**АППРОКСИМАЦИЯ ФУНКЦИЙ (ТЕОРИЯ)**

Задача приближения (аппроксимации) функций заключается в том, чтобы для данной функции построить другую, отличную от нее функцию, значения которой достаточно близки к значениям данной функции.

В основном аппроксимация функции применяется в случае, если:

1. Функция задана таблицей в конечном множестве точек, а вычисления нужно произвести в других точках.

2. Функция задана аналитически, но ее вычисление по формуле затруднительно.

При решении задачи поиска приближенной функции возникают следующие проблемы:

1. Необходимо выбрать вид приближенной функции. Для приближения широко используются многочлены, тригонометрические функции, показательные функции и т. д.

2. Необходимо выбрать критерий близости исходной и приближенной функции. Это может быть требование совпадения обеих функций в узловых точках (задача интерполяции), минимизация среднеквадратического уклонения (метод наименьших квадратов) и др.

3. Необходимо указать правило (алгоритм), позволяющее с заданной точностью найти приближение функции.

**Теорема Вейерштрасса.** Если функциянепрерывна на отрезке , то для любого существует многочлен степени абсолютное отклонение которого от функции на отрезке меньше .

Т.е. любую функцию можно как угодно точно аппроксимировать многочленом, но теорема ничего не говорит, ни о способах нахождения этого многочлена, ни о количестве точек, ни об их расположении.

Определение аппроксимирующей функции представляет собой задание вида функции и нахождение ее коэффициентов. При аппроксимации многочленами предварительно задаются степенью многочлена и находят его коэффициенты. При этом отклонение от должно быть наименьшим.

**Метод наименьших квадратов**

Метод наименьших квадратов–*среднеквадратичное приближение* функций с помощью многочлена:

*.*

На практике стараются подобрать аппроксимирующий многочлен как можно меньшей степени, обычно

Мерой отклонения многочлена от заданной функции на множество точек () при среднеквадратичном приближении является , равная сумме квадратов разностей между значениями многочлена и функции в заданных точках:

Коэффициенты многочлена надо подобрать так, чтобы величина была минимальной. В этом состоит *метод наименьших квадратов*.

Запишем сумму квадратов отклонений для всех точек:

Коэффициенты надо определить из условий минимума функции:

.

Минимум функции найдем, приравнивая нулю частные производные по этим переменным:

Эти соотношения являются системой уравнений для определения . Найдем частные производные функции:

…

Преобразуем систему и получим

;

…

Данная система называется *системой нормальных уравнений*. Для того чтобы найти коэффициенты, надо задать вид функции .

**Линейная аппроксимация.**

Зададим аппроксимирующую функцию как линейную:

;

Подставим значения функции и производных в систему и получим:

Вынесем коэффициенты за знак суммы, получим:

Обозначим:

; ; ; ;

Тогда систему можно переписать:

Применив формулу Крамера, получим

где – число точек (пары).

В итоге получаем формулу линейной аппроксимации:

.

**Квадратичная аппроксимация.**

Зададим аппроксимирующую функцию в виде квадратного трехчлена:

Подставим функцию и ее производные в систему, получим

Вынесем коэффициенты за знак суммы:

Мы получили систему линейных уравнений, поскольку в качестве аппроксимирующей функции выбрали многочлен. Если бы мы выбрали не многочлен,мы бы получили систему нелинейных уравнений.Теперь решим эту систему методом Крамера и найдем коэффициенты .

Для удобства можно ввести обозначения:

В этой системе все суммы (, … ) являются константами, а все неизвестные –это коэффициенты**.**

В итоге получим формулу квадратичной аппроксимации:

.

**ПРАКТИКА**

**Аппроксимация функций. Метод наименьших квадратов**

**Цель работы:**изучить метод наименьших квадратов, линейную и квадратичную аппроксимации, алгоритм метода, формулы для вычисления, написать программу на языке программирования, реализующую линейную и квадратичную аппроксимации методом наименьших квадратов.

***Ход решения для линейной аппроксимации*:**

1. Определить по выше описанным формулам.
2. Вычислить.
3. Задать любое значение и вывести на экран результаты: аппроксимирующую линейную функцию *.*

***Ход решения для квадратичной аппроксимации:***

1. Определить по вышеописанным формулам.
2. Вычислить .
3. Задать любое значение и вывести на экран результаты: аппроксимирующую квадратичную функцию.

*В формулу линейной и квадратичной аппроксимации можно подставить любое значение , вычислить значение аппроксимирующей функции и сравнить его с исходными данными.*

***Замечание****.* Результаты линейной и квадратичной аппроксимации должны быть приближенно равными.

ВАРИАНТЫ ЗАДАНИЙ

Вариант1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Вариант 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Вариант 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Вариант4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Вариант5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Вариант6

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

Вариант 7

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | 27 | 25 | 23 | 21 | 19 | 17 | 15 | 13 | 11 |

Вариант 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 4 | 5,5 | 7,7 | 9,4 | 11,3 | 13,2 | 14,9 | 16,9 |
|  | 15 | 11 | 7 | 3 | -1 | -5 | -9 | -13 |

Вариант 9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 4 | 17 | 23,3 | 31,3 | 39,3 | 47,3 | 55,3 | 63,3 |
|  | 1,1 | 2,2 | 3,3 | 4,4 | 5,5 | 6,6 | 7,7 | 8,8 | 9,9 |

Вариант 10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 1,5 | 2 | 2,5 | 3 | 3,5 | 4 | 4,5 | 5 |
|  | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |

**Реализация алгоритмов на языке C#**

Для реализации алгоритмов линейной и квадратичной аппроксимацийнеобходимо произвести немного больше вычислений. Для удобства необходимо создать массив данных, представляющий собой набор элементов одного и того же типа объединённых общим именем. Как показано ниже, создадим фиксированный массив для ряда и для ряда :

//созданиемассива

int n = 8;

double[] x = newdouble[8] { 15, 12, 9, 6, 3, 0, -3, -6 };

double[] y = newdouble[8] { 1, 5.5, 11.5, 16.5, 21.75, 27, 32.25, 37.5 }.

Для линейной аппроксимации достаточно организовать всего один метод:

**//**функциядлялинейнойаппроксимации

Staticdoublefi(double a0, double a1, double x)

{

Returna0 + a1 \* x;

}

В основной части программы задаются начальные значения переменных, необходимых для реализации алгоритма, вычисляются значения неизвестных с помощью циклов с параметром FOR, и организуется вывод каждого значения и функции

Console.WriteLine(«\n Линейная аппроксимация: \n»);

doubles1 = 0, s2 = 0, s3 = 0, s4 = 0;

**//**вычисление s1-s4

for (inti = 0; i<x.Length; i++)

{

s1 += x[i];

s3 += Math.Pow(x[i], 2);

}

for (inti = 0; i<y.Length; i++)

{

s2 += y[i];

s4 += x[i] \* y[i];

}

**//**вычислениеа0, a1

Double a0 = (s2 \* s3 - s1 \* s4) / (n \* s3 - Math.Pow(s1, 2));

Double a1 = (n \* s4 - s1 \* s2) / (n \* s3 - Math.Pow(s1, 2));

**//**выводнаэкранвычисленныхзначений

Console.WriteLine("s1= {0}, s2= {1}, s3= {2}, s4= {3}", s1, s2, s3, s4);

Console.WriteLine("a0= {0}, a1= {1}", a0, a1);

**//**выводнаэкранзначенияфункции

For (inti = 0; i<x.Length; i++)

{

Console.WriteLine("x = " + x[i] + "\tf(x) = " + fi(a0, a1, x[i]));

}

Для квадратичной аппроксимации алгоритм решения задачи строится аналогичным образом. При этом значения можно вычислить с помощью цикла с параметром:

for (inti = 0; i<x.Length; i++)

{

c4 += Math.Pow(x[i], 3);

c6 += Math.Pow(x[i], 4);

c7 = c7 + (x[i] \* x[i] \* y[i]);

}

Поскольку мы решаем задачу согласно методу Крамера, то для вычисления каждой из дельт необходимо описать несколько дополнительных методов. Ниже представлен программный код методов вычисления и , при этом нахождение остальных дельт осуществляется подобным образом.

static double delta(double c1, double c2, double c4, double c6)

{

return 8 \* (c2 \* c6 - Math.Pow(c4, 2)) - c1 \* (c1 \* c6 - c2 \* c4) + c2 \*

\* (c1 \* c4 - Math.Pow(c2, 2));

}

Staticdouble delta1(double c1, double c2, double c3, double c4, double c5, doublec6, double c7)

{

return(c3 \* c2 \* c6 + c5 \* c4 \* c2 + c7 \* c1 \* c4 - c2 \* c2 \* c7 –

–c4 \* c4 \* c3 - c6 \* c1 \* c5);

}

Далее можно вычислить значения :

Double A0 = delta1 (c1, c2, c3, c4, c5, c6, c7) / delta (c1, c2, c4, c6);

Double A1 = delta2 (c1, c2, c3, c4, c5, c6, c7) / delta (c1, c2, c4, c6);

DoubleA2 = delta3 (c1, c2, c3, c4, c5, c6, c7) / delta (c1, c2, c4, c6).

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Аппроксимация. Основная задача аппроксимации. Случаи ее применения.

2. Линейная аппроксимация. Квадратичная аппроксимация.

3. Функции, применяющиеся в качестве аппроксимирующих.

4. Преимущества и недостатки аппроксимации.

5. Способы повышения точности аппроксимации.