

АППРОКСИМАЦИЯ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ С ПОМОЩЬЮ ЯЗЫКА C#

А.В. Полуэктов¹, А.В. Сухарский¹, Ф.В. Макаренко¹

¹ФГБОУ ВО «Воронежский государственный лесотехнический университет
имени Г.Ф. Морозова»

Аннотация. В статье рассматриваются теоретические основы и практические аспекты аппроксимации экспериментальных данных с использованием языка программирования C#. Приведены детальные реализации алгоритмов линейной и полиномиальной аппроксимации как на чистом C# без внешних зависимостей, так и с применением мощной библиотеки численных вычислений Math.NET Numerics. Особое значение в работе придаётся вопросам оценки точности и надёжности полученных моделей. Для каждого рассмотренного метода предложены алгоритмы и готовые программные решения на C# для количественной оценки этих погрешностей, включая расчёт стандартных ошибок коэффициентов, доверительных интервалов, коэффициента детерминации R^2 , а также использование QR- и SVD-разложений для повышения численной устойчивости.

Ключевые слова: аппроксимация, метод наименьших квадратов, C#, Math.NET Numerics, линейная регрессия, полиномиальная регрессия, погрешность аппроксимации, стандартная ошибка, коэффициент детерминации, обусловленность, переобучение, QR-разложение, SVD.

APPROXIMATION OF EXPERIMENTAL DATA USING C#

A.V. Poluektov¹, A.V. Sukharskiy¹, F.V. Makarenko¹

¹Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. This article examines the theoretical foundations and practical aspects of approximating experimental data using the C# programming language. Detailed implementations of linear and polynomial approximation algorithms are provided, both in pure C# without external dependencies and utilizing the powerful numerical computing library Math.NET Numerics. Particular emphasis is placed on assessing the accuracy and reliability of the resulting models. For each method considered, algorithms and ready-to-use C# software solutions are proposed for the quantitative evaluation of approximation errors, including the calculation of standard errors of coefficients, confidence

intervals, the coefficient of determination R^2 , as well as the application of QR and SVD decompositions to enhance numerical stability.

Keywords: approximation, least squares method, C#, Math.NET Numerics, linear regression, polynomial regression, approximation error, standard error, coefficient of determination, condition number, overfitting, QR decomposition, SVD.

Введение

В научных исследованиях, инженерных расчётах и при анализе реальных процессов часто возникает задача восстановления функциональной зависимости по набору экспериментальных точек. Такая задача называется аппроксимацией. Одним из самых распространённых методов аппроксимации является метод наименьших квадратов (МНК), позволяющий подобрать параметры модели так, чтобы сумма квадратов отклонений между моделью и исходными данными была минимальной.

Аппроксимация данных с помощью языка C#

Язык программирования C#, благодаря своей мощной экосистеме (.NET), удобству работы с массивами и возможностям интеграции с библиотеками численных вычислений, отлично подходит для реализации алгоритмов аппроксимации [1].

Постановка задачи. Пусть даны экспериментальные данные в виде пар (x_i, y_i) , где $i = 0, 1, \dots, n - 1$. Требуется найти функцию $f(x)$ (например, полином степени m), которая наилучшим образом описывает эти данные в смысле минимума суммы квадратов ошибок:

$$S = \sum_{i=0}^{n-1} (f(x_i) - y_i)^2 \rightarrow \min$$

Линейная аппроксимация (полином первой степени). Для простоты начнём с линейной зависимости:

$$f(x) = ax + b$$

Коэффициенты a и b находятся по формулам МНК:

$$a = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2},$$
$$b = \frac{\sum y_i - a \sum x_i}{n}$$

Реализация на C#:

```
public static (double a, double b) LinearRegression(double[] x, double[] y)
```

```

{
    if (x.Length != y.Length || x.Length == 0)
        throw new ArgumentException("Некорректные входные данные");
    int n = x.Length;
    double sumX = 0, sumY = 0, sumXY = 0, sumX2 = 0;
    for (int i = 0; i < n; i++)
    {
        sumX += x[i];
        sumY += y[i];
        sumXY += x[i] * y[i];
        sumX2 += x[i] * x[i];
    }
    double denominator = n * sumX2 - sumX * sumX;
    if (Math.Abs(denominator) < 1e-10)
        throw new InvalidOperationException("Данные не позволяют построить
линейную модель");
    double a = (n * sumXY - sumX * sumY) / denominator;
    double b = (sumY - a * sumX) / n;
    return (a, b);
}

```

Пример использования:

```

double[] x = { 1, 2, 3, 4, 5 };
double[] y = { 2.1, 3.9, 6.0, 8.1, 9.9 };
var (a, b) = LinearRegression(x, y);
Console.WriteLine($"y = {a:F3} * x + {b:F3}");
// Вывод: y = 1.980 * x + 0.140

```

Полиномиальная аппроксимация (общий случай) [2]. Для аппроксимации полиномом степени m :

$$f(x) = c_0 + c_1x + c_2x^2 + \dots + c_mx^m$$

необходимо решить систему линейных уравнений вида:

$$A \cdot c = b$$

где матрица A – это матрица моментов:

$$A_{ij} = \sum_{k=0}^{n-1} x_k^{i+j}, i, j = 0..m$$

а вектор b :

$$b_i = \sum_{k=0}^{n-1} y_k x_k^i$$

Решение этой системы даёт коэффициенты полинома. Для решения системы можно использовать метод Гаусса, но на практике проще воспользоваться библиотекой Math.NET Numerics. Библиотека Math.NET Numerics – это мощный open-source инструмент для численных вычислений в .NET. Установите её через NuGet:

```
dotnet add package MathNet.Numerics
```

Пример полиномиальной аппроксимации:

```
using MathNet.Numerics.LinearAlgebra;
using MathNet.Numerics.LinearAlgebra.Double;
public static double[] PolynomialFit(double[] x, double[] y, int degree)
{
    if (x.Length != y.Length)
        throw new ArgumentException("Массивы должны быть одинаковой
длины");
    int n = x.Length;
    var A = Matrix<double>.Build.Dense(n, degree + 1);
    var b = Vector<double>.Build.Dense(y);
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j <= degree; j++)
        {
            A[i, j] = Math.Pow(x[i], j);
        }
    }
    // Решаем задачу наименьших квадратов: min ||Ax - b||
    var coefficients = A.QR().Solve(b);
    return coefficients.ToArray();
}
```

Пример использования:

```
double[] x = { 0, 1, 2, 3, 4, 5 };
double[] y = { 1.0, 1.8, 3.2, 5.1, 7.9, 11.0 };
var coeffs = PolynomialFit(x, y, 2); // Квадратичная аппроксимация
```

```
Console.WriteLine($"f(x) = {coeffs[0]:F3} + {coeffs[1]:F3}*x + {coeffs[2]:F3}*x^2");
```

Анализ погрешности аппроксимации. Для того чтобы аппроксимация была не просто формальным упражнением, а инструментом для принятия решений, необходимо количественно оценить её точность и надёжность. В случае линейной регрессии существует хорошо развитая статистическая теория, позволяющая это сделать [3].

Основные допущения. Мы предполагаем, что истинная зависимость имеет вид $y = ax + b + \epsilon$, где ϵ – случайная ошибка (шум) с нулевым средним $E[\epsilon] = 0$ и постоянной дисперсией σ^2 (гомоскедастичность). Ошибки для разных измерений независимы.

Оценка дисперсии шума [4]. Первым шагом является оценка неизвестной дисперсии σ^2 шума. Она оценивается через остаточную сумму квадратов (Residual Sum of Squares, RSS):

$$RSS = \sum_{i=1}^n (y_i - \bar{y}_i)^2,$$

где $\bar{y}_i = ax_i + b$ – предсказанное моделью значение. Поскольку мы оценили два параметра (a и b), число степеней свободы равно $n - 2$. Тогда несмещённая оценка дисперсии:

$$s^2 = \frac{RSS}{n - 2}$$

Стандартные ошибки коэффициентов. Стандартная ошибка показывает, насколько точно мы оценили каждый коэффициент. Для свободного члена b и наклона a они вычисляются по формулам:

$$SE(b) = s \sqrt{\frac{\sum x_i^2}{n \sum x_i^2 - (\sum x_i)^2}},$$

$$SE(a) = s \sqrt{\frac{n}{n \sum x_i^2 - (\sum x_i)^2}}.$$

Чем меньше стандартная ошибка, тем надёжнее оценка коэффициента.

Доверительный интервал даёт диапазон, в котором с заданной вероятностью (например, 95%) находится истинное значение параметра. Для уровня значимости $\alpha=0.05$ (95% доверительный интервал) он строится так:

$$a \in [a - t_{\alpha/2, n-2} \cdot SE(a), a + t_{\alpha/2, n-2} \cdot SE(a)],$$

$$b \in [b - t_{\alpha/2, n-2} \cdot SE(b), b + t_{\alpha/2, n-2} \cdot SE(b)],$$

где $t_{\alpha/2, n-2}$ – критическое значение t-распределения Стьюдента с $n - 2$ степенями свободы.

Коэффициент детерминации R^2 – он уже был упомянут, но теперь его можно интерпретировать статистически. R^2 показывает, какая доля дисперсии зависимой переменной y объясняется моделью. $R^2 = 1$ означает идеальную подгонку, $R^2 = 0$ – что модель не лучше, чем просто среднее значение $\bar{y}$.

Реализация на C#

```
public class LinearRegressionResult
{
    public double Slope { get; set; }
    public double Intercept { get; set; }
    public double RSquared { get; set; }
    public double SlopeStdError { get; set; }
    public double InterceptStdError { get; set; }
    public double[] Residuals { get; set; }
}

public static LinearRegressionResult DetailedLinearRegression(double[] x,
double[] y)
{
    var (a, b) = LinearRegression(x, y); // используем ранее написанный метод
    int n = x.Length;
    double yMean = y.Average();

    // Вычисляем предсказанные значения и остатки
    double[] yPred = new double[n];
    double[] residuals = new double[n];
    double ssRes = 0, ssTot = 0;
    for (int i = 0; i < n; i++)
    {
        yPred[i] = a * x[i] + b;
        residuals[i] = y[i] - yPred[i];
        ssRes += residuals[i] * residuals[i];
        ssTot += (y[i] - yMean) * (y[i] - yMean);
    }
    double rSquared = 1.0 - (ssRes / ssTot);
    // Оценка дисперсии шума
    double s2 = ssRes / (n - 2);
    double s = Math.Sqrt(s2);
}
```

```

// Суммы для вычисления стандартных ошибок
double sumX = x.Sum();
double sumX2 = x.Select(xi => xi * xi).Sum();
double denom = n * sumX2 - sumX * sumX;
double seIntercept = s * Math.Sqrt(sumX2 / denom);
double seSlope = s * Math.Sqrt(n / denom);
return new LinearRegressionResult
{
    Slope = a,
    Intercept = b,
    RSquared = rSquared,
    SlopeStdError = seSlope,
    InterceptStdError = seIntercept,
    Residuals = residuals
};
}

```

Этот расширенный класс `LinearRegressionResult` предоставляет всю необходимую информацию для глубокого анализа качества линейной модели.

Заключение

Аппроксимация экспериментальных данных – важный инструмент анализа в науке и инженерии. Язык `C#` предоставляет гибкие возможности для реализации как простых, так и сложных алгоритмов аппроксимации. Для учебных или небольших проектов достаточно собственной реализации, но для промышленного применения рекомендуется использовать проверенные библиотеки, такие как `Math.NET Numerics`, которые обеспечивают высокую точность, производительность и устойчивость к численным ошибкам. Ключевым аспектом, отличающим качественный анализ от простого подбора кривой, является тщательная оценка погрешностей и надёжности полученной модели.

Список литературы

1. Сумин, В. И. Использование ситуационного моделирования в разработке систем принятия решений для сложных организационных систем / В. И. Сумин, А. С. Кравченко, А. В. Толкачев // Моделирование систем и процессов. – 2024. – Т. 17, № 3. – С. 71-79. – DOI 10.12737/2219-0767-2024-69-77. – EDN FBBHJO.

2. Шагеев, Д. А. Поиск решения проблемы выбора формы свёртки локальных векторов с целью повышения точности измерений в методе анализа иерархий / Д. А. Шагеев // Моделирование систем и процессов. – 2024. – Т. 17, № 3. – С. 95-105. – DOI 10.12737/2219-0767-2024-93-103. – EDN FUFKOH.

3. Дмитриев, Е. А. Математическое моделирование и реализация информационно-расчетных задач «маскировка» / Е. А. Дмитриев, А. Ф. Тараканов, Е. А. Шипилова // Моделирование систем и процессов. – 2024. – Т. 17, № 2. – С. 32-42. – DOI 10.12737/2219-0767-2024-17-2-32-42. – EDN DJJKYT.

4. Шорников, С. А. Сравнение языков программирования C# и Python / С. А. Шорников // Актуальные вопросы инноваций и современные научные открытия : Сборник научных статей по материалам V Международной научно-практической конференции, Уфа, 31 мая 2024 года. – Уфа: Общество с ограниченной ответственностью "Научно-издательский центр "Вестник науки", 2024. – С. 361-364. – EDN AAAQSV.

5. Исследование способов защиты информации от утечки по акустическому и виброакустическому каналам / В. К. Зольников, С. А. Сазонова, А. И. Заревич, С. С. Башун // Моделирование систем и процессов. – 2025. – Т. 18, № 2. – С. 27–40.

6. Разработка информационной подсистемы банка по выдаче кредитов / С. А. Сазонова, В. Ф. Асминин, А. И. Заревич // Информационные технологии в строительных, социальных и экономических системах. – 2025. – № 2 (36). – С. 10–17.

7. Моделирование колебательных процессов в пакете MVStudium / А. В. Полуэктов, К. В. Зольников, В. И. Анциферова // Моделирование систем и процессов. – 2021. – Т. 14, № 4. – С. 139–148. – DOI 10.12737/2219-0767-2021-14-4-139-148. – EDN PPOUDI.

8. Математическая модель индуктивного делителя напряжения с электронной компенсацией / А. И. Заревич, С. В. Муравьев // Известия Томского политехнического университета. – 2010. – Т. 317, № 4. – С. 129–133.

9. Релятивистский магнетрон с распределенным выходом микроволнового излучения / И. И. Винтизенко, А. И. Заревич, С. С. Новиков // Письма по технической физике. – 2005. – Том 31, № 5. – С. 388-390.

References

1. Sumin, V. I. Application of Scenario Modeling in the Development of Decision Support Systems for Complex Organizational Systems / V. I. Sumin, A. S.

Kravchenko, A. V. Tolkachev // Modeling of Systems and Processes. – 2024. – Vol. 17, No. 3. – Pp. 71–79. – DOI 10.12737/2219-0767-2024-69-77. – EDN FBBHJO.

2. Shageev, D. A. Solving the Problem of Selecting the Convolution Form of Local Vectors to Improve Measurement Accuracy in the Analytic Hierarchy Process / D. A. Shageev // Modeling of Systems and Processes. – 2024. – Vol. 17, No. 3. – Pp. 95–105. – DOI 10.12737/2219-0767-2024-93-103. – EDN FUFKOX.

3. Dmitriev, E. A. Mathematical Modeling and Implementation of Information-Calculational "Masking" Tasks / E. A. Dmitriev, A. F. Tarakanov, E. A. Shipilova // Modeling of Systems and Processes. – 2024. – Vol. 17, No. 2. – Pp. 32–42. – DOI 10.12737/2219-0767-2024-17-2-32-42. – EDN DJJKYT.

4. Shornikov, S. A. Comparison of the C# and Python Programming Languages / S. A. Shornikov // Current Issues of Innovations and Modern Scientific Discoveries: Collection of Scientific Articles from the Proceedings of the V International Scientific and Practical Conference, Ufa, May 31, 2024. – Ufa: Limited Liability Company "Scientific Publishing Center 'Bulletin of Science'", 2024. – Pp. 361–364. – EDN AAAQSV.

5. Study of methods for protecting information from leakage via acoustic and vibroacoustic channels / V. K. Zolnikov, S. A. Sazonova, A. I. Zarevich, S. S. Bashun // Modeling of systems and processes. - 2025. - Vol. 18, No. 2. - P. 27-40.

6. Development of a bank information subsystem for issuing loans / S. A. Sazonova, V. F. Asminin, A. I. Zarevich // Information technologies in construction, social and economic systems. - 2025. - No. 2 (36). - P. 10-17.

7. Modeling of oscillatory processes in the MVStudium package / A. V. Poluektov, K. V. Zolnikov, V. I. Antsiferova // Modeling of systems and processes. - 2021. - Vol. 14, No. 4. - Pp. 139–148. - DOI 10.12737/2219-0767-2021-14-4-139-148. - EDN PPOUDI.

8. Mathematical model of an inductive voltage divider with electronic compensation / A. I. Zarevich, S. V. Muravyov // Bulletin of Tomsk Polytechnic University. - 2010. - Vol. 317, No. 4. - P. 129-133.

9. Relativistic magnetron with distributed output of microwave radiation / I. I. Vintzenko, A. I. Zarevich, S. S. Novikov // Technical Physics Letters. – 2005. – Vol. 31, № 5. – P. 388–390.