order i)

1. *Suppose we want to evaluate Chebyshev polynomial of order i at x_0 ($T_i(x_0)$).*

2. *The first two order of Chebyshev polynomials can be easily evaluated using the following form:*

$$T_0(x_0) = 1$$

$$T_1(x_0) = x_0$$

3. *The Chebyshev polynomial of order $i > 1$ can be computed using the values of Chebyshev polynomials of order $i-1$ and $i-2$ and the following recursive formula:*

$$T_i(x_0) = 2x_0T_{i-1}(x_0) - T_{i-2}(x_0)$$

4. *We only need to apply the formula up to the order i to evaluate Chebyshev polynomial of order i at x_0.*

3 The Properties of Chebyshev Polynomials

1. **Range:** $T_n(x) \in [-1, 1]$. For odd n, $T_n(-1) = -1$ and $T_n(1) = 1$. For even n, $T_n(-1) = 1$ and $T_n(1) = 1$.

2. **Symmetry:** For odd n, $T_n(x)$ is an odd function, *i.e.*, $T_n(-x) = -T_n(x)$. For even n, $T_n(x)$ is an even function, *i.e.*, $T_n(-x) = T_n(x)$.

3. **Roots:** $T_n(x)$ has n distinct roots in $[-1, 1]$. The roots $\{x_i\}_{i=1}^n$ have the following expression:

$$x_i = -\cos\left(\frac{(2i-1)\pi}{2n}\right)$$

4. **Extrema:** $T_n(x)$ has $n+1$ distinct extrema in $[-1, 1]$. All extrema give the value of either -1 or 1. The extrema $\{y_i\}_{i=0}^n$ have the following expression:

$$y_i = \cos\left(\frac{i\pi}{n}\right)$$

5. **Orthogonality:** The family of Chebyshev polynomials is orthogonal with the following Chebyshev weighting function:

$$w(x) = (1 - x^2)^{-\frac{1}{2}}$$

Orthogonality implies that, for two polynomials of the family $T_i(x)$ and $T_i(x)$ ($i \neq j$):

$$\int_{\underline{x}}^{\overline{x}} T_i(x)T_j(x)w(x)dx = 0$$

In addition, in case $i = j = 0$:

$$\int_{\underline{x}}^{\overline{x}} T_i(x)T_j(x)w(x)dx = \pi$$

In case $i = j \neq 0$:

$$\int_{\underline{x}}^{\overline{x}} T_i(x)T_j(x)w(x)dx = \frac{\pi}{2}$$

6. **Discrete Orthogonality:** If $\{x_i\}_{i=1}^n$ are the n roots of Chebyshev polynomial of order n ($T_n(x)$), then for all $i, j \leq n$:

$$\sum_{k=1}^{n} T_i(x_k)T_j(x_k) = \begin{cases} 0 & i \neq j \\ \frac{n}{2} & i = j \neq 0 \\ n & i = j = 0 \end{cases}$$

This property makes the Chebyshev collocation very easy to compute and thus attractive.

4 Chebyshev Collocation

Suppose we want to approximate a function $f(x)$ which is defined over $[-1, 1]$, using Chebyshev polynomials. Let's fix the order to n. Then the approximating function takes the following form:

$$\tilde{f}(x) = \sum_{i=0}^{n} \theta_i T_i(x)$$

where $T_i(x)$ is the Chebyshev polynomial of order i, and θ_i is the coefficient (weight) associated with the Chebyshev polynomial of order i. Now the problem is to choose $n+1$ coefficients $\theta = \{\theta_i\}_{i=0}^n$ such that the error between $f(x)$ and $\tilde{f}(x)$ is minimized using a certain criteria.

It is known that, if we use collocation method, and use the roots of Chebyshev polynomial of order $n+1$ (there are $n+1$ roots), the computation of the coefficients become very easy. We will show that now.

Let's call the roots of order $m = n+1$ Chebyshev polynomial as $\{z_i\}_{i=1}^m$. We know that the roots take the following form:

$$z_i = -\cos\left(\frac{(2i-1)\pi}{2m}\right)$$

Notice that minus is put at the top of the expression in order to achieve that $\{z_i\}_{i=1}^m$ are in an ascending order.

Using collocation method, the condition associated with z_i is the following:

$$\tilde{f}(z_i) - f(z_i) = 0$$

Or

$$\sum_{k=0}^{n} \theta_k T_k(z_i) - f(z_i) = 0$$

If we pick some $0 \leq j \leq n$ and multiply both sides by $T_j(z_i)$, we obtain:

$$\sum_{k=0}^{n} \theta_k T_k(z_i) T_j(z_i) - T_j(z_i) f(z_i) = 0$$

Sum up across $i = 1, 2, ..., m$, we get:

$$\sum_{i=1}^{m} \sum_{k=0}^{n} \theta_k T_k(z_i) T_j(z_i) = \sum_{i=1}^{m} T_j(z_i) f(z_i)$$

From the discrete orthogonality property, the terms in the left hand side where $k \neq j$ disappears (equal to zero). Therefore, we get:

$$\sum_{i=1}^{m} \theta_j T_j(z_i) T_j(z_i) = \sum_{i=1}^{m} T_j(z_i) f(z_i)$$

Using the discrete orthogonality property, for $j = 0$, we get:

$$\theta_0 m = \sum_{i=1}^{m} T_0(z_i) f(z_i)$$

Or

$$\theta_0 = \frac{1}{m} \sum_{i=1}^{m} T_j(z_i) f(z_i) = \frac{1}{m} \sum_{i=1}^{m} f(z_i)$$

For $0 < j \leq n$, we get:

$$\theta_j \frac{m}{2} = \sum_{i=1}^{m} T_j(z_i) f(z_i)$$

This is equivalent to:

$$\theta_j = \frac{2}{m} \sum_{i=1}^{m} T_j(z_i) f(z_i)$$

This means that the coefficients associated with Chebyshev collocation are extremely easy to obtain.

5 Generalization: Chebyshev Regression

The Chebyshev collocation method is known to be easily extended to the case where more points than the maximum order of Chebyshev polynomials are used. Since we will have more conditions than the number of coefficients, the method is called Chebyshev regression. Conversely, Chebyshev collocation that we learned can be understood as the exactly identified case of the Chebyshev regression.

Suppose we want to approximate a function $f(x)$ which is defined over $[-1, 1]$, using Chebyshev polynomials. Let's fix the order of Chebyshev polynomials to n. Then the approximating function takes the following form:

$$\tilde{f}(x) = \sum_{j=0}^{n} \theta_j T_j(x)$$

where $T_j(x)$ is the Chebyshev polynomial of order j, and θ_j is the coefficient (weight) associated with the Chebyshev polynomial of order j. Now the problem is to choose $n+1$ coefficients $\theta = \{\theta_j\}_{j=0}^n$ such that the error between $f(x)$ and $\tilde{f}(x)$ is minimized using a certain criteria.

Let's choose m, the number of conditions that we use. In order to at least exactly identify the coefficients (this case is the Chebyshev collocation) we need to have $m \geq n+1$. Once we pick m, denote the roots of order m Chebyshev coefficients as $\{z_i\}_{i=1}^m$. We know that the roots take the following form:

$$z_i = -\cos\left(\frac{(2i-1)\pi}{2m}\right)$$

What we want to do is to minimize the sum of squared error between the true function and the approximating function, both of which are evaluated at $\{z_i\}_{i=1}^m$. More formally:

$$\theta_j = \operatorname{argmin} \sum_{i=1}^m \left\{ f(z_i) - \tilde{f}(z_i) \right\}^2$$

Or

$$\theta_j = \operatorname{argmin} \sum_{i=1}^m \left\{ f(z_i) - \sum_{k=0}^n \theta_k T_k(z_i) \right\}^2$$

Or

$$\theta_j = \operatorname{argmin} \sum_{i=1}^m \left\{ f(z_i) - \sum_{k=0}^n \theta_k T_k(z_i) \right\}^2$$

Take the first order condition, we obtain:

$$\theta_j = \frac{\sum_{i=1}^m f(z_i) T_j(z_i)}{\sum_{i=1}^m T_j(z_i) T_j(z_i)}$$

Using the discrete orthogonality property, we get:

$$\theta_0 = \frac{1}{m} \sum_{i=1}^m f(z_i)$$

$$\theta_j = \frac{2}{m} \sum_{i=1}^m f(z_i) T_j(z_i) \qquad j = 1, 2, ..., n$$

This is exactly the same formula that we obtained for Chebyshev collocation.

6 (Rough) Discussion on Near-Minimax Property

So we learned that Chebyshev approximation (collocation or regression) is very easy to obtain. That is an appeal of Chebyshev polynomials. But there is more important property of Chebyshev polynomials.

It can be shown that, as long as we are approximating a smooth function, Chebyshev approximation is close to the so-called minimax polynomial, whose approximating function minimizes the largest deviation from the original function among the polynomials of the same order. The minimax polynomial is usually very hard to compute, but Chebyshev polynomial gives a close alternative with a very small computational cost. This is what makes Chebyshev polynomial very popular among polynomials used for weighted residual methods.

7 Collocation Points Conversion

Before summarizing the algorithm, we need to take care of one thing. So far we assume that the domain of the objective function is same as the domain of the Chebyshev polynomials, namely $[-1, 1]$. But, of course, it is not always the case.

Suppose the function $f(x)$ is defined over $[\underline{x}, \overline{x}]$. Then, we can easily convert a point $z \in [-1, 1]$ to a corresponding point $x \in [\underline{x}, \overline{x}]$ using the following conversion formula:

$$x = \frac{z+1}{2}(\overline{x} - \underline{x}) + \underline{x}$$

Or, solving for z, we get the formula to convert from x to z:

$$z = \frac{2(x - \underline{x})}{\overline{x} - \underline{x}} - 1$$

This is the simplest way to convert an arbitrary compact domain into the domain of Chebyshev polynomials. However, there is another way. The other way is motivated by the property of the roots of Chebyshev polynomials. In particular, it is easy to see that roots of Chebyshev polynomials, which are used as collocation points for Chebyshev collocation, are always internal points in $[-1, 1]$. On the other hand, we might want to guarantee that the original function and the approximating function takes the same value at end points. This is not generally true if we use the conversion formula above. We are sort of extrapolating near end points if we use standard conversion formula. So how we can modify the conversion formula?

Suppose we are using m collocation points $\{z_i\}_{i=1}^m$. These collocation points have the following formula, because these are the roots of order m Chebyshev polynomial:

$$z_i = -\cos\left(\frac{(2i-1)\pi}{2m}\right)$$

And $\{z_i\}_{i=1}^m$ satisfies:

$$-1 < z_1 < z_2 < ... < z_{m-1} < z_m < 1$$

Now, notice that:

$$z_1 = -\cos\left(\frac{\pi}{2m}\right)$$

$$z_m = -\cos\left(\frac{(2m-1)\pi}{2m}\right) = -z_1$$

Therefore, by using the following formula to convert from z to x, we can convert z_1 into $\underline{x}$ and z_m into $\overline{x}$:

$$x = \frac{\frac{1}{\cos\left(\frac{\pi}{2m}\right)}z + 1}{2}(\overline{x} - \underline{x}) + \underline{x}$$

Or

$$x = \frac{\sec\left(\frac{\pi}{2m}\right)z + 1}{2}(\overline{x} - \underline{x}) + \underline{x}$$

This means that we can convert $\underline{x}$ into z_1 and $\overline{x}$ into z_m using the following conversion formula:

$$z = \frac{1}{\sec\left(\frac{\pi}{2m}\right)}\left(\frac{2(x - \underline{x})}{\overline{x} - \underline{x}} - 1\right)$$

The set of collocation points $\{x_i\}_{i=1}^m$ which are constructed using the algorithm above satisfies the following:

$$\underline{x} = x_1 < x_2 < ... < x_{m-1} < x_m = \overline{x}$$

and is called *extended Chebyshev array.*

8 Summary: Chebyshev Regression Algorithm with Extended Array

Now we are ready to specify the algorithm of Chebyshev regression (collocation is a special case of it) using extended Chebyshev array.

Algorithm 2 (Chebyshev Regression with Extended Array)

1. *We want to approximate a function $f(x)$ defined over $[\underline{x}, \overline{x}]$.*
2. *Pick the order of Chebyshev polynomial n. It means that we are approximating $f(x)$ by:*

$$\tilde{f}(x) = \sum_{j=0}^{n} \theta_j T_j(x)$$

3. *Pick the number of collocation points $m \geq n + 1$.*

4. *With m chosen, the collocation points on $[-1,1]$ ($\{z_i\}_{i=1}^m$) are the roots of order m Chebyshev polynomial, which take the following form:*

$$z_i = -\cos\left(\frac{(2i-1)\pi}{2m}\right)$$

5. *Compute corresponding collocation points over $[\underline{x}, \overline{x}]$. As we are going to use the extended array, $\{x_i\}_{i=1}^m$ are computed using the following formula:*

$$x = \frac{\sec\left(\frac{\pi}{2m}\right) z + 1}{2}(\overline{x} - \underline{x}) + \underline{x}$$

6. *Chebyshev regression method gives the following simple solution formula for $\{\theta_j\}_{j=0}^n$:*

$$\theta_0 = \frac{1}{m}\sum_{i=1}^{m} f(x_i)$$

$$\theta_j = \frac{2}{m}\sum_{i=1}^{m} f(x_i)T_j(z_i) \qquad j = 1, 2, ..., n$$

Notice that $f(x)$ is evaluated using $\{x_i\}_{i=1}^m$ and $T_j(z)$ is evaluated using $\{z_i\}_{i=1}^m$.

7. *Once we obtain $\{\theta_j\}_{j=0}^n$, we can easily evaluate the approximating function $\tilde{f}(x)$. For example, suppose we want to evaluate $\tilde{f}(x)$ at x_0.*

8. *Convert x_0 to z_0 using the following formula:*

$$z = \frac{1}{\sec\left(\frac{\pi}{2m}\right)}\left(\frac{2(x - \underline{x})}{\overline{x} - \underline{x}} - 1\right)$$

9. *We can easily obtain Chebyshev polynomial of order 0 and 1:*

$$T_0(z_0) = 1$$

$$T_1(z_0) = z_0$$

10. *Use the following recursive formula to evaluate Chebyshev polynomials up to order n:*

$$T_i(z_0) = 2z_0 T_{i-1}(z_0) - T_{i-2}(z_0)$$

11. *Now we can evaluate $\tilde{f}(x)$ as follows:*

$$\tilde{f}(x_0) = \sum_{j=0}^{n} \theta_j T_j(z_0)$$

Below are some final remarks:

1. The coefficients associated with Chebyshev polynomials are strictly decreasing as the order increases. Looking at how the coefficients decrease helps determining the order of Chebyshev polynomials n used for approximation.

2. As long as the number of collocation points m is fixed, reducing n does not affect the value of coefficients. This property is called Chebyshev economization.

3. Chebyshev polynomial is not always the best one, especially when we are approximating a non-smooth function. For example, it is known that the performance of the Chebyshev approximation is not great if the original function has a kink.