

ПРИМЕНЕНИЕ МНОГОУРОВНЕВОЙ АППРОКСИМАЦИИ ДЛЯ ПОСТРОЕНИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ НЕСТАЦИОНАРНЫХ ПРОЦЕССОВ

Предложен новый подход к проблеме построения математических моделей нестационарных процессов, наблюдения над которыми происходят в достаточно малые промежутки времени. При этом используются некоторые идеи многоуровневых вычислительных сетей.

1. ПОСТАНОВКА ЗАДАЧИ

Пусть имеется результат измерений некоторого процесса $z(t)$ в течение какого-то интервала времени $[0, T]$. Будем считать, что значения процесса $z(t)$ измеряются в дискретные моменты времени $t_1, t_2, \dots, t_N$, т.е. в нашем распоряжении имеется выборка

$$z = [z_1, z_2, \dots, z_N], \quad (1.1)$$

где $z_i = z(t_i)$, $i = 1, \dots, N$. Здесь z – вектор-строка порядка N . На основании этой выборки нужно построить математическую модель или оценку этого процесса, т.е. найти некоторую функцию $m(t)$, которая бы не только хорошо аппроксимировала выборку, но и позволяла предсказывать значения процесса $z(t)$ в последующие после T моменты времени (краткосрочный и среднесрочный прогноз).

В настоящее время существует теория аппроксимации, которая решает эту проблему. Наиболее популярным является построение линейной параметрической модели, когда модель процесса строится в виде

$$m(t) = \sum_{i=1}^n a_i f_i(t) = a^T f(t), \quad (1.2)$$

где $f_1(t), \dots, f_n(t)$ – известные базовые функции, $a_1, \dots, a_n$ – неизвестные параметры. В результате построения математической модели процесса $z(t)$ сводится к вычислению оценок этих параметров по выборке (1.1). В методе наименьших квадратов (МНК) эти оценки находятся из условия минимизации функции

$$J = \sum_{i=1}^N (z_i - m(t_i))^2 = \|z - m\|^2, \quad (1.3)$$

где $m = [m(t_1), m(t_2), \dots, m(t_N)]$ – вектор-строка порядка N . Здесь для простоты дальнейших записей введено обозначение $\|z - m\|^2 = (z - m)(z - m)^T$. Подставляя (1.2) в (1.3), вычисляя производные функции J по параметрам $a_1, \dots, a_n$ и приравнявая их к нулю, получаем систему линейных уравнений, из решения которых и получаем оценки для этих параметров.

Как показывают результаты исследований, построенная таким образом модель процесса оказывается хорошей, когда выборка большая, берется на достаточно большом интервале времени и процесс $z(t)$ стационарный. Для экономических процессов последнее, как правило, не выполняется. Для подобных процессов характерна нестационарность и сезонные колебания. Поэтому точность оценивания во многом определяется удачным выбором базовых функций $f_1(t), \dots, f_n(t)$. При этом увеличение числа этих функций теоретически должно улучшать оценки, но практически ничего не дает, так как начинают сказываться ошибки вычислений при решении полученных уравнений.

Имеется отдельная теория построения моделей, когда при получении выборки процесс $z(t)$ измеряется с какими-то случайными ошибками. Здесь эти про-

блемы обсуждаться не будут, поскольку основное внимание будет уделено задаче выбора базовых функций $f_1(t), \dots, f_n(t)$. Дело в том, что если $f_1(t), \dots, f_n(t)$ известны «плохо», то их незнание может привести к значительно большим ошибкам оценивания, чем ошибки измерений. Поэтому в данной работе будем считать, что ошибки измерений отсутствуют.

Ниже рассматривается попытка перенести некоторые идеи построения вычислительных (точнее нейронных) сетей к решению рассматриваемой проблемы. При этом будет использоваться соответствующая терминология. Непосредственное использование нейронных сетей для рассматриваемой проблемы, по-видимому, нецелесообразно, так как элементы этих сетей строятся на базе функций экспонентного типа, что не может быть использовано для временных колебательных нестационарных процессов. Поэтому в данной работе обсуждается проблема построения вычислительных сетей, в которых базовые функции могут быть произвольными (например, тригонометрическими или степенными). Входом такой сети является текущее время t , а выходом – оценка $m(t)$.

Выборку (1.1) будем называть *обучающей* для настройки вычислительной сети. В частности, выборку (1.1) можно разделить на две части. Одну часть можно использовать для настройки или обучения сети, а другую для проверки точности предсказания процесса.

2. МНОГОСЛОЙНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Используя идеи построения нейронных сетей, можно ввести формулы для построения оценок для процесса $z(t)$, которые условно назовем вычислительными сетями.

Однослойная вычислительная сеть

Оценка процесса $m(t)$ строится в виде суммы (1.2). Задача сводится к нахождению параметров a по значениям выборки (1.1).

Двухслойная вычислительная сеть

Оценка $m(t)$ строится в виде

$$m(t) = \sum_{i=1}^{n_1} a_i f_i(t) (1 + G_i(t)), \quad (2.1)$$

где

$$G_i(t) = \sum_{j=1}^{n_2} b_{ij} g_{ij}(t).$$

Здесь $f_i(t)$ и $g_{ij}(t)$ – заданные функции; a_i, b_{ij} – неизвестные параметры ($i = 1, \dots, n_1, j = 1, \dots, n_2$). Смысл множителя $G_i(t)$ в том, что он «растягивает» функцию $f_i(t)$ по амплитуде.

Трехслойная вычислительная сеть

Модель $m(t)$ строится в виде

$$m(t) = \sum_{i=1}^{n_1} a_i f_i(t) [1 + \sum_{j=1}^{n_2} b_{ij} g_{ij}(t) (1 + H_{ij}(t))], \quad (2.2)$$

где $H_{ij}(t) = \sum_{k=1}^{n_3} c_{ijk} h_{ijk}(t)$.

Здесь $f_i(t)$, $g_{ij}(t)$, $h_{ijk}(t)$ – заданные функции; a_i , b_{ij} , c_{ijk} – неизвестные параметры ($i = 1, \dots, n_1$, $j = 1, \dots, n_2$, $k = 1, \dots, n_3$). Смысл множителя $H_{ij}(t)$ в том, что он «растягивает» функцию $g_{ij}(t)$ по амплитуде.

Можно и дальше увеличивать число слоев, однако это вряд ли что-либо дает из-за ошибок вычислений. В вычислительном отношении удобны *простые* сети, когда функции $g_{ij}(t)$ не зависят от i , а функции $h_{ijk}(t)$ – от i и j .

Во всех этих вариантах задача сводится к нахождению параметров a_i , b_{ij} , c_{ijk} по значениям выборки (1.1). Эти параметры выбираются так, чтобы минимизировать функционал (1.3). По аналогии с нейронными сетями вычисление этих параметров будем называть *настройкой сети*.

3. НАСТРОЙКА ОДНОСЛОЙНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

Решение этой задачи хорошо известно. Введем матрицу $F = [f(t_1), f(t_2), \dots, f(t_N)]$ порядка $n_1 \times N$. Тогда функцию (1.3) можно записать в виде

$$J = (z - a^T F)(z - a^T F)^T.$$

В результате минимизации этого выражения по вектору a получаем оценку

$$a = (FF^T)^{-1} Fz^T. \quad (3.1)$$

Для существования этой оценки необходима невырожденность матрицы $R = FF^T$, для чего, в частности, требуется, чтобы функции $f_1(t), \dots, f_n(t)$ были линейно независимыми и $N > n$. Кроме того, при практическом решении задач могут быть затруднения при обращении матрицы R для больших n .

4. НАСТРОЙКА ПРОСТОЙ ДВУХСЛОЙНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

Поскольку многослойные модели являются нелинейными относительно искомых параметров, то нахождение их оценок является довольно сложной задачей. В теории нейронных сетей для решения аналогичных проблем используются градиентные методы. Вычислить градиенты функции (1.3) по параметрам a_i , b_{ij} , c_{ijk} не сложно, однако сходимость градиентных методов в силу специфики задачи оказывается очень плохой. Поэтому можно рассматривать другие разные алгоритмы настройки сетей.

Ниже приводится один такой алгоритм, который изложен на примере простой двухслойной вычислительной сети, т.е. для модели

$$m(t) = \sum_{i=1}^{n_1} a_i f_i(t) (1 + \sum_{j=1}^{n_2} b_{ij} g_j(t)). \quad (4.1)$$

Сначала введем некоторые обозначения. Обозначим через B матрицу порядка $n_1 \times n_2$, составленную из коэффициентов b_{ij} , и через G – $n_2 \times N$ -матрицу, состав-

ленную из $[g(t_1), \dots, g(t_N)]$. Тогда строку $m = [m(t_1), m(t_2), \dots, m(t_N)]$ можно записать в виде

$$m = a^T L(B), \quad (4.2)$$

где $L(B) = F + BG$. Здесь F означает поэлементное перемножение матриц.

Введем еще обозначения. Через b_j обозначим j -й столбец матрицы B и через $g_j^T = [g_j(t_1), \dots, g_j(t_N)]$ – j -ю строку матрицы G . Тогда (4.2) можно переписать в виде

$$m = a^T F + \sum_{i=1}^{n_2} b_i^T H(a, g_i), \quad (4.3)$$

где $H(a, g) = (a \oplus g^T) \times F$.

Здесь знак $\oplus$ означает кронекеровское произведение матриц.

Вектор a и матрицу B будем искать с помощью последовательных приближений. В качестве нулевого приближения для матрицы B возьмем нулевую матрицу $B^{(0)} = 0$. Тогда нулевое приближение для вектора a будет, согласно (3.2), иметь вид

$$a^{(0)} = (L^{(0)} L^{(0)T})^{-1} L^{(0)} z^T, \quad (4.4)$$

где $L^{(k)} = L(B^{(k)}) = F + B^{(k)} G$, причем $L^{(0)} = F$.

1-й шаг. Первый столбец b_1 матрицы B находится из условия минимума функционала

$$J = \langle a^{(0)T} F + b_1^T H(a^{(0)}, g_1) - z \rangle^2.$$

Здесь, как и раньше, z – N -мерная вектор-строка, составленная из значений $z_1, \dots, z_N$. В результате минимизации этого выражения по b_1 получаем

$$b_1 = [H(a^{(0)}, g_1) H^T(a^{(0)}, g_1)]^{-1} H(a^{(0)}, g_1) (z - a^{(0)T} F)^T.$$

Подставляя этот столбец в $B^{(0)}$, получаем матрицу $B^{(1)}$. Первое приближение для вектора a находится из условия минимума функции $J = \langle a^T L^{(1)} - z \rangle^2$.

В результате по аналогии с (3.2) получаем

$$a^{(1)} = (L^{(1)} L^{(1)T})^{-1} L^{(1)} z^T.$$

2-й шаг. Вторым столбец b_2 матрицы B находится из условия минимума функции

$$J = \langle a^{(1)T} L^{(1)} + b_2^T H(a^{(1)}, g_2) - z \rangle^2.$$

В результате минимизации этого выражения по b_2 получаем

$$b_2 = [H(a^{(1)}, g_2) H^T(a^{(1)}, g_2)]^{-1} H(a^{(1)}, g_2) (z - a^{(1)T} L^{(1)})^T.$$

Подставляя этот столбец в $B^{(1)}$, получаем матрицу $B^{(2)}$. Второе приближение для вектора a находится из условия минимума функции $J = \langle a^T L^{(2)} - z \rangle^2$.

В результате по аналогии с (3.2) получаем

$$a^{(2)} = (L^{(2)} L^{(2)T})^{-1} L^{(2)} z^T.$$

Продолжая эту процедуру, можно вычислить все столбцы матрицы B и соответствующий вектор a .

5. ТЕСТОВЫЙ ПРИМЕР

В качестве процесса $z(t)$ возьмем функцию

$$z(t) = r + q e^{at} |\sin \omega t|. \quad (5.1)$$

Можно дать следующую «экономическую» интерпретацию. Пусть $z(t)$ – количество заявок на туристические поездки. Этот процесс имеет сезонный характер, что отражается синусом, и со временем растет,

что отражается экспонентой. Модуль синуса взят для того, чтобы усложнить задачу математически, так как функция $z(t)$ становится ломаной.

Пусть выборка (1.1) состоит из данных за четыре полупериода времени длиной T ($\omega=\pi/T$). Причем первые три полупериода используются для обучения сети, а четвертый – для проверки обучения. Пусть фиксируется K значений функции $z(t)$ в течение полупериода. Если считать, что $t_i=i$ ($i=0,1,2, \dots$), то $T=K$ и весь объем обучающей выборки равен $N=3K+1$.

На рис. 1 и 2 приведены результаты расчетов. Расчеты проводились при $r=3$, $q=1$, $\alpha=0,02$, $K=24$. На рис. 1 приведены результаты построения модели с помощью однослойной вычислительной сети при базовых функциях

1, $\sin t$, $\cos t$, $\sin 2t$, $\cos 2t$, $\sin 3t$, $\cos 3t$... (5.2)

– кривая 2, и при базовых функциях

1, t/T , $(t/T)^2$, $(t/T)^3$,... (5.3)

– кривая 3. Кривая 1 соответствует процессу (5.1). Видно, что кривая 2 хорошо аппроксимирует выборку только на втором полупериоде. Это объясняется нестационарностью процесса (5.1) и стационарностью функций (5.2). Кривая 3 дает плохую аппроксимацию, так как функции (5.3) не являются колебательными. Расчеты показывают, что дальнейшее увеличение базовых функций (5.2) или (5.3) практически не улучшают аппроксимацию.

На рис.2 приведены результаты построения модели с помощью двухслойной вычислительной сети при базовых функциях первого уровня (5.2) и базовых функциях второго уровня (5.3). Настройка сети выполнялась по описанному выше алгоритму. На этом рисунке кривая 1 соответствует процессу (5.1), кривая 2 – случаю однослойной сети с базовыми функциями (5.2) (кривая 2 на рис. 1), кривая 3 – случаю двухслойной сети. Хорошо видно, что последняя кривая достаточно хорошо аппроксимирует процесс (5.1).

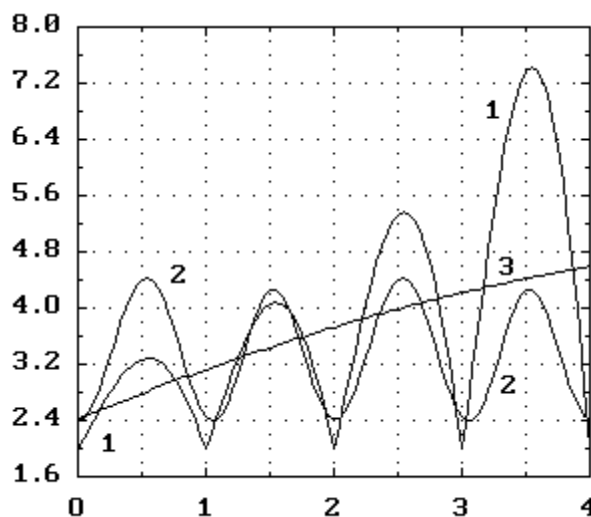


Рис. 1

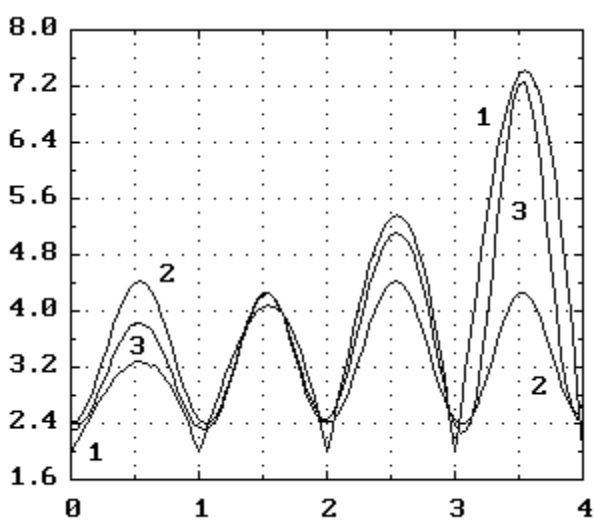


Рис. 2

Статья представлена кафедрой прикладной математики Томского государственного университета, поступила в научную редакцию «Кибернетика и информатика» 1 сентября 2003 г.