УДК 519.254+519.233.5+519.654

Кандидаты техн. наук В.А. Борщев, И.М. Егорова (УкрГАЖТ),
преподаватель И.Н. Гришина (Харьковский колледж транспортных технологий),
д-р физ.-мат. наук А.Н. Николенко (УкрГАЖТ)

V.A. Borschev, I.M. Yegorova,
I.N. Grishyna, A.N. Nikolenko

АЛГОРИТМ И ПРОГРАММА ЧИСЛЕННОГО ВОССТАНОВЛЕНИЯ ПАРНОЙ РЕГРЕССИИ ПРОИЗВОЛЬНОГО ОБЩЕГО ВИДА

ALGORITHM AND PROGRAM OF NUMERAL RENEWAL OF ARBITRARY GENERAL VIEW PAIR REGRESSION

Введение. При решении многих прикладных научных задач, в силу определенных сложностей в разработке математических моделей, возникает необходимость проведения физического эксперимента на предметной модели либо натурного эксперимента. Обработка полученных результатов экспериментальных исследований может включать несколько этапов: предварительный анализ и первичная обработка данных; анализ структуры и тесноты статистической связи между исследуемыми переменными − корреляционный анализ; исследование вида зависимости между количественными переменными − регрессионный анализ; дисперсионный анализ и другое. Наиболее сложным этапом, требующим значительных затрат времени и искусства исследователя, является регрессионный анализ.

Анализ последних достижений. Регрессионный анализ предусматривает: подбор класса функций, в рамках которого производится дальнейший поиск неизвестной функции регрессии; нахождение параметров (коэффициентов) функции регрессии данного класса с последующим анализом точности полученного уравнения связи.

Сформулированные рабочие гипотезы об общем виде искомой функции регрессии могут быть проверены с привлечением соответствующих математико-статистических критериев. Эти критерии базируются, прежде всего, на идее компромисса между сложностью регрессионной модели и точностью ее оценивания [1].

Оценивание неизвестных значений параметров, входящих в уравнение регрессионной зависимости, в большинстве случаев можно получить с использованием метода наименьших квадратов (МНК-оценок). В классических предположениях МНК-оценки совпадают с оценками максимального правдоподобия и являются наилучшими среди всех несмещенных оценок [1]. Существующие методы получения МНК-оценок, как правило, основаны на использовании классов линейных функций или классов функций, которые путем соответствующих преобразований могут быть приведены к классу линейных функций.

Постановка задачи. Восстановление парной регрессии произвольного общего вида $y = y(x)$, в общем случае нелинейной, по ограниченной выборке с использованием МНК-оценок может быть сведено к нахождению на множестве классов функций

$$F = \bigcup_k \left\{ f_k\left(x, \boldsymbol{a}\right)\right\}, \qquad (1)$$

где $f_k\left(x, \boldsymbol{a}\right)$ – функция подобранного класса;

$\boldsymbol{a}$ – параметр, $\boldsymbol{a} = (a_0, ..., a_m)_k$, функции, наиболее близкой к регрессии $y = y(x)$, минимизирующей функционал.

$$I\left[f\left(x, \boldsymbol{a}\right)\right] = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - f\left(x_i, \boldsymbol{a}\right)\right]^2, \qquad (2)$$

где n – объем выборки $\left\{x_i, y_i\right\}$.

Размерность векторного параметра $\boldsymbol{a}$ различна для функций, принадлежащих разным классам. Компоненты векторного параметра $\boldsymbol{a}$ представляют собой $m+1$ неизвестных параметров (коэффициентов) функции подобранного класса.

Функционал (2) при известной функции подобранного класса $Y = f_k(x, \boldsymbol{a})$ представляет функцию $I\left(\boldsymbol{a}\right) = I\left[f_k\left(x, \boldsymbol{a}\right)\right]$ переменного параметра $\boldsymbol{a} = (a_0, ..., a_m)_k$ с точкой минимума $\boldsymbol{a}^* = (a_0^*, ..., a_m^*)_k$:

$$I\left(\boldsymbol{a}^*\right) \to \min. \qquad (3)$$

Для оценки качества функции $Y^* = f_k(x, \boldsymbol{a}^*)$, аппроксимирующей регрессию $y = y(x)$ и удовлетворяющей условие (3), обычно используется среднеквадратичное значение регрессионного остатка

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left[y_i - f_k(x_i, \boldsymbol{a}^*)\right]^2}. \qquad (4)$$

Таким образом, восстановление по выборке ограниченного объема парной регрессии, в общем случае нелинейной, может быть сведено к нахождению конечного множества оптимальных решений (2), (3) – $\boldsymbol{a}^* = (a_0^*, ..., a_m^*)_k$ для различных функций $Y = f_k(x, \boldsymbol{a})$ подобранных классов множества (1). Из конечного множества функций различных классов, для которых найдены точки минимума $\boldsymbol{a}^* = (a_0^*, ..., a_m^*)_k$, как решение рассматриваемой задачи, наиболее близкое к регрессии, может быть выбрана функция $Y^* = f^*(x, \boldsymbol{a}^*)$, удовлетворяющая условие

$$I(Y^*) = \inf_F \left\{ I\left[f_k(x, \boldsymbol{a}^*)\right]\right\}. \qquad (5)$$

Этому соответствует наименьшее значение регрессионного остатка (4).

Основной материал. Для подобранного класса функций $Y = f_k(x, \boldsymbol{a})$ алгоритм численного нахождения точки минимума (3) при числе переменных

$m+1$ ($\boldsymbol{a}^* = (a_0^*,...,a_m^*)_k$) может быть реализован с использованием методов нелинейного программирования. Как показал опыт решения подобных задач, эффективным является метод прямого поиска Хука-Дживса [2]. Метод состоит из последовательности шагов исследующего поиска вокруг базисной точки, а затем в случае успеха – поиска по образцу. Для рассматриваемой задачи этот метод сводится к следующему:

1. Выбрать начальную базисную точку $\boldsymbol{aBasis}_1$ и шаг длиной h для каждой переменной a_i, $i = \overline{0,m}$.

2. Вычислить $I(\boldsymbol{a})$ в базисной точке ($\boldsymbol{a} = \boldsymbol{aBasis}_1$) с целью получения сведений о локальном поведении функции $I(\boldsymbol{a})$. Эти сведения будут использоваться для нахождения подходящего направления поиска по образцу, с помощью которого можно надеяться достичь большего убывания значения функции. Функция $I(\boldsymbol{a})$ в базисной точке $\boldsymbol{aBasis}_1$ находится следующим образом:

2.1. Вычисляется значение функции $I(\boldsymbol{aBasis}_1)$ в базисной точке.

2.2. Каждая переменная по очереди изменяется прибавлением длины шага. Таким образом, вычисляется значение функции $I(\boldsymbol{aBasis}_1 + h\boldsymbol{e}_0)$, где $\boldsymbol{e}_0$ – единичный вектор в направлении оси a_0. Если это приводит к уменьшению значения функции, то $\boldsymbol{aBasis}_1$ заменяется на $\boldsymbol{aBasis}_1 + h\boldsymbol{e}_0$. В противном случае вычисляется значение функции $I(\boldsymbol{aBasis}_1 - h\boldsymbol{e}_0)$, и если ее значение уменьшилось, то $\boldsymbol{aBasis}_1$ заменяем на $\boldsymbol{aBasis}_1 - h\boldsymbol{e}_0$. Если ни один из проделанных шагов не приводит к уменьшению значения функции, то точка $\boldsymbol{aBasis}_1$ остается неизменной и рассматриваются изменения в направлении оси a_1, т. е. находится значение функции $I(\boldsymbol{aBasis}_1 + h\boldsymbol{e}_1)$ и т. д. Когда будут рассмотрены все $m+1$ переменные, мы будем иметь новую базисную точку $\boldsymbol{aBasis}_2$.

2.3. Если $\boldsymbol{aBasis}_2 = \boldsymbol{aBasis}_1$, т. е. уменьшение функции не было достигнуто, то исследование повторяется вокруг той же базисной точки $\boldsymbol{aBasis}_1$, но с уменьшенной длиной шага. На практике удовлетворительным является уменьшение шага в десять раз от начальной длины.

2.4. Если $\boldsymbol{aBasis}_2 \neq \boldsymbol{aBasis}_1$ то производится поиск по образцу.

3. При поиске по образцу используется информация, полученная в процессе исследования (п. 2), и минимизация функции завершается поиском в направлении, заданном образцом. Это производится следующим образом:

3.1. Разумно двигаться из базисной точки $\boldsymbol{aBasis}_2$ в направлении $\boldsymbol{aBasis}_2 - \boldsymbol{aBasis}_1$, поскольку поиск в этом направлении уже привел к уменьшению значения функции. Поэтому вычислим функцию $I(\boldsymbol{a})$ в точке образца $\boldsymbol{aPoint}_1 = \boldsymbol{aBasis}_1 + 2(\boldsymbol{aBasis}_2 - \boldsymbol{aBasis}_1)$. В общем случае на любом j-м шаге функция $I(\boldsymbol{a})$ вычисляется в точке $\boldsymbol{aPoint}_j = \boldsymbol{aBasis}_j + 2(\boldsymbol{aBasis}_{j+1} - \boldsymbol{aBasis}_j)$.

3.2. Затем исследование следует продолжать вокруг точки $\boldsymbol{aPoint}_1$ (в общем случае − вокруг точки $\boldsymbol{aPoint}_j$).

3.3. Если наименьшее значение на шаге 3.2 меньше значения в базисной точке $\boldsymbol{aBasis}_2$ (в общем случае $\boldsymbol{aBasis}_{j+1}$), то получают новую базисную точку $\boldsymbol{aBasis}_3$ ($\boldsymbol{aBasis}_{j+2}$), после чего следует повторить шаг 3.1. В противном случае не производить поиск по образцу из точки $\boldsymbol{aBasis}_2$ (в общем случае $\boldsymbol{aBasis}_{j+1}$), а

продолжить исследования в точке $\boldsymbol{aBasis}_2$ ($\boldsymbol{aBasis}_{j+1}$).

4. Завершить этот процесс, когда длина шага будет уменьшена до заданного малого значения.

На рис. 1 приведен исходный код модуля программы на языке программирования высокого уровня, реализующей алгоритм численного нахождения точки минимума $\boldsymbol{a}^* = (a_0^*,...,a_m^*)_k$ и регрессионного остатка (4) для подобранного класса функций.

```
Unit Regres;
interface
   uses RegTypes;{Внешний модуль описания подобранного класса
функций}
   var
    D:real;  {Делитель шага}
    procedure ParRegresCoef(n,NumCoef:integer; X,Y:XYType;
                      var a:aType; RegFunct:RegrFunctType;
                      h,hmin:real; var S:real)
{Подпрограмма вычисления коэффициентов парной регрессии
Параметры:
       n - число пар наблюдений;
 NumCoef - число коэффициентов парной регрессии;
     X,Y - массивы значений фактора X и функции отклика Y,
Y[i]=f(X[i]), i=1..n;
       a - массив значений коэффициентов функции парной регрессии
a[i],i=0..NumCoef - 1;
RegFunct - идентификатор функции регрессии;
       h - начальный шаг варьирования коэффициентов парной
регрессии;
    hmin - минимальное значение шага варьирования коэффициентов
парной регрессии;
       S - среднеквадратичное отклонение, определенное по сумме
квадратов регрессионных остатков.}
implementation
 procedure ParRegresCoef;
     var
      i :integer;
      aIter, aPoint, aBasis :aType;
      Sum,SumI,SumB :real;
      IsBasis :boolean;
     function SquSum(a:aType):real;
      var
       i    :integer;
       Sum  :real;
      begin
       Sum:=0;
```

Рис. 1. Исходный код модуля программы (начало)

```
    for i:=1 to n do
     Sum:=Sum+sqr(RegrFunct(a,X[i])-Y[i]);
    SquSum:=Sum;
   end;
  begin
{Минимизация с использованием метода Хука-Дживса}
   aIter:=a;
   aPoint:=a;
   aBasis:=a;
   SumI:=SquSum(a);
   IsBasis:=True;
   SumB:=SumI;
   repeat
    i:=0;
    while i<NumCoef do
     begin
  {Вариации коэффициентов регрессии}
      a[i]:=aIter[i]+h;
      Sum:=SquSum(a);
      if Sum<SumI then
       aIter[i]:=a[i]
      else
       begin
        a[i]:=aIter[i]-h;
        Sum:=SquSum(a);
        if Sum<SumI then
         aIter[i]:=a[i]
        else
         a[i]:=aIter[i];
       end;
      SumI:=SquSum(a);
      i:=i+1;

    end;
    if SumI<SumB then
     begin
  {Поиск по образцу}
      for i:=0 to NumCoef-1 do
       begin
        aPoint[i]:=2*aIter[i]-aBasis[i];
        aBasis[i]:=aIter[i];
        a[i]:=aPoint[i];
        aIter[i]:=a[i];
       end;
```

Рис. 1. Исходный код модуля программы (продолжение)

```
      SumB:=SumI;
      IsBasis:=False;
      SumI:=SquSum(a);
     end
    else
     begin
      if not IsBasis then
       begin
    {Замена базисной точки}
        aPoint:=aBasis;
        aIter:=aBasis;
        a:=aBasis;
        SumI:=SquSum(a);
        IsBasis:=True;
        SumB:=SumI;
       end
      else
       begin
        h:=h/D;
        if h<(hmin/D) then
         begin
           a:=aPoint;
           S:=sqrt(SquSum(a)/(n-1));
           Exit;
         end;
       end;
     end;
  until False;
 end;
End.
```

Рис. 1. Исходный код модуля программы (окончание)

Функция предварительно подобранного класса $Y = f_k(x, \boldsymbol{a})$, передаваемая в подпрограмму **ParRegresCoef** в виде параметра **RegFunct** (см. рис. 1), описывается в отдельном модуле. На рис. 2, как пример использования такой функции вида $y = a_0 + a_1 e^{-a_2 x}$, приводится исходный код модуля её описания.

Выводы. На основе продолжительного опыта использования предложены эффективный алгоритм численного восстановления функции парной регрессии произвольного общего вида по выборке ограниченного объема и программа на языке программирования, реализующая этот алгоритм.

```
    Unit RegTypes; {Модуь описания вида функции парной регрессии
подобранного класса}
    interface
     const
      nmax=500; NumCoefmax=10;
     type
      aType=array [0..NumCoefmax-1] of real;
      XYType=array [1..nmax] of real;
      RegrFunctType=function( a:aType; x:real ):real;
     var
       h,hmin  :real;{Начальный и минимальный шаги варьирования
           коэффициентов парной регрессии              }
       NumCoef :integer;{Число коэффициентов функции парной
регрессии                                                  }
       a :aType;{Массив значений коэффициентов функции парной
регрессии a[i],i=0..NumCoef - 1                            }
     function RegFunct(a:aType; x:real):real;{Функция регрессии}
implementation
         {$F+}
     function RegFunct;
       begin
        {Экспоненциальная функция регрессии}
        RegFunct:= a[0]+a[1]*exp(a[2]*(-x));
       end;
     begin
       NumCoef:=3;
     End.
```

Рис. 2. Пример исходного кода программного модуля описания функции подобранного класса

Список литературы

1. Айвазян, С.А. Прикладная статистика: Исследование зависимостей [Текст]: справ. издание / С.А. Айвазян, И.С. Енюков, Л.Д. Мешалкин; под ред. С.А. Айвазяна. – М.: Финансы и статистика, 1985. – 487 с.

2. Банди, Б. Методы оптимизации [Текст]: Вводный курс / Брайан Д. Банди; пер. с англ. О.В. Шихеевой под ред. В.А. Волынского. – М.: Радио и связь, 1988. – 128 с.

Ключевые слова: обработка данных, нелинейная парная регрессия, метод наименьших квадратов, численный алгоритм, метод прямого поиска, вычислительная программа.

Аннотации

Описано чисельний метод відновлення парної регресії довільного загального вигляду, у т. ч. нелінійної, з використанням МНК-оцінок. Наведено алгоритм і програму обчислювальної обробки даних, які реалізують цей метод.

Описан численный метод восстановления парной регрессии произвольного общего вида, в т. ч. нелинейной, с использованием МНК-оценок. Приведены алгоритм и программа вычислительной обработки данных, реализующие этот метод.

With the use of least-squares method estimations the numeral method of renewal of arbitrary general view pair regression, including nonlinear, is described. Which realize this method, the algorithm and the program of the calculable processing of data is resulted.