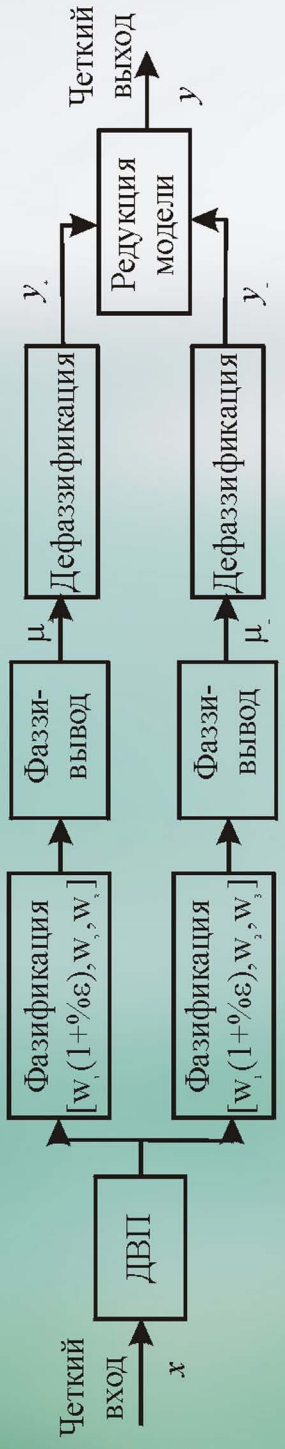
А.И. МИХАЛЕВ
Е.А. ВИНОКУРОВА
С.Л. СОТНИК

КОМПЬЮТЕРНЫЕ МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ДАННЫХ



$$F(x_1,\ldots,x_{m_0}) = \sum_{i=1}^{m_1} a_i \varphi\left(\sum_{j=1}^{m_0} w_{ij} x_j + b_i\right)$$

Системные Технологии

А.И. МИХАЛЕВ

Е.А. ВИНОКУРОВА

С.Л. СОТНИК

КОМПЬЮТЕРНЫЕ МЕТОДЫ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ДАННЫХ

Учебное пособие

2014

УДК 621.391:681.301

Михалев А.И., Винокурова Е.А., Сотник С.Л. Компьютерные методы интеллектуальной обработки данных: учебное пособие. – Днепропетровск: НМетАУ, ИК "Системные технологии", 2014. – 209 стр.

Ил.: . Библ.: наим.

Данное учебное пособие предназначено для освещения основных вопросов по компьютерным методам интеллектуальной обработки данных.

Пособие состоит из пяти разделов, в первом из которых рассмотрены основы компьютерной обработки данных. Второй раздел посвящен методам выявления ассоциаций и закономерностей с использованием мягких вычислений. Приведены примеры применения эвристических (ГА) и эволюционных, в частности, муравьиных алгоритмов.

В свою очередь, в третьем разделе представлен обзор ИАД-методов нечеткой кластеризации, а в четвертом, на примере применения метода экспоненциального среднего и стратегий формирования ансамблей моделей-предикторов таких как беггинг, бустинг и стэкинг, показаны эффективные алгоритмы адаптивного прогнозирования данных. Пятый раздел посвящен перспективному направлению в обработке данных – динамическому интеллектуальному анализу (Dynamical Data Mining).

Учебное пособие предназначено для студентов и слушателей, специализирующихся в направлении 050101– "Компьютерные науки".

Рецензенты:

Шумейко Александр Алексеевич, доктор технических наук, профессор Днепродзержинского государственного технического университета;

Коваленко Игорь Иванович, доктор технических наук, профессор Национального университета кораблестроения имени адмирала Макарова.

Рекомендовано к печати
Ученым советом Национальной металлургической академии Украины
Протокол от 28.10. 2013 г., № 9

ISBN 978-966-2596-14-4

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Интеллектуальный анализ данных (ИАД, Data Mining, KDD – knowledge discovery in databases) представляет собой достаточно новое направление в области информационных систем, ориентированное на решение задач поддержки принятия решений на основе количественных и качественных исследований больших и сверхбольших массивов разнородных данных [22, 57, 78, 95, 100, 136].

Принципиальное отличие интеллектуального анализа данных от известных методологий, используемых в существующих системах поддержки принятия решений, состоит в переходе от технологии оперативного экспресс-анализа текущих ситуаций, характерной для традиционных систем обработки данных, к фундаментальным методам исследований, существенно опирающимся на мощный аппарат современной дискретной и нейро-нечеткой математики.

ИАД (Data Mining) является мультидисциплинарной областью, возникшей и развивающейся на базе достижений прикладной статистики, распознавания образов, методов искусственного интеллекта, теории баз данных и представляет собой процесс обнаружения значимых корреляций, зависимостей и тенденций в результате анализа содержимого информационных хранилищ с применением методов распознавания и выявления ассоциаций (аналогичных последовательностей, кластеров) данных [95]. В результате удается выделить шаблоны, отражающие фрагменты многоаспектных взаимоотношений в данных, и вывести из них правила для принятия решений и прогнозирования их последствий.

Совершенствование сложных технических систем, систем принятия решений, технологий получения, хранения и передачи данных привели к появлению нового поколения методов интеллектуального анализа данных, математическую основу которых составляют противоположные традиционным компьютерным вычислениям (hard computing) мягкие вычисления (soft computing) или сложная компьютерная методология, компонентами которой являются: нечеткая логика – приближенные вычисления, грануляция информации, вычисление на словах; нейрокомпьютинг – обучение, адаптация, классификация, системное моделирование и идентификация; генетические вычисления – синтез, настройка и оптимизация с помощью систематизированного случайного поиска и эволюции; вероятностные вычисления – управление неопределенностью. Позднее в этот конгломерат были включены т.н. «мягкие» рассуждения на базе свидетельств, сети доверия, хаотические системы и разделы теории машинного обучения.

Современная теория интеллектуального анализа данных охватывает решение таких задач как классификация, сегментация, ассоциативный анализ, выявление аномалий, анализ последовательности событий, анализ временных рядов, категоризация текста, структурный анализ и многие

другие. Интеллектуальные средства интерпретации и представления данных позволяют выявлять следующие виды закономерностей: ассоциация, последовательность, классификация, кластеризация, прогнозирование.

Алгоритмический аппарат интеллектуального анализа данных включает в себя множество методов и алгоритмов: деревья решений (decision trees), предназначенные для определения факторов, влияющих в наибольшей степени на определенный атрибут и использующиеся для решения задач классификации, регрессии, выявления взаимосвязей; алгоритм Байеса, использующийся для решения задач определения взаимозависимостей атрибутов, классификации, прогноза; методы построения ассоциативных правил, предназначенные для выявления взаимосвязей между различными характеристиками и их значениями.

На рис. В.1 приведены возможности и основной инструментарий интеллектуального анализа данных.

Вместе с тем изучение ИАД невозможно без знаний принципов классической и неоклассической обработки данных: понятия линейной свертки, преобразований Фурье и Лапласа, классических численных методов анализа временных рядов, кратномасштабного вейвлет-анализа и т.д. В этой связи *первый раздел* данного вводного курса в компьютерные методы интеллектуальной обработки данных посвящен изучению классических подходов и современных методов анализа сигналов.

Второй раздел посвящен методам выявления ассоциаций и закономерностей. Рассмотрены мягкие вычисления, примеры применения эвристических (ГА) и эволюционных, в частности, муравьиных алгоритмов. Листинги программ данного раздела размещены на сайте https://cmidpbook.codeplex.com/, загрузка программ архивом - https://cmidpbook.codeplex.com/SourceControl/latest# .

В *третьем разделе* представлен обзор ИАД-методов кластеризации, а в *четвертом*, на примере применения метода экспоненциального среднего, показана возможность прогнозирования данных временного ряда. Применение данного метода, в свою очередь, обобщено в *пятом разделе*, который посвящен динамическому интеллектуальному анализу данных (Dynamical Data Mining).

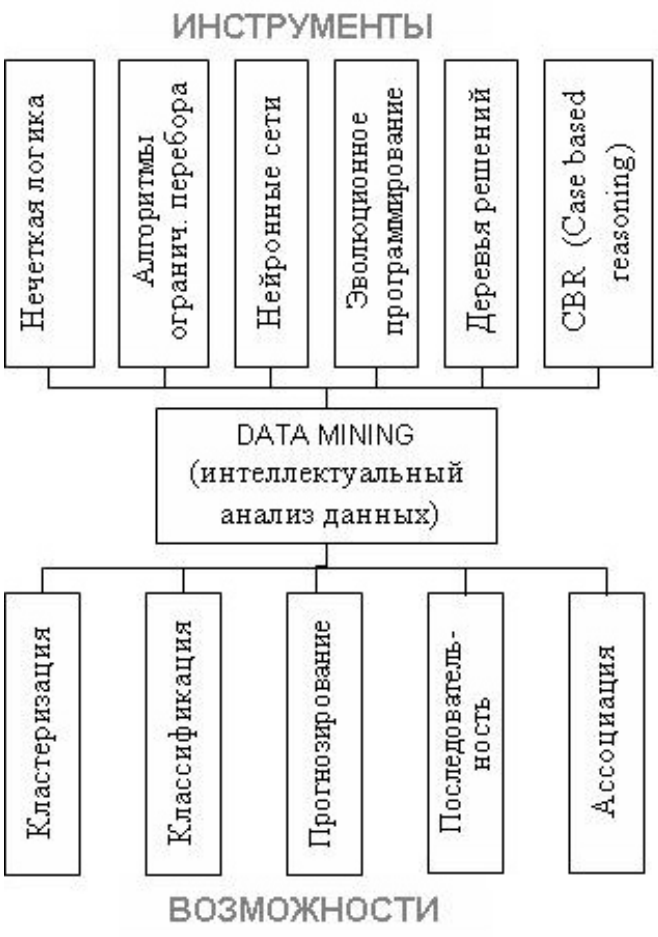


Рис. В.1. - Интеллектуальный анализ данных: возможности и основной инструментарий

Данное учебное пособие написано на основе лекций, прочитанных в течение ряда лет студентам и слушателям, обучающихся специальностям в области «Компьютерных наук».

РАЗДЕЛ I

ОСНОВЫ КОМПЬЮТЕРНОЙ ОБРАБОТКИ ДАННЫХ*

1.1. Сигналы и их общая классификация

Сигнал в широком смысле может рассматриваться как зависимость одной физической величины от другой: от времени, координат, температуры, давления и т.д. Например, в электромеханических системах машиностроительного или металлургического производства в качестве сигналов могут выступать зависимости числа оборотов двигателей приводов, перемещения штоков исполнительных устройств во времени и т.п. [1, 2].

Под термином *сигнал* в технических приложениях, как правило, подразумевают напряжение, несколько реже — ток или другие наблюдаемые в процессе протекания тех или иных физических процессов величины.

Различают *детерминированные* и *случайные* сигналы. Детерминированный сигнал считается полностью известным — его значение в любой момент времени можно определить точно. Случайный же сигнал в любой момент времени представляет собой случайную величину, которая принимает конкретные значения с некоторой *вероятностью (стохастические сигналы)* или удовлетворяет определенным соотношениям информационной энтропии и порядка *(при n ≥3) – хаотические сигналы.* Последние при условии выполнения свойства масштабной инвариантности являются также *фрактальными сигналами.*

Важнейшей характеристикой сигнала является его энергия. При этом выделяют так называемые сигналы с *ограниченной энергией (*сигналы с *интегрируемым квадратом).* С точки зрения математики такие сигналы можно рассматривать как функции $s(t) \in L_2$ [14]. Для таких сигналов $s(t)$ выполняется соотношение

$$\int_{-\infty}^{\infty} s^2(t)dt < \infty.$$

В свою очередь, следует отметить, что многие важные соотношения теории сигналов получены именно в предположении о конечности энергии анализируемых сигналов.

Другим важнейшим признаком классификации сигналов является их *периодичность.* Для периодического сигнала с периодом T выполняется соотношение $s(t + nT) = s(t)$ при любом t, где n — произвольное целое число. Если величина T является периодом сигнала $s(t)$, то периодами для него будут и кратные ей значения: $2T$, $3T$ и т. д. Как правило, говоря о периоде сигнала**,** имеют в

* Данный раздел существенно опирается на материал монографии [1].

виду минимальный из возможных периодов. Величина, обратная периоду, называется *частотой повторения* сигнала: $f = 1/T$, измеряемой *в герцах*. В теории сигналов также часто используется понятие *круговой частоты* $\omega = 2\pi f$, измеряемой *в радианах в секунду*.

Очевидно, что любой периодический сигнал (за исключением сигнала, тождественно равного нулю) имеет бесконечную энергию.

Если сигнал *существует* на конечном временном интервале, его относят к классу *сигналов* с *конечной длительностью* (их еще называют *финитными* сигналами).

Очевидно, что сигнал конечной длительности будет иметь и конечную энергию , конечно, если только он не содержит разрывов второго рода (с уходящими в бесконечность ветвями значений).

Важнейшую роль в технике обработки сигналов играют *гармонические* колебания, математически описываемые в виде: $s(t) = A\cos(\omega t + \varphi)$.

Гармонический сигнал характеризуется тремя числовыми параметрами: амплитудой A, частотой ω и начальной фазой φ. Такие сигналы являются одними из широко распространенных *тестовых* сигналов, применяющихся для анализа характеристик электрических цепей. Кроме него к тестовым относят еще два очень важных сигнала в виде дельта-функции и функции единичного скачка.

Дельта-функция $\delta(t)$ или *функция Дирака,* представляет собой бесконечно узкий импульс с бесконечной амплитудой, расположенный при нулевом значении аргумента функции. При этом площадь импульса равна единице:

$$\delta(t) = \begin{cases} 0, t \neq 0, \\ \infty, t = 0, \end{cases} \qquad \{1(t) \text{ - функция единичного скачка}\}.$$

Как видно, сигнал в виде дельта-функции невозможно реализовать физически точно, однако эта *синтетическая функция* очень важна для теоретического анализа и синтеза сигналов и систем.

На графиках дельта-функция обычно изображается жирной стрелкой, высота которой пропорциональна множителю, стоящему перед дельта-функцией (рис. 1.1).

Одно из важных свойств дельта-функции — это так называемое *фильтрующее свойство (sifting property of the delta function).* Оно состоит в том, что если дельта-функция присутствует под интегралом в качестве множителя, то результат интегрирования будет равен значению остального подынтегрального выражения в той точке, где сосредоточен дельта-импульс:

$$\int_{-\infty}^{\infty} f(t)\delta(t - t_0)dt = f(t_0). \qquad (1.1)$$

Замечание. *Фильтрующее свойство* $\delta(t)$ *следует из теоремы о среднем* [14].

$$\int_a^b f_1(x)\cdot f_2(x)dx = f_1(\xi)\int_a^b f_2(x)dx \quad .$$

Если $f_2(x)=\delta(x);\quad \int_{-\infty}^{\infty}\delta(x)dx=1$ *, тогда, действительно,*

$$\int_{-\infty}^{\infty} f(x)\cdot\delta(t-t_0)dt = f(t_0).$$

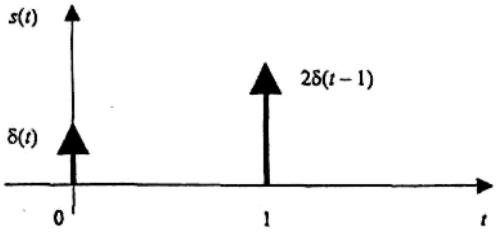


Рис. 1.1- График сигнала s(t) = δ(t) + 2δ(t - 1)

Пределы интегрирования в выражении (1.1) не обязательно должны быть бесконечными, главное, чтобы в интервал интегрирования попадало значение t_0, в противном случае интеграл будет равен нулю.

В свою очередь, из того факта, что интеграл от дельта-функции дает безразмерную единицу, следует, что размерность самой дельта-функции обратна размерности ее аргумента. Например, дельта-функция времени имеет размерность сек$^{-1}$, то есть размерность частоты.

Дельта-функция обладает также еще одним важным свойством, которое носит название *модельного (sampling property of the delta function)* и заключается в следующем:

$$f(t)\delta(t-t_0)=f(t_0)\delta(t) \quad \text{или, когда } t_0=0 \quad f(t)\delta(t)=f(0)\delta(t).$$

Пример. $sin\,t\,\delta\left(t-\frac{\pi}{6}\right)=sint\Big|_{t=\pi/6}\delta\left(t-\frac{\pi}{6}\right)=sin\frac{\pi}{6}\delta\left(t-\frac{\pi}{6}\right)=0.5\delta\left(t-\frac{\pi}{6}\right).$

Функция *единичного скачка* $\sigma(t)=1(t)=1_0(t)$ *(unit step function)*, она же *функция Хевисайда,* она же *функция включения,* равна нулю для отрицательных значений аргумента и единице — для положительных. При нулевом значении аргумента функцию считают либо неопределенной, либо равной 1/2:

$$\sigma(t)=\begin{cases} 0, t<0, \\ 1/2, t=0, \\ 1, t>0. \end{cases} \quad \text{в общем случае } t\in(-\infty,\infty) \qquad (1.2)$$

В MATLAB данную функцию можно смоделировать с помощью оператора сравнения, возвращающего значение 0 или 1:

$$>>\ y\ =\ (x\ >=\ 0)\,.$$

Отличие такой реализации функции включения от формулы (1.2) состоит только в том, что при нулевом значении аргумента результат равен единице; впрочем, в большинстве случаев это отличие допустимо.

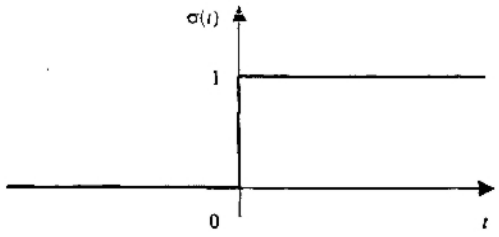


Рис. 1.2 - Функция единичного скачка

Функцию единичного скачка удобно использовать при создании математических выражений для сигналов конечной длительности. Простейшим примером может явится формирование прямоугольного импульса с амплитудой *A* и длительностью T:

$$s(t) = A(\sigma(t) - \sigma(t-T))$$

В свою очередь, любую кусочно-заданную зависимость можно записать в виде единого математического выражения с помощью функции единичного скачка.

Для примера приведем график функции s(t)=(σ(t) - σ(t-2)) + 3(σ(t-2) - σ(t-4)) + 5(σ(t-4) - σ(t-8))= σ (t) + 2σ(t-2) + 2(t-4) - 5σ(t-8).

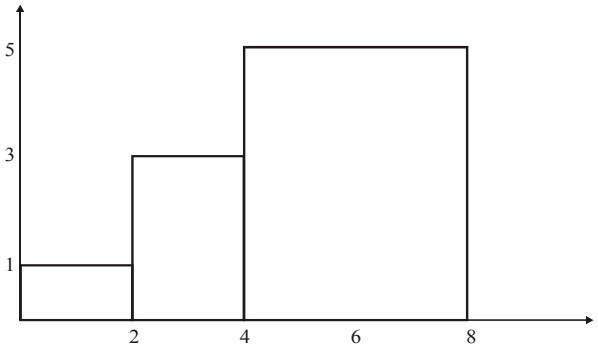


Рис.1.3

В MATLAB имеются функции для реализации дельта-сигналов и единичного скачка. Они носят имена ученых Поля Дирака и Оливера Хевисайда, впервые использовавших эти функции в своих работах: **Dirac(t)** и **Heaviside(t)**. Приведем пример их использования.

```
syms k a t;                    % Define symbolic variables
u=k*sym('Heaviside(t-a)') % Create unit step function at t = a
u = k*Heaviside(t-a)
d=diff(u)                      % Compute the derivative of the unit step function
d = k*Dirac(t-a)
int(d) % Integrate the delta function
    ans =
Heaviside(t-a)*k
```

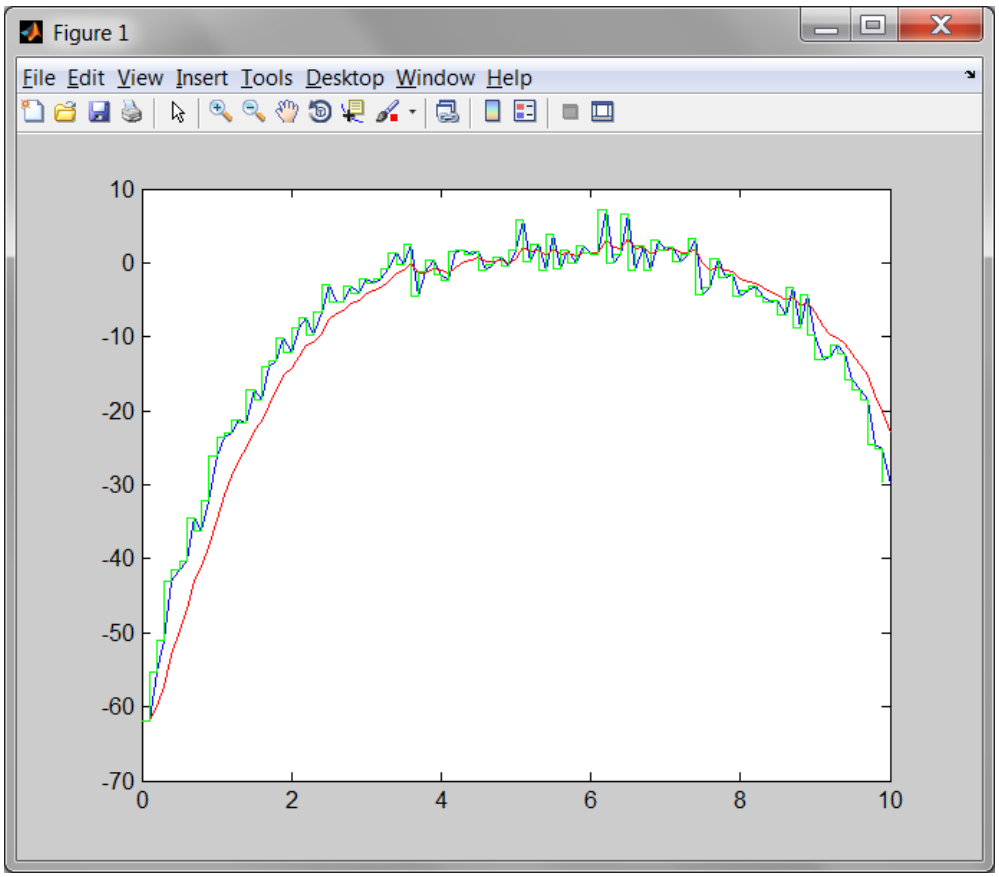


Рис. 1.4 - Представление сигнала в виде суммы сигма-функций (функций Хевисайда)

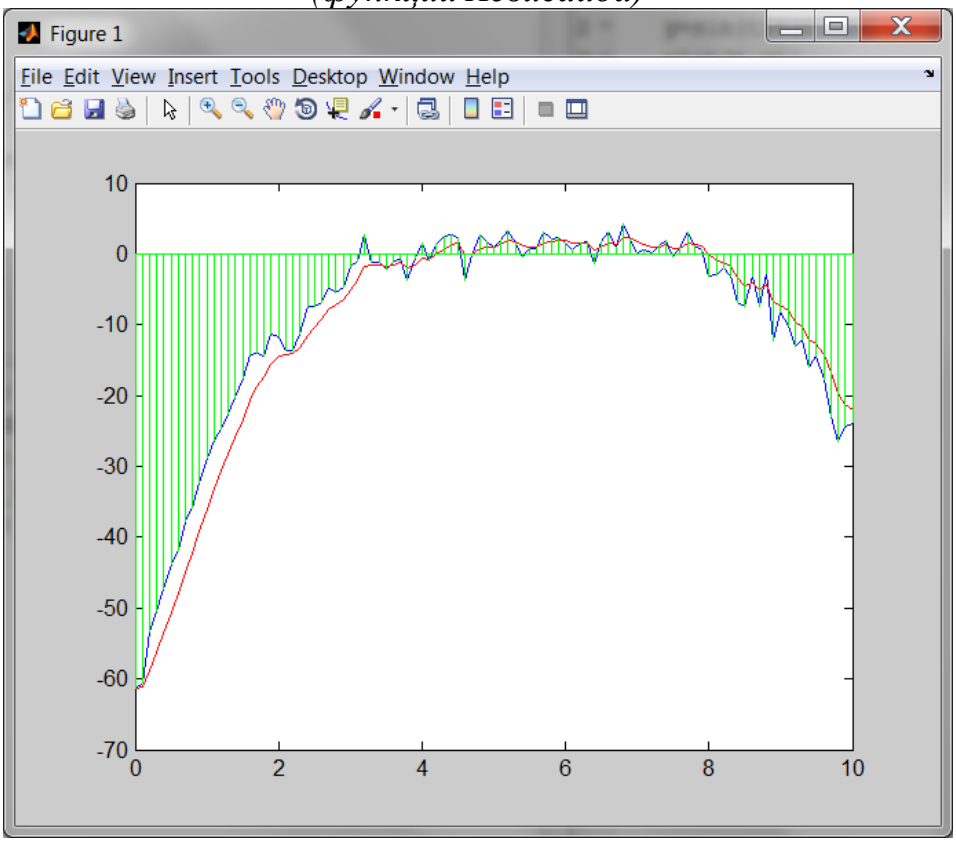


Рис. 1.5 - Представление сигнала в виде суммы дельта-функций (функций Дирака)

1.2. Основные характеристики сигналов

Одной из составляющих анализа сигналов является измерение их основных количественных характеристик. К таким характеристикам, прежде всего, относят *энергию* и *мощность* сигнала. Их определения, принятые в теории сигналов, отличаются от обычных, а потому требуют некоторых комментариев [2].

Начнем с физических понятий мощности и энергии. Если к резистору с сопротивлением R приложено постоянное напряжение U, то выделяющаяся в резисторе мощность будет равна

$$P = \frac{U^2}{R} = UI .$$

За время T в этом резисторе выделится тепловая энергия, равная

$$E = \frac{U^2 T}{R}.$$

Пусть теперь к тому же резистору приложено не постоянное напряжение, а сигнал $s(t)$. Рассеивающаяся в резисторе мощность при этом тоже будет зависеть от времени, то есть в данном случае речь идет о *мгновенной* мощности (instantaneous power):

$$p(t) = \frac{s^2(t)}{R}.$$

Чтобы вычислить выделяющуюся за время T энергию, мгновенную мощность необходимо проинтегрировать:

$$E = \int_0^T p(t)dt = \frac{1}{R}\int_0^T s^2(t)dt.$$

При этом также можно ввести понятие *средней* мощности (*average power*) за заданный промежуток времени, разделив энергию на длительность временного интервала:

$$P_{ср} = \frac{E}{T} = \frac{1}{RT}\int_0^T s^2(t)dt.$$

Во все приведенные формулы входит сопротивление нагрузки R. Однако, если энергия и мощность рассматриваются не как физические величины, а как средство *сравнения* различных сигналов, этот параметр можно из формул исключить (принять $R = 1$). Тогда получим определения энергии, мгновенной мощности и средней мощности, принятые в теории сигналов:

$$E = \int_0^T s^2(t)dt, \text{ - энергия, выделившаяся за время } T,$$

$$p(t) = s^2(t) \text{ - мгновенная мощность,} \qquad (1.3)$$

$$P_{ср} = \frac{1}{T}\int_0^T s^2(t)dt.$$ - средняя мощность.

Энергия сигнала может быть конечной или бесконечной. Например, любой сигнал конечной длительности будет иметь конечную энергию (если только он не содержит дельта-функций или ветвей, уходящих в бесконечность). А вот *любой периодический сигнал, напротив, имеет бесконечную энергию*.

Если энергия сигнала бесконечна, можно определить его среднюю мощность на всей временной оси. Для этого нужно воспользоваться формулой (1.3) и выполнить предельный переход, устремив интервал усреднения в бесконечность:

$$P_{ср} = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} s^2(t)dt. \qquad (1.4)$$

Если этот интеграл сходится, то средняя мощность конечна!

Для периодического сигнала $s(t) = \sin\omega t$ имеем

$$P_{ср} = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}\sin^2\omega t dt = \lim_{T\to\infty} I = \frac{1}{2}.$$

Здесь $$I = \frac{1}{2T}\int_{-T/2}^{T/2}(1-\cos 2\omega t)dt =$$

$$= \frac{1}{2T}t\Big|_{-T/2}^{T/2} - \frac{\sin(2\omega t)}{2\cdot 2\omega T}\Big|_{-T/2}^{T/2} = \frac{1}{2} - \frac{1}{2\omega T}\sin(\omega T)\Big|$$

$$\sigma_s = \sqrt{P_{ср}} = \sqrt{\frac{1}{2}} = \frac{\sqrt{2}}{2} \simeq 0{,}707.$$

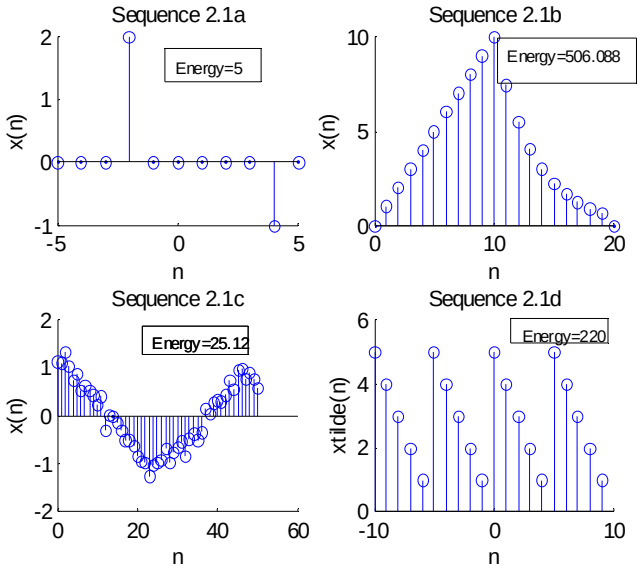
Квадратный корень из средней мощности дает с*реднеквадратическое (действующее)* значение сигнала (английский термин — *root mean square*, RMS):

$$\sigma_s = \sqrt{P_{ср}} = \sqrt{\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} s^2(t)dt}. \qquad (1.5)$$

Для периодического сигнала с периодом T предельный переход в формулах (1.4) и (1.5) выполнять не обязательно — достаточно выполнить усреднение по периоду.

Примеры 1.1 и 1.2 иллюстрируют вычисление энергии и мощности дискретных сигналов в MATLAB.

Пример 1.1



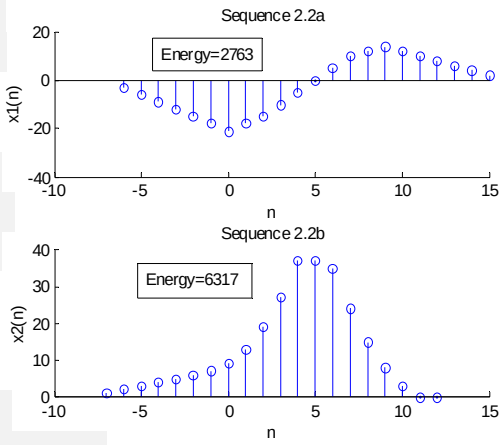

```
n=[-5:5];%primer 1.1
x=2*impseq(-2,-5,5)-impseq(4,-
5,5);
subplot(2,2,1);stem(n,x); ti-
tle('Sequence 2.1a')
xlabel('n'); ylabel('x(n)');
Ex1=sum(x.*conj(x))
n=[0:20];
x1=n.*(stepseq(0,0,20)-
stepseq(10,0,20));
x2=10*exp(-0.3*(n-
10)).*(stepseq(10,0,20)-
stepseq(20,0,20));
x=x1+x2;
subplot(2,2,2);stem(n,x); ti-
tle('Sequence 2.1b')
xlabel('n'); ylabel('x(n)');
Ex2=sum(x.*conj(x))
n=[0:50];
x=cos(0.04*pi*n)+0.2*randn(size(n));
subplot(2,2,3);stem(n,x); title('Sequence 2.1c')
xlabel('n'); ylabel('x(n)');
Ex3=sum(x.*conj(x))
n=[-10:9]; x=[5,4,3,2,1];
xtilde=x'*ones(1,4);
xtilde=(xtilde(:))';
subplot(2,2,4);stem(n,xtilde); title('Sequence 2.1d')
xlabel('n'); ylabel('xtilde(n)');
Ex4=sum(xtilde.*conj(xtilde))

function [x,n]=impseq(n0,n1,n2)
n=[n1:n2];x=[(n-n0)==0];
function [x,n] = stepseq(n0,n1,n2)
n=[n1:n2];
x=[(n-n0)>=0];
```

Пример 1.2.

```
n=[-2:10];%primer 1.2
x=[1:7,6:-1:1];
[x11,n11]=sigshift(x,n,5);
[x12,n12]=sigshift(x,n,-4);
[x1,n1]=sigadd(2*x11,n11,-3*x12,n12);
subplot(2,1,1);stem(n1,x1); title('Sequence 2.2a')
xlabel('n'); ylabel('x1(n)');
Ex1=sum(x1.*conj(x1))
[x21,n21]=sigfold(x,n);
[x21,n21]=sigshift(x21,n21,3);
```

```
[x22,n22]=sigshift(x,n,2);
[x22,n22]=sigmult(x,n,x22,n22);
[x2,n2]=sigadd(x21,n21,x22,n22);
subplot(2,1,2);stem(n2,x2); title('Sequence 2.2b')
xlabel('n'); ylabel('x2(n)');
Ex2=sum(x2.*conj(x2))
function [y,n]=sigshift(x,m,n0)
n=m+n0;
y=x;
function [y,n]=sigadd(x1,n1,x2,n2)
n=min(min(n1),min(n2)):max(max(n1),max(n2));
y1=zeros(1,length(n));
y2=y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y=y1+y2;
function [y,n]=sigfold(x,n)
y=fliplr(x);
n=-fliplr(n);
function [y,n]=sigfold(x,n)
y=fliplr(x);
n=-fliplr(n);
```



1.3 Ряд и преобразование Фурье

Периодические сигналы могут быть разложены в ряд Фурье. При этом они представляются в виде суммы гармонических функций либо комплексных экспонент с частотами, образующими арифметическую прогрессию. Для того чтобы такое разложение было возможным, фрагмент сигнала длительностью в один период должен удовлетворять *условиям Дирихле:*

- не должно быть разрывов второго рода (с уходящими в бесконечность ветвями функции);
- число разрывов первого рода (скачков) должно быть конечным;
- число экстремумов должно быть конечным (в качестве примера функции, которая на конечном интервале имеет бесконечное число экстремумов, можно привести sin(l/x) в окрестности нуля).

При этом в зависимости от конкретной формы базисных функций различают несколько форм записи ряда Фурье.

Замечание. *Ряд Фурье используется для представления не только периодических сигналов, но и сигналов конечной длительности. При этом оговаривается временной интервал, для которого строится ряд Фурье, а в остальные моменты времени сигнал считается равным нулю. Для расчета коэффициентов ряда такой подход фактически означает* ***периодическое продолжение*** *сигнала за границами рассматриваемого интервала.*

1.3.1 Синусно-косинусная форма ряда Фурье:

Тригонометрический ряд: {1, cos*nx*, sin*nx*, (n=1,2,...)}

В этой форме ряд Фурье имеет следующий вид:

$$s(t)=\frac{a_0}{2}+\sum_{k=1}^{\infty}(a_k cos(k\omega_1 t)+b_k sin(k\omega_1 t))=\frac{a_0}{2}+\sum_{k=1}^{\infty}(a_k cos(k\frac{2\pi t}{T})+b_k sin(k\frac{2\pi t}{T})). \quad (1.6)$$

Здесь $\omega_1=\frac{2\pi}{T}$ — круговая частота, соответствующая периоду повторения сигнала, равному T. Входящие в формулу кратные ей частоты $k\omega_1$, называются *гармониками;* гармоники нумеруются в соответствии с индексом k; частота $\omega_k = k\omega_1$ называется k-й гармоникой сигнала. Коэффициенты ряда a_k и b_k рассчитываются по формулам:

$$a_k=\frac{2}{T}\int_{-T/2}^{T/2} s(t)cos(k\omega_1 t)dt,$$

$$b_k=\frac{2}{T}\int_{-T/2}^{T/2} s(t)sin(k\omega_1 t)dt.$$

Константа a_0 рассчитывается по общей формуле для $a_к$. Ради этой общности и введена несколько странная на первый взгляд форма записи постоянного слагаемого (с делением на два). Само же это слагаемое представляет собой среднее значение сигнала на периоде:

$$\frac{a_0}{2}=\frac{1}{T}\int_{-T/2}^{T/2} s(t)dt\ .$$

Замечание. *Пределы интегрирования не обязательно должны быть такими, как в приведенных выше формулах (от -T/2 до T/2). Интегрирование может производиться по любому интервалу длиной T — результат от этого не изменится. Конкретные пределы выбираются из соображений удобства вычислений; например, может оказаться удобнее выполнять интегрирование от 0 до T или от -T до 0.*

Если s(t) является четной функцией, то все b_k будут равны нулю и в формуле ряда Фурье будут присутствовать только ***косинусные*** *слагаемые. Если s(t) является* ***нечетной*** *функцией, равны нулю будут, наоборот, косинусные коэффициенты a_k и в формуле останутся лишь* ***синусные*** *слагаемые.*

Пример 1.3 Для иллюстрации предположим, что $g(t) = t$ и что используется интервал $-\pi \le t \le \pi$. Поскольку подынтегральное выражение нечетное, а интервал интегрирования симметричный относительно $t = 0$, то имеем

$$a_m = \frac{1}{\pi}\int_{-\pi}^{\pi} t\cos mt dt = 0.$$

Для b_m получим

$$b_m = \frac{1}{\pi}\int_{-\pi}^{\pi} t\sin mt dt.$$

Интегрирование по частям дает

$$b_m = \frac{1}{\pi}\left[\frac{t(-\cos mt)}{m}\Big|_{-\pi}^{\pi} + \frac{1}{m}\int_{-\pi}^{\pi}\cos mt dt\right] =$$

$$= \frac{1}{\pi}\left[\frac{\pi}{m}2(-1)^{m+1} + \frac{1}{m}\frac{\sin mt}{m}\Big|_{-\pi}^{\pi}\right] = \frac{2}{m}(-1)^{m+1}.$$

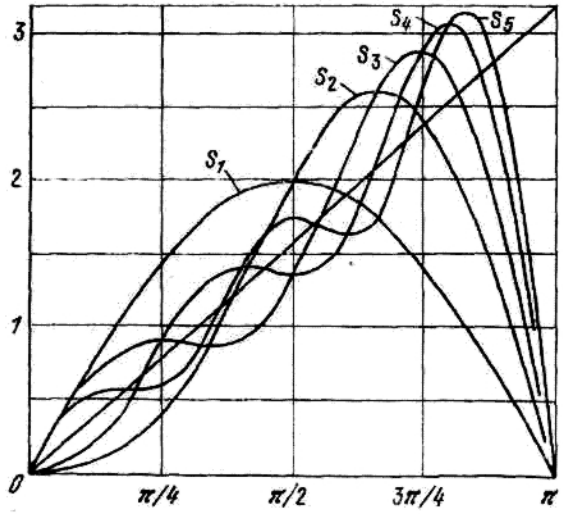


Рис. Пр 1.3 - Частичные суммы $S_N = \sum_{n=1}^{N}(-1)^n \frac{\sin(nt)}{n}$

Следовательно, получаем формальное разложение

$$t = 2\left[\sin t - \frac{\sin 2t}{2} + \frac{\sin 3t}{3} - \frac{\sin 4t}{4} + \ldots\right].$$

На рис. Пр 1.3 показаны несколько первых частичных сумм $S_{N,}$ в качестве приближений к функции $g(t) = t$.

Отметим эффект периодичности на концах интервала $-\pi \le t \le \pi$.

Пример 1.4. В качестве второй иллюстрации разложения заданной функции в формальный ряд Фурье рассмотрим прямоугольный импульс

$$g(t) = \begin{cases} \frac{1}{2}, & 0 \le t \le \pi \\ -\frac{1}{2}, & -\pi \le t \le 0. \end{cases}$$

Поскольку $g(-t) = -g(t),$ то в разложении не будет членов с косинусами. Коэффициенты синусных членов задаются выражением

$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} g(t)\sin ktdt = \frac{1}{\pi}\int_{0}^{\pi} \sin ktdt.$$

Выполнив интегрирование, получим

$$b_k = \frac{1}{\pi}\frac{\left|1+(-1)^{k+1}\right|}{k} = \begin{cases} \frac{2}{\pi k}, & k - нечетное \\ 0, & k - четное. \end{cases}$$

Таким образом, имеем формальное разложение

$$g(t) = \frac{2}{\pi}\left[\sin t + \frac{1}{3}\sin 3t + \frac{1}{5}\sin 5t + \ldots\right] =$$

$$= \frac{2}{\pi}\sum_{k=0}^{\infty}\frac{1}{2k+1}\sin(2k+1)t.$$

На рис. Пр. 1.4 приведены графики частичных сумм (обозначенных S_N) для 1, 5 и 9 членов ряда на интервале $0 \le t < \pi$. Для $-\pi < t \le 0$ кривые будут отрицательными по отношению к показанным. Рисунок иллюстрирует качество аппроксимации.

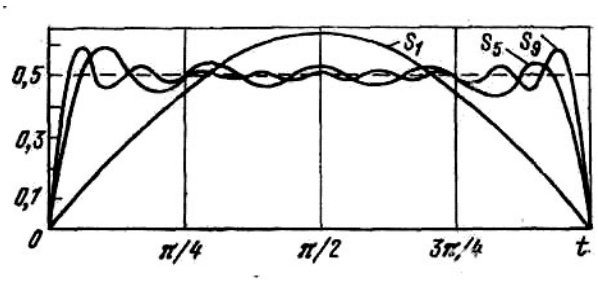


Рис.1.4

Рис. Пр 1.4- Частичные суммы S_1, S_5, S_9 для прямоугольного импульса

1.3.2 Нечетные и четные функции

Нечетные и четные периодические функции встречаются так часто, что их ряды Фурье заслуживают особого рассмотрения. Тот факт, что произвольная функция $g(t)$ может быть записана как сумма нечетной и четной функций $g(t) = \frac{[g(t)-g(-t)]}{2} + \frac{[g(t)+g(-t)]}{2}$, делает это рассмотрение особенно важным.

Для нечетной функции видно, что a_k=0 для всех *к,* тогда как

$$b_k = \frac{2}{\pi}\int_{0}^{\pi} g(t)\sin ktdt.$$

Дифференцирующие фильтры (*см. далее*) представляют собой типичные нечетные функции.

Для четной функции имеем

$$a_k = \frac{2}{\pi}\int_0^{\pi} g(t)\cos ktdt,$$

а все b_k=0. Типовой сглаживающий фильтр (*см. далее*) представляет собой четную функцию.

Используя дополнительно соответствующую симметрию относительно $t = \pi/2$, можно получить ряд Фурье, содержащий коэффициенты только с нечетными индексами или только с четными индексами. Другие регулярные схемы ненулевых коэффициентов могут быть также получены при соответствующем введении дополнительной нечетной и четной симметрии в определение функции.

1.3.3 Вещественная форма ряда Фурье

Некоторое неудобство использования в расчетах синусно-косинусной формы ряда Фурье состоит в том, что для каждого значения индекса суммирования k (то есть для каждой гармоники с частотой $k\omega_1$) в формуле фигурируют два слагаемых — синус и косинус. Воспользовавшись формулами тригонометрических преобразований, сумму этих двух слагаемых можно трансформировать в косинус (или синус) той же частоты с иной амплитудой и некоторой начальной фазой:

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} A_k \cos(k\omega_1 t + \varphi_k). \qquad (1.7)$$

Если $s(t)$ является четной функцией, фазы φ_k могут принимать только значения 0 и π, а если $s(t)$ — функция нечетная, то возможные значения для фазы равны $\pm\pi/2$.

1.3.4 Комплексная форма

Данная форма представления ряда Фурье является наиболее употребимой в различных приложениях. Она может быть получена из вещественной формы путем представления косинуса в виде полусуммы комплексных экспонент (такое представление вытекает из формулы Леонарда Эйлера):

$$e^{jx} = \cos x + j\sin x, \quad \cos x = \frac{1}{2}(e^{jx} + e^{-jx}).$$

Применив данное преобразование к вещественной форме ряда Фурье, получим суммы комплексных экспонент с положительными и отрицательными показателями:

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \frac{A_k}{2}(\exp(jk\omega_1 t + j\varphi_k) + \exp(-jk\omega_1 t - j\varphi_k)).$$

При этом экспоненты со знаком минус в показателе трактуются как члены ряда с отрицательными номерами. В рамках такого общего подхода постоянное слагаемое $a_o/2$ становится членом ряда с нулевым номером. В результате получается комплексная форма записи ряда Фурье:

$$s(t) = \sum_{k=-\infty}^{\infty} \dot{C}_k e^{-jk\omega_1 t}. \qquad (1.8)$$

Комплексные коэффициенты ряда связаны с амплитудами A_k и фазами φ_k, фигурирующими в вещественной форме записи ряда Фурье (1.7), следующими несложными соотношениями:

$$\dot{C}_k = \frac{1}{2} A_k e^{-j\varphi_k},$$

$$A_k = 2\left|\dot{C}_k\right|, \quad \varphi_k = arg(\dot{C}_k).$$

Можно выписать и формулы связи с коэффициентами a_k и b_k синусно-косинусной формы ряда Фурье (1.6):

$$\dot{C}_k = \frac{a_k}{2} - j\frac{b_k}{2},$$

$$a_k = 2\,Re(\dot{C}_k), \quad b_k = -2\,Im(\dot{C}_k).$$

Замечание. *Точка в обозначении амплитуд экспонент $\dot{C}_k$ означает их комплексность.*

Покажем, как из косинусно-синусной формы ряда Фурье может быть получена форма ряда вида (1.8):

$$cos\,nx = \frac{e^{jnx} + e^{-jnx}}{2} \quad u \quad sin\,nx = \frac{e^{jnx} - e^{-jnx}}{2j},$$

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n\,cos\,nx + b_n\,sin\,nx =$$

$$= \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \frac{e^{jnx} + e^{-jnx}}{2} - jb_n \frac{e^{jnx} - e^{-jnx}}{2}\right) = \sum_{n=-\infty}^{\infty} \dot{C}_n e^{jnx},$$

где $C_0 = \frac{a_0}{2}$ *и при* $n > 1$ $\dot{C}_n = \frac{a_n - jb_n}{2}$; $\dot{C}_{-n} = \frac{a_n + jb_n}{2}$.

$$s(t) = \sum_{n=-\infty}^{\infty} \dot{C}_n e^{jnx} = \sum_{k=-\infty}^{\infty} \dot{C}_k e^{-jk\omega_1 t}.$$

$$\dot{C}_k = \frac{1}{2}(a_k - jb_k),$$

$$a_k = 2\,Re(\dot{C}_k), \quad b_k = -2\,Im(\dot{C}_k).$$

Отсюда

$$\dot{C}_k = \frac{1}{T}\int_{-T/2}^{T/2} s(t)\cdot e^{-jk\omega_1 t}\,dt \qquad (1.9)$$

- формула для вычисления коэффициентов в комплексной форме ряда Фурье.

Если в формуле (1.9) $T \to \infty$, то периодический сигнал по сути превращается в непериодический (сумма в выражении ряда Фурье в пределе

переходит в интеграл) и имеет место ***преобразование Фурье*** *(см. подраздел 1.3.5)*. При этом $\omega_1 = \frac{1}{T} \to 0$, т.е. спектр становится непрерывным.

С другой стороны, если *s*(*t*) является *четной* функцией, коэффициенты ряда C_k будут чисто *вещественными,* а если *s*(*t*) — функция *нечетная,* коэффициенты ряда окажутся чисто *мнимыми.*

Совокупность амплитуд гармоник ряда Фурье называют *амплитудным спектром* $A_k = \frac{1}{2}\sqrt{a_k^2 + b_k^2}$, а совокупность их фаз — *фазовым спектром* $\varphi_k = arctg\left(-\frac{b_k}{a_k}\right)$. Эти понятия не следует путать с амплитудно- и фазочастотными *характеристиками,* которые относятся не к сигналам, а к системам, формирующим и/или преобразующим сигналы, реализованные в виде *электрических или радиотехнических цепей,* представляющих из себя в частотном отношении так называемые *формирующие фильтры* различного назначения [2,6,11-13].

Если анализируемый сигнал *s(t)* является вещественным, то его амплитудный и фазовый спектры обладают симметрией:

$$A_{-k} = A_k, \quad \varphi_{-k} = -\varphi_k, \quad \dot{C}_{-k} = \dot{C}_k.$$

При $k \to \infty$, $\omega_k = \frac{1}{T_k} \to \infty$. В то же время *A(ω) и φ(ω)* - это есть непрерывные, амплитудный и фазовый спектры сигнала, соответственно.

В качестве *примера* разложения сигналов в ряд Фурье рассмотрим последовательность прямоугольных импульсов с амплитудой *A,* длительностью *τ* и периодом повторения *T* [2] .

Начало отсчета времени примем расположенным в середине импульса (рис. 1.6).

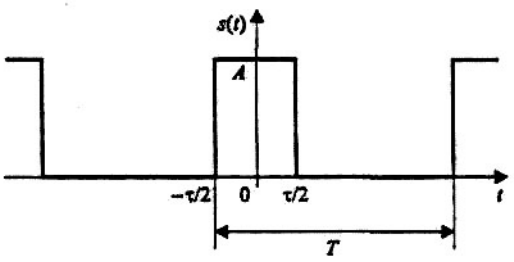


Рис. 1.6 - Периодическая последовательность прямоугольных импульсов

Сигнал является четной функцией, поэтому для его представления удобнее использовать синусно-косинусную форму ряда Фурье — в ней будут присутствовать только косинусные слагаемые a_k, равные

$$a_k = \frac{2}{T}\int_{-\tau/2}^{\tau/2} A\cos\left(\frac{2\pi k}{T}t\right)dt = \frac{2A}{\pi k}\sin\left(\frac{\pi k\tau}{T}\right).$$

Из полученного выражения видно, что длительность импульсов и период их следования входят в нее не обособленно, а исключительно в виде отношения. Этот параметр — отношение периода к длительности импульсов — называют *скважностью* последовательности импульсов и обозначают буквой *q: q = T/τ*. Введем этот параметр в полученную формулу для коэффициентов ряда Фурье, а затем приведем формулу к виду sin(*x*)/*x*.

$$a_k = \frac{2A}{\pi k} sin\left(\frac{\pi k}{q}\right) = \frac{2A}{q} \frac{sin\left(\frac{\pi k}{q}\right)}{\frac{\pi k}{q}} \qquad (1.10)$$

Замечание. *В зарубежной литературе вместо скважности используется обратная величина, называемая коэффициентом заполнения (duty cycle) и равная τ/T.*

При такой форме записи становится хорошо видно, чему равно значение постоянного слагаемого ряда: поскольку при $x \to 0$ sin(x)/x $\to$ 1, то $\frac{a_0}{2} = \frac{A}{q} = \frac{A\tau}{T}$.

Представление последовательности прямоугольных импульсов в виде ряда Фурье будет иметь вид:

$$s(t) = \frac{A}{q} + \sum_{k=1}^{\infty} \frac{2A}{\pi k} sin\left(\frac{\pi k}{q}\right) cos\left(\frac{2\pi k}{T} t\right).$$

Амплитуды гармонических слагаемых ряда зависят от номера гармоники по закону sin(*x*)/*x* (рис. 1.7).

График функции sinc(*x*)=sin(*x*)/*x* имеет лепестковый характер.

В свою очередь, следует подчеркнуть, что для графиков дискретных спектров периодических сигналов возможны два варианта градуировки горизонтальной оси — в номерах гармоник и в частотах. На рис. 1.7 градуировка оси соответствует номерам гармоник, а частотные параметры спектра нанесены на график с помощью размерных линий.

Итак, ширина лепестков, измеренная в количестве гармоник, равна скважности последовательности (при $k = nq$ имеем sin(*nk/q*) = 0, если $n \neq 0$). Отсюда следует ***важное свойство спектра последовательности прямоугольных импульсов — в нем отсутствуют (т.е. имеют нулевые амплитуды) гармоники с номерами, кратными скважности.***

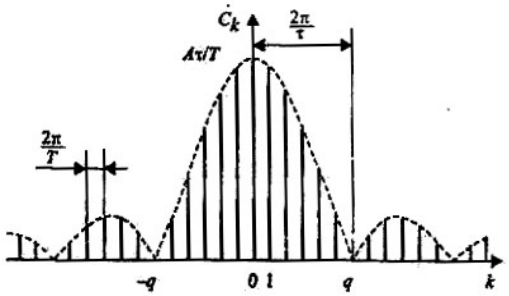


Рис. 1.7 - Коэффициенты ряда Фурье для последовательности

прямоугольных импульсов

Расстояние по частоте между соседними гармониками равно частоте следования импульсов - $2\pi/T$. Ширина лепестков спектра, измеренная в единицах частоты равна $2\pi/\tau$, то есть обратно пропорциональна длительности импульсов. Это как будет показано далее, ***проявление общего закона: чем короче сигнал, тем шире его спектр.***

Покажем на примере *m-file: harmonics.m Matlab*-реализации меандра (*меандр* - последовательность прямоугольных импульсов со скважностью, равной двум), как складывается заданный сигнал из отдельных гармоник (рис. 1.8):

```
N = 10;                  % число ненулевых гармоник
    t = -1:0.01:1;       % вектор моментов времени
A = 1;                   % амплитуда
T = 0.8;                 % период
    nh = (1:N)*2-1;      % номера ненулевых гармоник
  % строки массива – колебания отдельных гармоник
    harmonics = cos(2*pi*nh'*t/T);
Am = 2/pi./nh;           % амплитуды гармоник
Am(2:2:end) = -Am(2:2:end);      % чередование знаков
   s1 = harmonics.* repmat(Am',1, length(t));
% строки - частичные суммы гармоник
   s2 = cumsum(s1);
for k=1:N;
subplot (ceil(N/2), 2, k), plot(t, s2(k,:));
end
```

Как видно, последовательность прямоугольных импульсов плохо подходит для представления рядом Фурье — она содержит скачки, а сумма любого числа гармонических составляющих с любыми амплитудами всегда будет непрерывной функцией. Поэтому поведение ряда Фурье в окрестностях разрывов представляет особый интерес. На графиках рис. 1.8 хорошо видно, что в окрестности точки разрыва суммирование ряда Фурье дает наклонный участок, причем крутизна наклона возрастает с ростом числа суммируемых гармоник. В самой точке разрыва ряд Фурье сходится к полусумме правого и левого пределов:

$$s'(t_0) = \frac{1}{2}\left(\lim_{t \to t_0 - 0} s(t) + \lim_{t \to t_0 + 0} s(t) \right).$$

Здесь $s(t)$ — исходный сигнал, $s'(t)$ — сумма ряда Фурье для него.

На примыкающих к разрыву участках сумма ряда Фурье дает заметные пульсации**,** причем на графиках рис. 1.8 заметно [2], что амплитуда этих пульсаций не уменьшается с ростом числа суммируемых гармоник — пульсации лишь сжимаются по горизонтали, приближаясь к точке разрыва. Это явление, присущее рядам Фурье для любых сигналов с разрывами первого рода (скачками), называется *эффектом Гиббса.*

Можно показать, что амплитуда первого (самого большого) выброса составляет примерно 9 % от величины скачка.

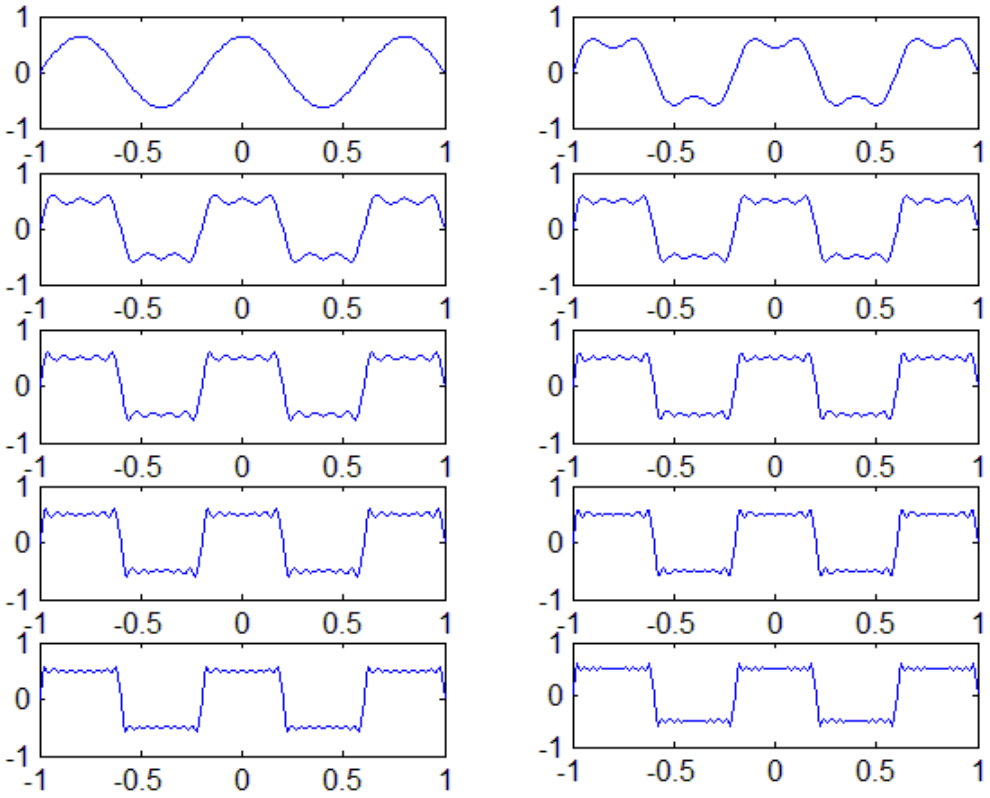


Рис. 1.8 - Промежуточные стадии суммирования ряда Фурье для меандра

1.3.5 Преобразование Фурье

Преобразование Фурье (Fourier transform) является инструментом спектрального анализа *непериодических* сигналов. Его можно применять и к сигналам периодическим, но это требует использования аппарата обобщенных функций [3].

Для наглядной иллюстрации перехода от ряда Фурье к преобразованию Фурье часто используется не вполне строгий математически, но зато понятный подход. Представим себе периодическую последовательность импульсов произвольного вида и сформируем для нее ряд Фурье. Затем, не меняя формы одиночных импульсов, увеличим период их повторения (заполнив промежутки нулевым значением) и снова рассчитаем коэффициенты ряда Фурье. Формула (1.9) для расчета коэффициентов ряда показывает, что *тот же самый* интеграл придется вычислить, но для более тесно расположенных частот $\omega_k = k\omega_1$. Изменение пределов интегрирования не играет роли — ведь на добавившемся между импульсами пространстве сигнал имеет нулевое значение. Единственное дополнительное изменение будет состоять в уменьшении общего уровня гармоник из-за деления результата интегрирования на увеличившийся период T.

На рис. 1.9 описанные изменения иллюстрируются на примере двукратного увеличения периода следования прямоугольных импульсов.

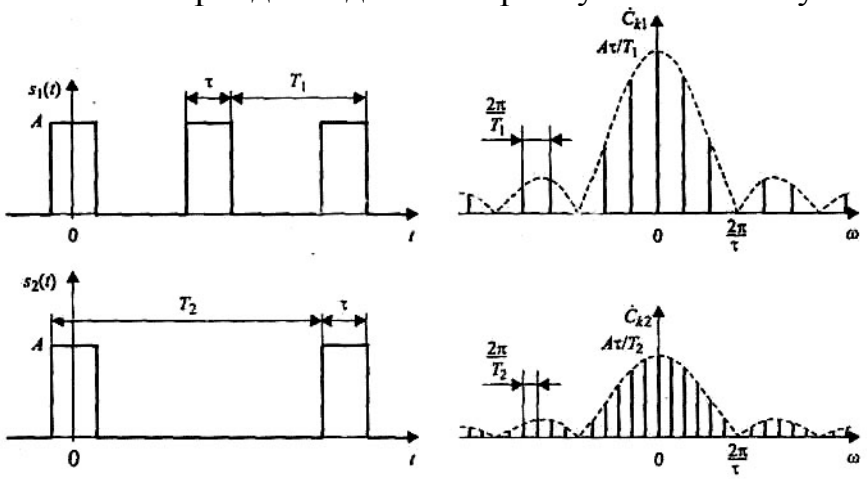


Рис. 1.9 - Изменение спектра последовательности импульсов при двукратном увеличении периода их следования

Заметим, что *горизонтальная ось спектральных графиков проградуирована в значениях частот, а не номеров гармоник.*

Итак, с ростом периода следования импульсов гармоники располагаются ближе друг к другу по частоте, а общий уровень спектральных составляющих становится все меньше. При этом вид вычисляемого интеграла (1.9) не меняется.

Если устремить период к бесконечности (превратив тем самым периодическую последовательность в одиночный импульс), гармоники спектра будут плотно занимать всю частотную ось, а их амплитуды упадут до нуля (станут бесконечно малыми). В то же время *взаимное соотношение* между уровнями гармоник остается неизменным и определяется все тем же интегралом (1.9). В связи с этим при спектральном анализе непериодических сигналов формула для расчета коэффициентов комплексного ряда Фурье модифицируется следующим образом:

1) частота перестает быть дискретно меняющейся и становится непрерывным параметром преобразования (т.е. $k\omega_1$ в формуле (1.9) заменяется на ω);
2) удаляется множитель $1/T$;
3) результатом вычислений вместо нумерованных коэффициентов ряда $\dot{C}_k$, является функция частоты $\dot{S}(\omega)$ — *спектральная функция* сигнала *s(t)*, которую также ещё называют *спектральной плотностью.*

В результате перечисленных модификаций формула (1.9) превращается в формулу *прямого преобразования Фурье (Transform Fourier):*

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t}dt \;=\; \{\mathrm{TF}[s(t)]\}. \qquad (1.11)$$

В формуле самого ряда Фурье суммирование, естественно, заменяется интегрированием (и, кроме того, перед интегралом появляется множитель $1/2\pi$). Получающееся выражение называется *обратным преобразованием Фурье:*

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \dot{S}(\omega) e^{j\omega t} d\omega \quad = \quad \{\mathrm{TF}^{-1}[S(\omega)]\} \quad . \tag{1.12}$$

Замечание. *Если использовать не круговую частоту ω, а обычную частоту f = ω/(2π), формулы прямого и обратного преобразования Фурье становятся еще более симметричными, отличаясь друг от друга, лишь знаком в показателе экспоненты:*

$$\dot{F}(f) = \dot{S}(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt,$$

$$s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \dot{S}(f) e^{j2\pi ft} df \ , \{\dot{F}(f) \rightleftharpoons s(t)\}$$

- операция преобразования сигнала из частотного диапазона во временной:

Для того чтобы преобразование Фурье было применимо, сигнал должен удовлетворять следующим требованиям:

□ должны выполняться условия Дирихле;

□ сигнал должен быть *абсолютно интегрируемым*. Это означает, что интеграл от его модуля должен быть конечной величиной:

$$\int_{-\infty}^{\infty} |s(t)| dt < \infty.$$

Если анализируемый сигнал *s(t)* — вещественная функция, то соответствующая спектральная функция $\dot{S}(\omega)$ является сопряженно-симметричной относительно нулевой частоты. Это означает, что значения спектральной функции на частотах ω и $-\omega$ являются комплексно-сопряженными по отношению друг к другу:

$$\dot{S}(-\omega) = \dot{S}'(\omega).$$

Если $s(t)$ – *четная* непериодическая функция, то, как и в случае периодической функции, спектр будет чисто *вещественным* (и, следовательно, будет являться *четной* функцией). Если, напротив, *s(t)* – функция *нечетная,* то спектральная функция $\dot{S}(\omega)$ будет чисто *мнимой* (и *нечетной).*

Модуль спектральной функции представляет собой *амплитудный спектр,* а ее аргумент — есть *фазовый спектр.* Легко показать, что для вещественного сигнала амплитудный спектр является четной, а фазовый — нечетной функцией частоты:

$$|\dot{S}(-\omega)| = |\dot{S}(\omega)|, \qquad \varphi_s(-\omega) = -\varphi_s(\omega).$$

Таким образом, преобразование Фурье (1.11) ставит в соответствие сигналу, заданному во времени, его спектральную функцию. При этом осуществляется переход из *временной области* в *частотную.* ***Преобразование Фурье является взаимно-однозначным, поэтому представление сигнала в частотной области (спектральная функция) содержит ровно столько же информации, сколько и исходный сигнал, заданный во временной области.***

Примеры расчета преобразования Фурье

Рассмотрим примеры расчета преобразования Фурье для некоторых сигналов, часто встречающихся при решении различных задач.

Прямоугольный импульс

Рассмотрим прямоугольный импульс, центрированный относительно начала отсчета времени (рис. 1.8):

$$s(t)=\begin{cases} A, & |t|\le \tau/2, \\ 0, & |t|>\tau/2. \end{cases}$$

$$\sin\left(\frac{\omega t}{2}\right)=\frac{e^{\frac{\omega t}{2}}-je^{-\frac{\omega t}{2}}}{2}$$

$$A\int_{-\tau/2}^{\tau/2} e^{-j\omega t}dt = A\frac{1}{j\omega}e^{-j\omega t}\Big|_{-\tau/2}^{\tau/2}$$

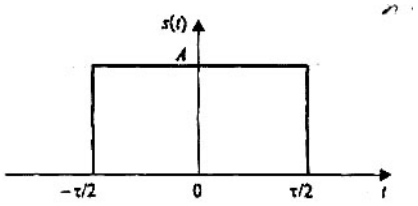


Рис. 1.8 - Прямоугольный импульс

Вычислим спектральную функцию:

$$\dot{S}(\omega)=\int_{-\tau/2}^{\tau/2} Ae^{-j\omega t}dt=\frac{2A}{\omega}\sin\left(\frac{\omega\tau}{2}\right)=A\tau\frac{\sin(\omega\tau/2)}{\omega\tau/2}=\Phi(b)-\Phi(a).$$

Как видно, спектр представляет собой функцию вида $\sin(x)/x$ (рис. 1.9). Амплитудный спектр имеет лепестковый характер с шириной лепестков, равной $2\pi/\tau$, то есть обратно пропорциональной длительности импульса. При этом значение спектральной функции на нулевой частоте равно площади импульса – *Ax.* Спектральная функция является вещественной, поэтому фазовый спектр принимает лишь два значения – 0 и π, в зависимости от знака функции $\sin(x)/x$. Значения фазы π и $-\pi$ неразличимы, разные знаки для фазового спектра при $\omega > 0$ и $\omega < 0$ использованы лишь с целью представления его в виде нечетной функции.

Далее исследуем, что изменится после сдвига импульса во времени. Зададим начало импульса в нулевой момент времени (рис. 1.10):

$$s(t)=\begin{cases}A, & 0\le t\le\tau,\\ 0, & t<0, t>\tau.\end{cases} \qquad (1.13)$$

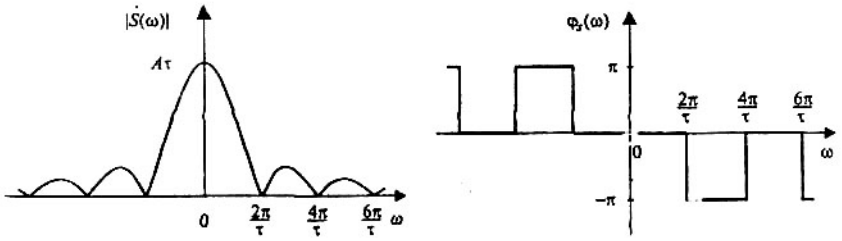


Рис.1.9 - Амплитудный (слева) и фазовый (справа) спектры прямоугольного импульса

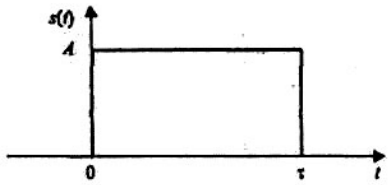


Рис. 1.10 - Прямоугольный импульс, задержанный во времени

Вычислим преобразование Фурье и построим графики амплитудного и фазового спектров (рис 1.11):

$$\dot{S}(\omega)=\int_0^{\tau}Ae^{-j\omega t}dt=\frac{A}{j\omega}(1-e^{-j\omega\tau})=A\tau\frac{\sin(\omega\tau/2)}{\omega\tau/2}\exp\left(-j\frac{\omega\tau}{2}\right). \qquad (1.14)$$

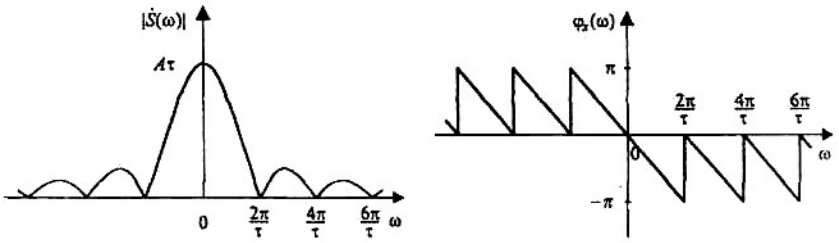


Рис. 1.11- Амплитудный (слева) и фазовый (справа) спектры задержанного прямоугольного импульса

Замечание. *Этот пример демонстрирует проявление свойства преобразования Фурье, касающегося изменения спектра при сдвиге сигнала во времени. Это свойство в общем виде будет рассмотрено в разделе «Свойства преобразования Фурье».*

Из формулы (1.14) и графиков рис. 1.11 видно, что после сдвига импульса во времени его амплитудный спектр остался прежним, а фазовый - приобрел сдвиг, линейно зависящий от частоты.

В виду того, что спектр данного сигнала простирается до бесконечности, лишь постепенно затухая, вводят понятие *эффективной ширины спектра.* Как видно из графиков, спектр имеет лепестковый характер и ширина главного лепестка равна $2\pi/\tau$. *При лепестковом характере спектра за его эффективную ширину принимают ширину главного лепестка.* Из графиков видно, что она составляет $2\pi/\tau$, то есть обратно пропорциональна длительности импульса. Это общее соотношение: чем короче сигнал, тем шире его спектр. Произведение же эффективных значений длительности сигнала и ширины его спектра, называемое *базой* сигнала, остается равным некоторой константе, зависящей только от конкретного способа определения этих параметров. В нашем примере это произведение, как видно, равно 2π. В целом же, для сигналов *простой формы* (не имеющих сложной внутриимпульсной структуры) величина их базы независимо от способа определения эффективных значений длительности и ширины спектра составляет несколько единиц.

Длительность сигнала и ширина его спектра подчиняются *соотношению неопределенности,* гласящему, что произведение этих параметров (*база сигнала*) не может быть меньше единицы. Ограничений максимального значения базы сигнала не существует. Отсюда следует, что можно сформировать сигнал большой длительности, одновременно имеющий и широкий спектр (такие сигналы называют *широкополосными,* или *сложными,* или *сигналами с большой базой).* В то же время короткий сигнал с узким спектром, согласно соотношению неопределенности, существовать не может.

Гауссов импульс

Рассмотрим еще один важный сигнал — гауссов импульс (рис. 1.12), зачастую используемый как тестовый. Как и предыдущий, он имеет бесконечную протяженность в обоих направлениях временной оси: $s(t) = Ae^{-a^2t^2}$.

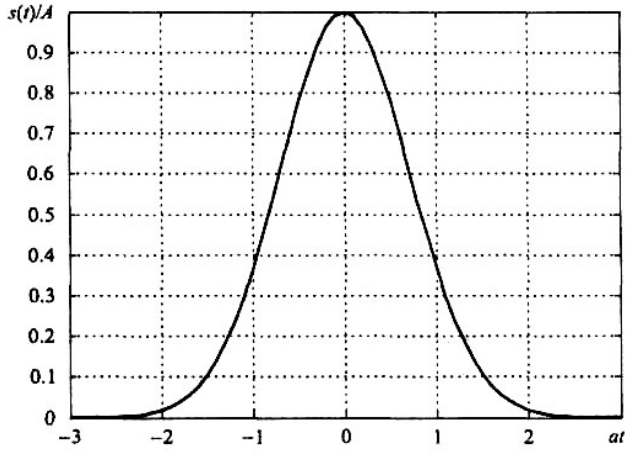


Рис. 1.12 - Гауссов импульс

Спектр:

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} Ae^{-a^2t^2} e^{-j\omega t} dt = \frac{A\sqrt{\pi}}{a} \exp\left(-\frac{\omega^2}{4a^2}\right).$$

Поскольку сигнал является четной функцией, его спектр, как и в предыдущем случае - чисто вещественный.

Отметим, что важным свойством гауссова импульса является то, что его спектр тоже описывается гауссовой функцией.

Гауссов импульс имеет бесконечную протяженность, как во временной, так и в частотной областях. Определим его эффективную длительность и ширину спектра по уровню $1/e$ от максимума: $\tau = 2/\alpha$, $\Delta\omega = 2\alpha$. База сигнала, таким образом, равна четырем (для сравнения, у прямоугольного импульса - 2π).

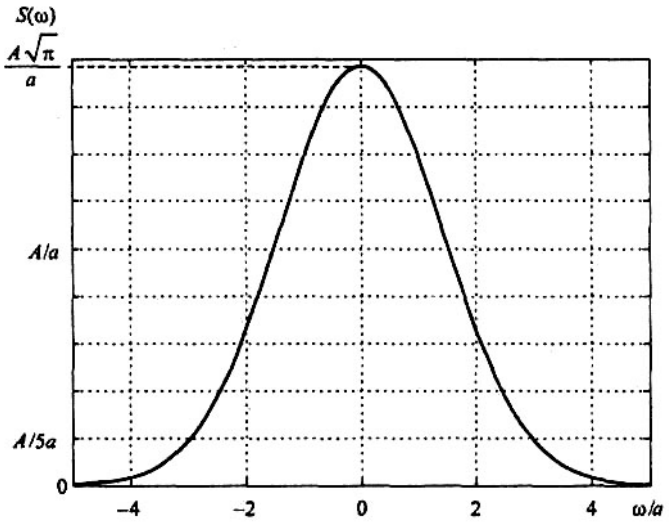


Рис. 1.13 - Амплитудный спектр гауссова импульса

Сигнал вида sin(x)/x

Следующий пример призван продемонстрировать *дуальность* преобразования Фурье. Если сравнить формулы прямого и обратного преобразования Фурье, можно заметить, что они отличаются друг от друга лишь знаком в показателе комплексной экспоненты и множителем перед интегралом. Отсюда следует, что если *четной* функции времени $f(t)$ соответствует спектральная функция $\dot{F}(\omega)$ (она будет также четной), то функции времени $g(t)$ будет соответствовать спектральная функция $2\pi\dot{G}(\omega)$. Проверим это на конкретном примере. В начале этого раздела мы выяснили, что прямоугольному импульсу соответствует спектральная функция вида $\sin(\omega)/\omega$. Теперь же рассмотрим временной сигнал вида $\sin(t)/t$ и проверим, будет ли его спектральная функция прямоугольной (рис. 1.14):

$$s(t) = A\frac{\sin(\pi t / T)}{\pi t / T}.$$

Рассчитываем спектр и строим график (рис. 1.15):

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} A\frac{\sin(\pi t / T)}{\pi t / T} e^{-j\omega t} dt = A\int_{-\infty}^{\infty} \frac{\sin(\pi t / T)\cos\omega t}{\pi t / T} dt =$$

$$= \frac{AT}{2\pi}\int_{-\infty}^{\infty} \frac{\sin\left(\omega + \frac{\pi}{T}\right)t + \sin\left(\omega - \frac{\pi}{T}\right)t}{t} dt =$$

$$= \frac{AT}{2\pi}\int_{-\infty}^{\infty} \frac{\sin\left(\omega + \frac{\pi}{T}\right)t}{t} dt + \frac{AT}{2\pi}\int_{-\infty}^{\infty} \frac{\sin\left(\omega - \frac{\pi}{T}\right)t}{t} dt$$

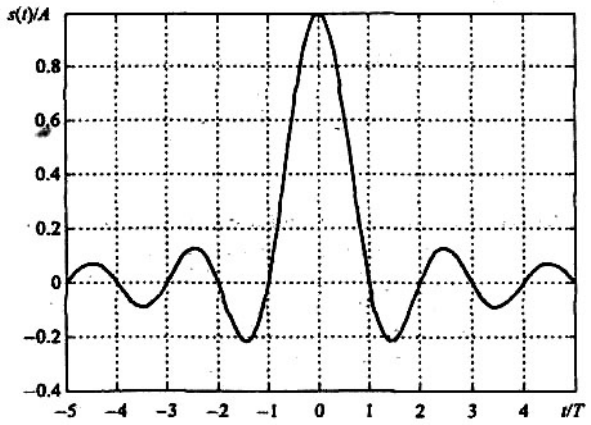


Рис. 1.14 - Сигнал вида sin(at)/(at)

Значение каждого из двух получившихся интегралов равно ± π в зависимости от знака множителей ($\omega \pm \pi/T$). Отсюда результат суммирования интегралов зависит от частоты следующим образом:

$$\dot{S}(\omega) = \begin{cases} AT, & |\omega| \le \dfrac{\pi}{T}, \\ 0, & |\omega| > \dfrac{\pi}{T}. \end{cases}$$

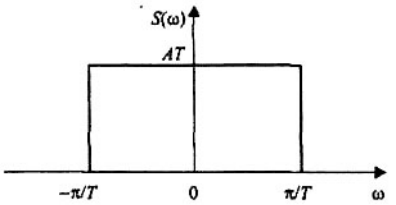


Рис. 1.15 - Сигнал вида sin(at)/(at) имеет прямоугольный спектр

Как видно, *дуальность* (симметрия) преобразования Фурье получила наглядное подтверждение.

В свою очередь, как следует из приведенного, сигнал данного вида имеет идеальный низкочастотный спектр – спектральная функция постоянна в некоторой полосе частот, начинающейся от нулевой частоты, и равна нулю за пределами этой полосы.

Рассмотрим основные свойства преобразования Фурье.

Свойства преобразования Фурье

Под свойствами преобразования Фурье подразумевается взаимное соответствие трансформаций сигналов и их спектров. Хорошее знание свойств преобразования Фурье позволяет предсказывать примерный (а иногда и точный) вид спектра анализируемого сигнала и, таким образом, контролировать правдоподобность модельного результата.

Рассматрим два абстрактных сигнала, $f(t)$ и $g(t)$, считая, что их спектральные функции равны $\dot{F}(\omega)$ и $\dot{G}(\omega)$, соответственно.

Линейность

Преобразование Фурье является *линейным* интегральным преобразованием. Смысл свойства линейности можно сформулировать так: спектр суммы равен сумме спектров, что соответствует *принципу суперпозиции* [7]. Говоря математическим языком, линейная комбинация сигналов имеет спектр в виде такой же (с теми же коэффициентами) линейной комбинации их спектральных функций:

$$\text{если } s(t) = \alpha f(t) + \beta g(t), \text{ то } \dot{S}(\omega) = \alpha \dot{F}(\omega) + \beta \dot{G}(\omega).$$

Задержка

Рассмотрим, как сказывается на спектральной функции задержка сигнала во времени. Пусть τ – время задержки:

$$s(t) = f(t - \tau).$$

Тогда спектральная функция:

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} f(t-\tau) e^{-j\omega t} dt = \int_{-\infty}^{\infty} f(t-\tau) e^{-j\omega(t-\tau)} d(t-\tau) e^{-j\omega\tau} = \dot{F}(\omega) e^{-j\omega\tau}.$$

Результат показывает, что спектр исходного сигнала оказался умноженным на комплексную экспоненту вида $e^{-j\omega\tau}$. Таким образом, амплитудный спектр сигнала не меняется (ведь модуль такой комплексной экспоненты равен 1; к тому же здравый смысл подсказывает, что соотношение между амплитудами спектральных составляющих из-за сдвига сигнала во времени измениться не должно). Фазовый спектр приобретает дополнительное слагаемое $-\omega\tau$, линейно зависящее от частоты.

Замечание. *Если в результате какого-либо преобразования сигнала его спектр умножается на некоторую функцию, не зависящую от преобразуемого сигнала, это означает, что данное* ***преобразование может быть*** *выполнено линейной системой с постоянными параметрами.*

Изменение масштаба оси времени

Рассматривая конкретные примеры, уже отмечалось общее правило: чем короче сигнал, тем шире его спектр. Теперь взглянем на это правило со строгих теоретических позиций. Если изменить длительность сигнала $f(t)$, сохраняя его форму, то новый сигнал s(t) следует записать как:

$$s(t) = f(at).$$

При $|a|>1$ сигнал сжимается, а при $|a|<1$ - растягивается. Если $a<0$, то дополнительно происходит зеркальное отражение сигнала относительно вертикальной оси. Посмотрим, как такое преобразование сказывается на спектре:

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} f(at)e^{-j\omega t}dt = \frac{1}{a}\int_{-\infty}^{\infty} f(at)e^{-j\frac{\omega}{a}at}d(at) = \frac{1}{a}\dot{F}\left(\frac{\omega}{a}\right).$$

Как видно, изменение длительности сигнала приводит к изменению ширины спектра в противоположную сторону (аргумент t на a умножается, а ω делится) в сочетании с увеличением (при растяжении, $a < 1$) или уменьшением (при сжатии, $a > 1$) уровня спектральных составляющих.

Полученная формула справедлива для $a > 0$. При $a < 0$ замена переменной $t \to at$ вызовет перестановку пределов интегрирования и, как следствие, изменение знака у результата:

$$\dot{S}(\omega) = -\frac{1}{a}\dot{F}\left(\frac{\omega}{a}\right), \quad a<0.$$

Объединяя оба случая, можно записать

$$\dot{S}(\omega) = -\frac{1}{|a|}\dot{F}\left(\frac{\omega}{a}\right), \quad a\neq 0.$$

В частном случае при a = -1 полученная формула дает следующее:

$$\dot{S}(\omega) = \dot{F}(-\omega) = \dot{F}^*(\omega).$$

Таким образом, зеркальное отражение сигнала относительно начала отсчета времени приводит к зеркальному отражению спектра относительно нулевой частоты. Для вещественного сигнала это соответствует комплексному сопряжению спектра.

Замечание. *В данном случае результат не сводится к умножению исходного спектра на некоторую функцию. В соответствии с предыдущим замечанием это означает, что* ***изменение длительности сигнала не может быть*** *осуществлено линейной системой с постоянными параметрами.*

Дифференцирование сигнала

Посмотрим, как влияет на спектр дифференцирование сигнала во временной области. Для этого нам придется воспользоваться определением понятия производной:

$$s(t) = \frac{df}{dt} = \lim_{\varepsilon \to 0} \frac{f(t+\varepsilon) - f(t)}{\varepsilon}.$$

Применим к этому выражению преобразование Фурье:

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} \lim_{\varepsilon \to 0} \frac{f(t+\varepsilon) - f(t)}{\varepsilon} e^{-j\omega t} dt = \lim_{\varepsilon \to 0} \frac{\dot{F}(\omega) e^{-j\omega\varepsilon} - \dot{F}(\omega)}{\varepsilon} =$$

$$\dot{F}(\omega) \lim_{\varepsilon \to 0} \frac{e^{-j\omega\varepsilon} - 1}{\varepsilon} = j\omega \dot{F}(\omega).$$

Таким образом, спектр производной получается путем умножения исходного сигнала на $j\omega$. Таким образом, при дифференцировании низкие частоты ослабляются, а высокие усиливаются. Фазовый спектр сигнала сдвигается на 90° для положительных частот и на - 90° для отрицательных.

Множитель $j\omega$ называют *оператором дифференцирования сигнала в частотной области.*

Интегрирование сигнала

Интегрирование, как известно, является операцией, обратной дифференцированию. Поэтому, исходя из результатов, полученных в предыдущем разделе, можно ожидать следующий результат:

$$\dot{S}(\omega) = \frac{\dot{F}(\omega)}{j\omega}.$$

Однако детальный анализ, выполненный, например, в [4,5], показывает, что эта формула справедлива лишь для сигналов, не содержащих постоянной составляющей, у которых

$$\dot{F}(0) = \int_{-\infty}^{\infty} f(t) dt = 0.$$

В общем же случае результат должен содержать дополнительное слагаемое в виде дельта-функции на нулевой частоте. Множитель перед дельта-функцией пропорционален постоянной составляющей сигнала:

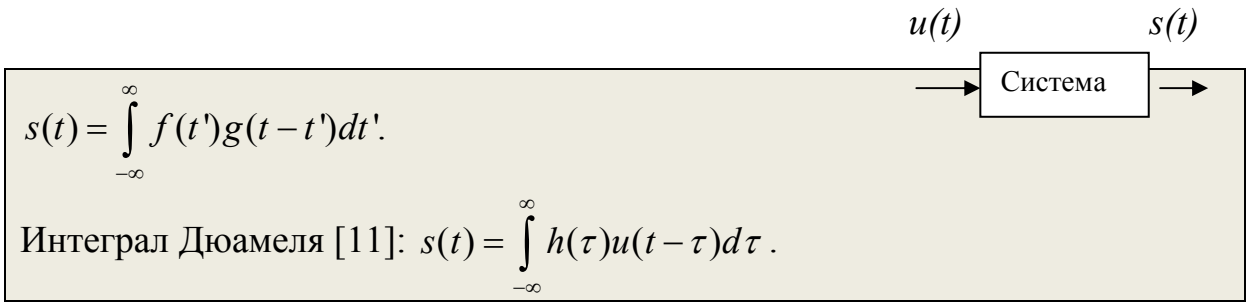
$$\dot{S}(\omega) = \frac{\dot{F}(\omega)}{j\omega} + \pi\dot{F}(0)\delta(\omega). \qquad (1.15)$$

Как результат, при интегрировании исходного сигнала высокие частоты ослабляются, а низкие усиливаются. Фазовый спектр сигнала смещается на -90° для положительных частот и на 90° для отрицательных.

Множитель $1/(j\omega)$ называют *оператором, интегрирования в частотной области.*

Спектр свертки сигналов

Свертка сигналов является очень часто используемой при обработке данных интегральной операцией, поскольку она описывает, в частности, прохождение сигнала через линейную систему с постоянными параметрами



$$s(t) = \int_{-\infty}^{\infty} f(t')g(t-t')dt'.$$

Интеграл Дюамеля [11]: $s(t) = \int_{-\infty}^{\infty} h(\tau)u(t-\tau)d\tau$.

Подвергнем такое представление сигнала преобразованию Фурье:

$$\dot{S}(\omega) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t')g(t-t')dt'e^{-j\omega t}dt =$$
$$= \int_{-\infty}^{\infty} f(t')e^{-j\omega t'}\int_{-\infty}^{\infty} g(t-t')e^{-j\omega(t-t')}d(t-t')dt' = \dot{F}(\omega)\dot{G}(\omega) \qquad (1.16)$$

Свертка: $s(t) = f(t) \otimes g(t) \Rightarrow \dot{F}[f(t) \otimes g(t)] = \dot{F}_f(\omega) \cdot \dot{F}_g(\omega)$.

Полученный результат очень важен, он часто используется на практике: *спектр свертки равен произведению спектров.*

Спектр произведения сигналов

Дуальность преобразования Фурье и соотношение (1.16), полученное в предыдущем разделе, позволяют легко предугадать результат. Покажем что:

$$s(t) = f(t)g(t).$$

Для этого найдем

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} f(t)g(t)e^{-j\omega t}dt = \int_{-\infty}^{\infty}\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\dot{F}(\omega')e^{j\omega' t}d\omega'\right)g(t)e^{-j\omega t}dt =$$
$$= \frac{1}{2\pi}\int_{-\infty}^{\infty}\dot{F}(\omega')\int_{-\infty}^{\infty}g(t)e^{-j(\omega-\omega')t}dtd\omega' = \frac{1}{2\pi}\int_{-\infty}^{\infty}\dot{F}(\omega')\dot{G}(\omega-\omega')d\omega'. \quad (1.17)$$

Как и следовало ожидать, *спектр произведения представляет собой свертку спектров.* Единственной дополнительной тонкостью является множитель $1/(2\pi)$ перед интегралом свертки.

Замечание. *При выводе соотношения (1.17) мы представили сигнал f(t) с помощью обратного преобразования Фурье (1.12) от его спектральной функции.*

Умножение сигнала на гармоническую функцию

Умножим исходный сигнал, спектр которого нам известен, на гармоническую функцию:

$$s(t) = f(t)\cos(\omega_0 t + \varphi_0).$$

При этом спектр такого сигнала:

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} f(t)\cos(\omega_0 t + \varphi_0)e^{-j\omega t}dt = \int_{-\infty}^{\infty} f(t)\frac{e^{j\omega_0 t + j\varphi_0} + e^{-j\omega_0 t - j\varphi_0}}{2}e^{-j\omega t}dt =$$
$$= \frac{1}{2}\int_{-\infty}^{\infty} f(t)e^{j\varphi_0}e^{-j(\omega-\omega_0)t}dt + \frac{1}{2}\int_{-\infty}^{\infty} f(t)e^{-j\varphi_0}e^{-j(\omega+\omega_0)t}dt = \quad (1.18)$$
$$= \frac{1}{2}e^{j\varphi_0}\dot{F}(\omega-\omega_0) + \frac{1}{2}e^{-j\varphi_0}\dot{F}(\omega+\omega_0).$$

Как видно, спектр “раздвоился” – распался на два слагаемых вдвое меньшего уровня (множитель 1/2), смещенных на ω_0 вправо (ω - ω_0) и влево (ω + ω_0) по оси частот. Кроме того, при каждом слагаемом имеется множитель, учитывающий начальную фазу гармонического колебания. Практическое применение этого свойства будет рассматриваться позже при обсуждении свойств сигналов с амплитудной модуляцией.

Связь преобразования Фурье и коэффициентов ряда Фурье

Пусть $s(t)$ — сигнал конечной длительности, а $\dot{S}(\omega)$ — его спектральная функция. Получим на основе $s(t)$ периодический сигнал, взяв период повторения T не меньше длительности сигнала:

$$s_T(t) = \sum_{k=-\infty}^{\infty} s(t - kT).$$

Сравнивая формулы (1.11) для расчета преобразования Фурье сигнала $s(t)$ и (1.9) для расчета коэффициентов ряда Фурье сигнала $s_T(t)$, можно заметить, что эти формулы предполагают вычисление одного и того же интеграла. Различие состоит лишь в том, что для расчета коэффициентов ряда Фурье в подынтегральное выражение подставляются не произвольные, а дискретные значения частоты $\omega_k = 2\pi k/T$ и, кроме того, результат интегрирования делится на период сигнала T.

Таким образом, между спектральной функцией $\dot{S}(\omega)$ одиночного импульса и коэффициентами $\dot{C}_k$ ряда Фурье для периодической последовательности таких импульсов существует простая связь:

$$\dot{C}_k = \frac{1}{T}\dot{S}\left(\frac{2\pi k}{T}\right).$$

Замечание. *Данная формула справедлива и в том случае, если период повторения импульсов меньше их длительности (то есть, если соседние импульсы периодической последовательности перекрываются).*

Фурье-анализ неинтегрируемых сигналов

При введении понятия преобразования Фурье были наложены условия его применимости: выполнение условий Дирихле и абсолютная интегрируемость сигнала. Однако в ряде случаев можно применить преобразование Фурье и к сигналам, этим условиям не удовлетворяющим, и получить при этом вполне осмысленный и практически полезный результат.

В данном разделе мы воспользуемся преобразованием Фурье для спектрального анализа таких сигналов, к которым оно формально неприменимо.

Дельта-функция

Вычислим преобразование Фурье для сигнала в виде дельта-функции в соответствии с фильтрующим свойством (1.1) которой имеем:

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1.$$

Спектр дельта-функции представляет собой константу, то есть является равномерным в бесконечной полосе частот. Это вполне согласуется с общим соотношением между длительностью сигнала и шириной его спектра: дельта-импульс имеет бесконечно малую длительность, а его спектр бесконечно широк.

Из полученного результата следует, что дельта-функцию можно записать в виде обратного преобразования Фурье следующим образом:

$$\delta(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{j\omega t} d\omega. \qquad (1.19)$$

Это полезное соотношение будет использовано при анализе следующего сигнала.

Постоянный во времени сигнал (константа)

Поскольку спектром дельта-функции является константа, то спектром константы ($s(t) = A$) благодаря дуальности преобразования Фурье должна быть дельта-функция частоты. Проверим это, воспользовавшись только что полученным соотношением (1.19):

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} Ae^{-j\omega t} dt = 2\pi A\delta(\omega).$$

Как видно, здесь опять хорошо прослеживается обратная пропорциональность между длительностью сигнала и шириной его спектра: *бесконечно протяженный сигнал имеет бесконечно узкий спектр.*

Функция единичного скачка

Функция единичного скачка (1.2) (см. раздел 1.2 - «Основные характеристики сигналов») представляет собой интеграл от дельта-функции, поэтому, в соответствии со свойствами преобразования Фурье (см. предыдущий раздел), получаем

$$\dot{S}(\omega) = \int_{-\infty}^{\infty} \sigma(t)e^{-j\omega t} dt = \pi\delta(\omega) - \frac{1}{j\omega}.$$

Поскольку дельта-функция имеет ненулевую (равную 1) постоянную составляющую, то в полном соответствии с формулой (1.15) в спектре появляется дополнительное слагаемое в виде дельта-функции на нулевой частоте.

Гармонический сигнал

Рассчитаем спектр гармонического сигнала общего вида:

$$s(t) = A\cos(\omega_0 t + \varphi).$$

Для расчета спектральной функции представим косинус в виде полусуммы комплексных экспонент и воспользуемся формулой (1.19):

$$\dot{S}(\omega)=\int_{-\infty}^{\infty}A\cos(\omega_0 t+\varphi_0)e^{-j\omega t}dt=\int_{-\infty}^{\infty}A\frac{e^{j\omega_0 t+j\varphi_0}+e^{-j\omega_0 t-j\varphi_0}}{2}e^{-j\omega t}dt=$$

$$=\int_{-\infty}^{\infty}\frac{A}{2}e^{j\varphi_0}e^{-j(\omega-\omega_0)t}dt+\int_{-\infty}^{\infty}\frac{A}{2}e^{-j\varphi_0}e^{-j(\omega+\omega_0)t}dt= \qquad (1.20)$$

$$=A\pi e^{j\varphi_0}\delta(\omega-\omega_0)+A\pi e^{-j\varphi_0}\delta(\omega+\omega_0).$$

Результат, как видим, представляет собой пару дельта-функций, расположенных на частотах $\pm\omega_0$. Множители при них отражают амплитуду и начальную фазу (то есть *комплексную амплитуду*) гармонического сигнала.

Замечание. *Тот же результат можно было бы получить, применив к спектру постоянного во времени сигнала свойство преобразования Фурье (1.18), касающееся умножения сигнала на гармоническую функцию.*

Комплексная экспонента

Впервые рассматривается сигнал, не являющийся вещественным:

$$s(t)=A\exp(j\omega_0 t).$$

Результат вычисления его спектра легко предугадать: только что рассмотренный гармонический сигнал дал спектральную функцию в виде двух дельта-функций, а косинус с помощью формулы Эйлера можно представить в виде полусуммы двух комплексных экспонент. Значит, спектром комплексной экспоненты должна являться *одиночная* дельта-функция:

$$\dot{S}(\omega)=\int_{-\infty}^{\infty}A\mathrm{e}^{j\omega_0 t}e^{-j\omega t}dt=2A\pi\delta(\omega-\omega_0). \qquad (1.21)$$

Как видно, получен ожидаемый результат. При этом следует обратить внимание на то, что поскольку сигнал не является вещественным, его спектр теряет свойство симметрии.

В свою очередь, следует отметить, что такое внимание к комплексным сигналам следует ввиду их важности для анализа модулированных сигналов, особенно при одновременном кодировании полезного сигнала и в амплитуде и в фазе.

Произвольный периодический сигнал

Как известно, периодический сигнал с периодом T может быть представлен в виде ряда Фурье (1.8):

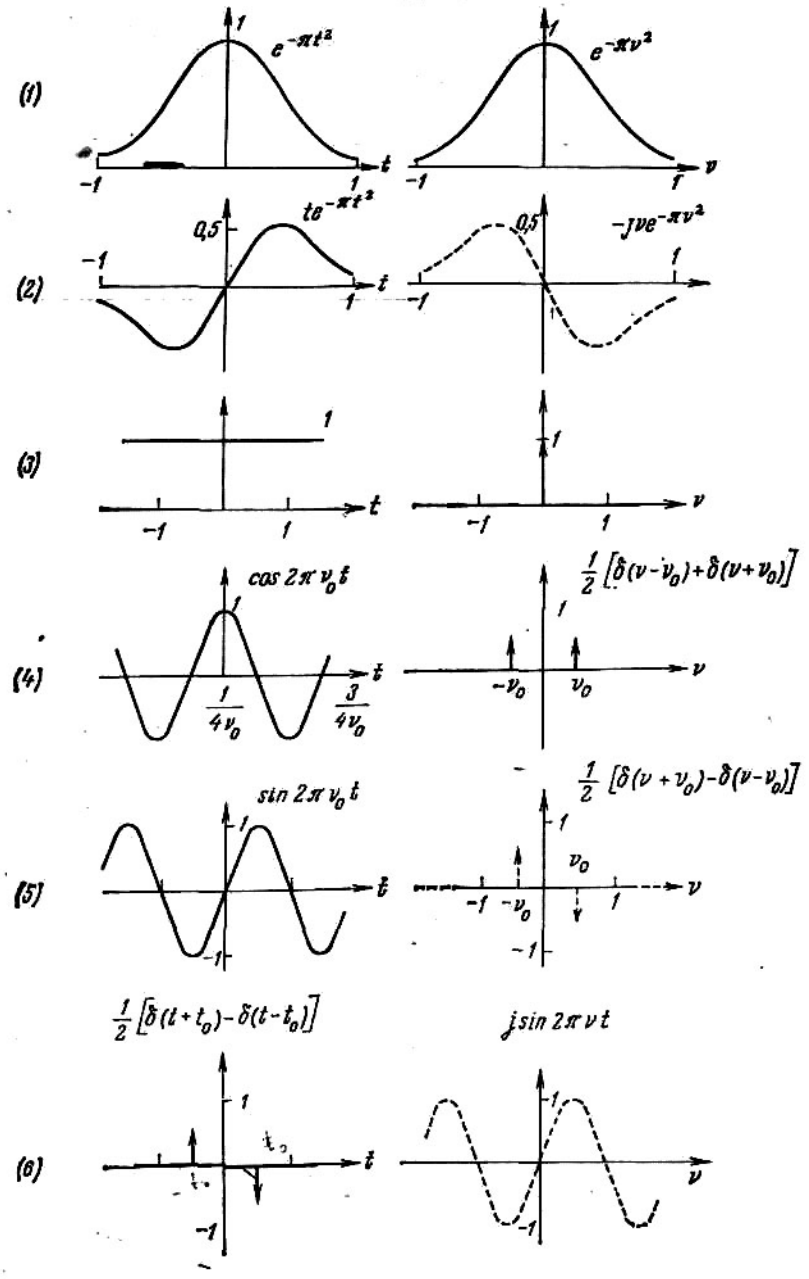
$$s(t)=\sum_{k=-\infty}^{\infty}\dot{C}_k e^{j\frac{2\pi k}{T}t}.$$

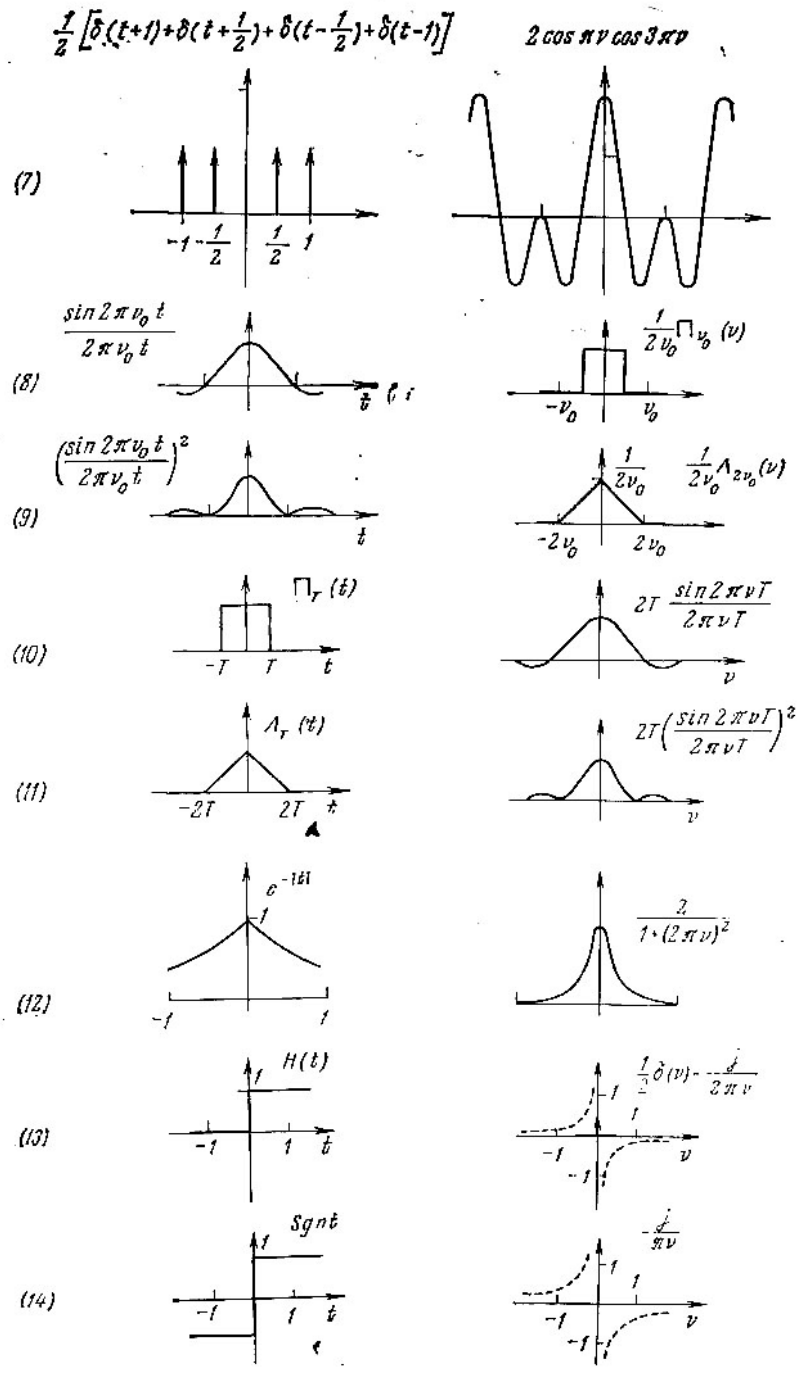
После вычисления спектров гармонического сигнала (1.20) и комплексной экспоненты (1.21) быть ясно, что спектральная функция такого сигнала представляет собой набор дельта-функций, расположенных на частотах гармоник ряда Фурье:

$$\dot{S}(\omega) = \sum_{k=-\infty}^{\infty} 2\pi \dot{C}_k \delta\left(\omega - \frac{2\pi k}{T}\right).$$

Множители при дельта-функциях равны соответствующим коэффициентам ряда Фурье $\dot{C}_k$, умноженным на 2π.

Таблица 1.1 - Преобразования Фурье основных (типовых) сигналов [49]





1.3.6 Преобразование Фурье физических функций

Функции, используемые в физике, известны лишь на ограниченном интервале (0, T). Для их доопределения на всю временную ось используют два способа. По первому способу функцию полагают равной нулю вне интервала (0, T) (рис. 1.16). Если функция действительно равна нулю вне (0, T), то в этом случае можно вычислить спектр с любой заданной точностью.

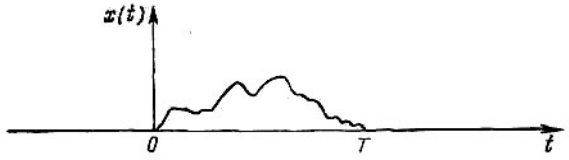


Рис. 1.16

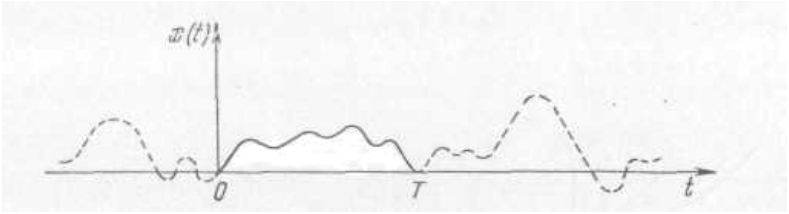


Рис. 1.17

Второй способ основан на игнорировании поведения функции вне интервала (0, *T*) (рис. 1.17). Так как функция задана только на интервале (0, *T*), то ее Фурье-образ определен только для дискретных значении частот, разделенных промежутками длиной 1/*T* или кратными 1/*T* (так же как и для периодических функций). Этот случай аналогичен случаю задания функции $x(t)$ лишь для n дискретных значений аргумента с промежутками между ними длиной T_e (частота дискретизации сигнала $F_e = = 1/T_e$).

Действительно, длина области определения функции равна $T = nT_e$. Поэтому величина разрешения по v составляет $\Delta v = 1/(nT_e)$. Обозначим через B длину интервала спектра заданной функции.

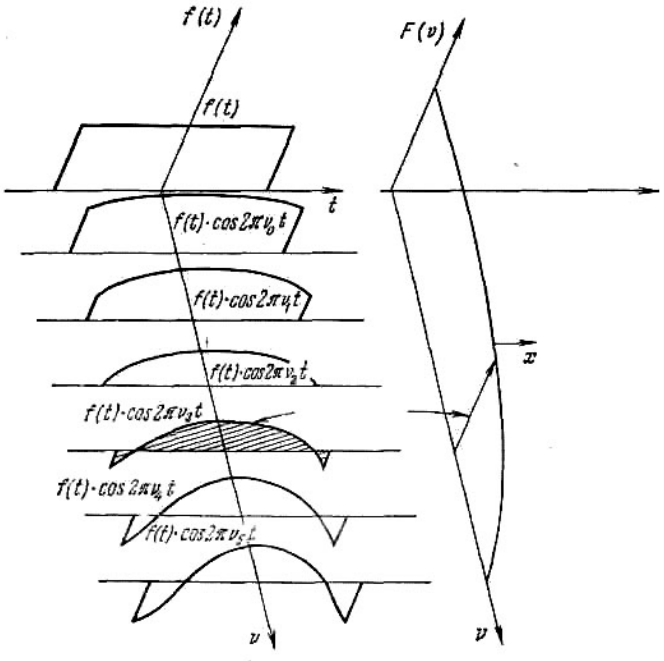


Рис. 1.18

Тогда $F_e \geq 2B$. Имеем $F_e = 1/T_e = 2\alpha B, \quad \alpha \geq 1$. Отсюда $\alpha v = 1/(nT_e) = 2\alpha B/n$.

В силу полученного максимальное число точек спектра равно $B/\Delta\nu = n/2a = n/2$, если $\alpha=1$. Итак, вся информация о функции содержится в этих $n/2$ точках спектра, совокупность которых образует Фурье-образ периодической функции с периодом T, полученной путем периодического продолжения с периодом T исходной функции вне интервала $(0,T)$.

Если вычислить $kn/2$ точек спектра (k – натуральное число), то расстояние между двумя соседними точками спектра будет в k раз меньше и, следовательно, период продолжения исходной функции будет в k раз больше. Новая периодическая функция с периодом kT совпадает с исходной функцией $x(t)$ на интервале $(0,T)$ и равна нулю между T и kT. При неограниченном увеличении k дискретный спектр этой функции стремится к непрерывному спектру функции, рассмотренному выше.

Физический смысл преобразования Фурье $\mathrm{TF}\{f(t)\}$: $f(\mathrm{t}) \rightarrow F(\nu)$ можно понять с помощью рис. 1.18.

Переходя к более принятым в математике обозначениям функций, следует отметить, что функции $X(\nu)$ и $x(t)$ описывают в различной форме один и тот же физический процесс. Если рассматривается функция $x(t)$, то состояние системы изучается на плоскости «амплитуда – время». При рассмотрении же функции $X(\nu)$ состояние системы изучается на плоскости «амплитуда – частота».

Для вычисления значения $X(\nu)$ при фиксированном значении $\nu=\nu_i$ необходимо подсчитать вклад всей функции $x(t)$, соответствующий частоте ν_i. Это означает, что производится неограниченно точная фильтрация. Такая фильтрация физически нереализуема. Следовательно, функция $X(\nu)$ не может быть известна с неограниченно точной локализацией независимой переменной ν на оси частот.

Аналогично, если восстанавливается функция $x(t)$ по известной $X(\nu)$, то необходимо знать весь спектр, в том числе и для бесконечно больших частот. Из формул прямого и обратного преобразований Фурье следует, что это также соответствует неограниченно точной фильтрации.

Итак, для вычисления точного значения $x(t)$ в фиксированный момент времени t необходимо располагать неограниченной частотной полосой. Мы сталкиваемся здесь с одной из форм ***общего принципа неопределенности*** *— познание окружающего мира возможно лишь в условиях «неточного» его описания.*

Для любой функции $x(t)$, вещественной или комплексной из $x(t) \rightleftharpoons X(\nu)$, следует $x^*(t) \rightleftharpoons X^*(-\nu)$ ($x^*(.)$ *и* $x(.)$ *– комплексно-сопряженные величины).*

При этом Фурье-образы функций $x(t)$ и $x^*(t)$ связаны друг с другом простыми отношениями:

$$x(t)\begin{Bmatrix}\text{Действительная}\\ \text{Мнимая}\\ \text{Произвольная четная}\\ \text{Произвольная нечетная}\\ \text{Произвольная}\end{Bmatrix} x^*(t) \rightleftharpoons X^*(-\nu) = \begin{Bmatrix} X(\nu)\\ -X(\nu)\\ X^*(\nu)\\ -X^*(\nu)\\ X^*(-\nu)\end{Bmatrix} \rightarrow x(-t) \rightleftarrows X(-\nu) = \begin{Bmatrix} X^*(\nu)\\ -X^*(\nu)\\ X(\nu)\\ -X(\nu)\\ X(-\nu)\end{Bmatrix} \quad (1.22)$$

1.3.7 Дискретное преобразование Фурье (ДПФ)

Алгоритмы обработки сигналов основываются на вычислении операции свертки. Эта операция включает спектральный анализ входного сигнала, умножение спектра сигнала на импульсную характеристику устройства обработки и обратное преобразование спектра выходного сигнала.

Устройства цифровой обработки данных относятся к классу линейных дискретных систем с постоянными параметрами. При обработке сигналов в таких системах выходной сигнал $y(n)$ равен *дискретной свертке* входного сигнала $x(n)$ с импульсной характеристикой системы $h(n)$, причем отсчеты сигналов следуют с периодом дискретизации $T_д$: $y(n)=x(n)\otimes h(n)$, где $\otimes$-операция дискретной свертки

Для вычисления свертки применяются прямое и обратное дискретные преобразования Фурье (ДПФ). Алгоритм вычисления свертки двух последовательностей $x(n)$ и $h(n)$ с помощью ДПФ включает:

прямое ДПФ для перехода в k-область

$$X(k)=\sum_{n=0}^{N-1} x(n)f(kn), \quad H(k)=\sum_{n=0}^{N-1} h(n)f(kn); \quad (1.23)$$

умножение на импульсную характеристику системы обработки данных в k-области $Y(k) = X(k)H(k)$;

обратное ДПФ для перехода в k-область $y(n)=\frac{1}{N}\sum_{k=0}^{N-1} Y(k)\times f(kn)^*$,

где $f(kn)$ – система линейно-независимых ортогональных функций, или базис разложения функции; $f(kn)^*$ — комплексно-сопряженная функция $f(kn)$; N – энергия базисной функции.

В качестве базиса разложения может быть принята любая полная система ортогональных функций. При реализации цифровых фильтров (ЦФ) предварительно рассчитанные значения коэффициентов *H(k)* хранятся в запоминающем устройстве (ЗУ) фильтра. Прямое и обратное ДПФ вы-

числяются с помощью одного алгоритма, требуется лишь перестановка данных и замена переменных, поэтому при их физической реализации для их вычисления используется аналогичная (для многих задач одна и та же) аппаратура.

Дискретные функции, заданные на интервале N, могут рассматриваться как векторы в N-мерном евклидовом пространстве, которые могут быть представлены в матричной форме

$$X_k = F_{kn} X_n, \quad X_n = \frac{1}{N} F^*_{kn} X_k,$$

где X_n, X_k — матрицы-столбцы сигнала и его спектра размером N; F_{kn} - унитарная матрица базисных функций $f(kn)$ размером $N \times N$; F^*_{kn} - матрица, элементы которой комплексно сопряжены с элементами матрицы F_{kn}.

Из правил умножения матриц следует, что для вычисления ДПФ необходимо выполнить $(N\text{-}1)^2$ умножений и $N(N\text{-}1)$ сложений комплексных чисел. При больших значениях N реализация такого объема вычислений в реальном масштабе времени на современных вычислительных средствах затруднительна. Поэтому в практике цифровой обработки сигналов применяется более эффективный метод – быстрое преобразование Фурье (БПФ) [6].

1.3.8 Быстрое преобразование Фурье (БПФ)

Суть метода заключается в том, что когда размер матрицы F_{kn} является составным числом, то матрица может быть факторизована, т.е. представлена в виде произведения слабо заполненных матриц, большинство элементов которых равно нулю. Это дает возможность производить вычисление ДПФ в несколько этапов, выполняя на каждом лишь небольшое количество операций. Благодаря этому достигается экономия вычислений. Если $N = r^L$, то r называют основанием преобразования, а L — числом этапов преобразования. Существуют преобразования и со смешанными основаниями.

Если матрица базисных функций F_{kn} может быть факторизована, то матрица F^*_{kn} тоже факторизуется. Свойством факторизации матриц обладают все наиболее распространенные системы базисных функций. Факторизация может проводиться различными способами, каждый из которых имеет свои достоинства и недостатки.

Таким образом, большое количество существующих систем ортогональных базисных функций, использование каждой из которых наиболее эффективно для решения конкретных задач обработки сигналов, и различные методы факторизации матриц F_{kn} приводят к многообразию алгоритмов БПФ. Проведем краткий обзор алгоритмов, основанных на проце-

дуре БПФ и получивших наибольшее распространение в цифровой фильтрации сигналов.

Алгоритм БПФ в базисе дискретных экспоненциальных функций. Как было отмечено, идея БПФ заключается в умножении вектора-столбца сигнала X_n на факторизованную матрицу базисных функций:

$$X_k = F_1 F_2 \ldots F_n X_n. \qquad (1.24)$$

где F_1, F_2,…. F_n – сомножители матрицы базисных функций.

Алгоритм вычисления БПФ включает n этапов. На 1-м вычисляется $X^n{}_k = F_n X_n$; на *втором* - $X^{n-1}{}_k = F_{n-1} X^n{}_k$; *на* n-м-$X_k = = F_1 X^2{}_k$, где $X^j{}_k$ – вектор-столбец промежуточных значений на j-м этапе вычисления ДПФ.

Дискретные экспоненциальные функции являются дискретными аналогами комплексных экспоненциальных функций для случая, когда время t изменяется дискретно ($t=nT_д$):

$$W^k = \exp[j(2\pi/N)k] = \cos[(2\pi/N)k] + j\sin[(2\pi/N)k]$$

где W^k – дискретная экспоненциальная функция или поворачивающий коэффициент.

Существуют различные методы факторизации матрицы W^k, которые во многом определяют качество алгоритма БПФ. Например, преимуществом метода, описанного Глассманом [6,7,8], является естественный порядок следования отсчетов спектра. Матрица W^k размером 8x8 (N=8) по этому методу факторизуется следующим образом:

Алгоритм БПФ включает три этапа, на каждом из которых осуществляется умножение вектора-столбца на слабозаполненные матрицы (1.25).

$$\begin{vmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^1 & W^4 & W^7 & W^2 & W^5 \\ W^0 & W^4 & W^0 & W^4 & W^0 & W^4 & W^0 & W^4 \\ W^0 & W^5 & W^2 & W^7 & W^4 & W^1 & W^6 & W^3 \\ W^0 & W^6 & W^4 & W^2 & W^0 & W^6 & W^4 & W^2 \\ W^0 & W^7 & W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{vmatrix} = \begin{vmatrix} W^0 & W^0 & & & & & & \\ & & W^0 & W^1 & & & & \\ & & & & W^0 & W^2 & & \\ & & & & & & W^0 & W^3 \\ W^0 & W^4 & & & & & & \\ & & W^0 & W^5 & & & & \\ & & & & W^0 & W^6 & & \\ & & & & & & W^0 & W^7 \end{vmatrix} \times$$

$$\times \begin{vmatrix} 1 & & 1 & & & & & \\ & 1 & & 1 & & & & \\ & & & & 1 & & W^2 & \\ & & & & & 1 & & W^2 \\ 1 & & W^4 & & & & & \\ & 1 & & W^4 & & & & \\ & & & & 1 & & W^6 & \\ & & & & & 1 & & W^6 \end{vmatrix} \begin{vmatrix} 1 & & 1 & & & & & \\ & 1 & & 1 & & & & \\ & & 1 & & 1 & & & \\ & & & 1 & & 1 & & \\ 1 & & & & W^4 & & & \\ & 1 & & & & W^4 & & \\ & & 1 & & & & W^4 & \\ & & & 1 & & & & W^4 \end{vmatrix} \qquad (1.25)$$

Большинство применяемых в цифровой обработке сигналов алгоритмов БПФ построено на алгоритмах БПФ с прореживанием по времени и по частоте. На рис.1.19 представлен граф 8-точечного БПФ по основанию 2 с прореживанием по времени. Незачерненные кружочки обозначают операции сложения — вычитания, причем верхний выход означает сумму, нижний — разность. Стрелкой обозначается операция умножения на коэффициент, указанный около нее. На графе рис. 1.19 можно выделить элементарный граф базовой операции (при r = 2 это БПФ двух отсчетов).

Базовую операцию можно представить следующим образом:

$$X = A + W^k B, \quad Y = A - W^k B,$$

где X, Y — выходные значения; A, B — входные.

Алгоритм БПФ с прореживанием по времени в основном аналогичен алгоритму БПФ с прореживанием по частоте. Отличаются эти алгоритмы выполнением базовой операции. Для алгоритма с прореживанием по частоте базовая операция имеет вид

$$X = A + B, \quad Y = (A - B)W^k.$$

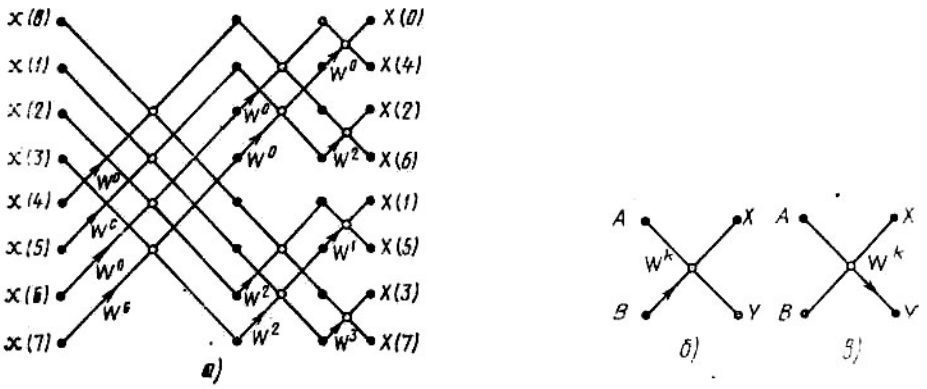


Рис. 1.19 - Граф 8-точечного БПФ (а), базовые операции алгоритма БПФ с прореживанием по времени (б) и по частоте (в)

Из приведенного на рис 1.19 графа видно, что *общее число операций при вычислении БПФ приблизительно равно (N/2) log_2N умножений и $Nlog_2N$ сложений комплексных чисел.*

Рассмотренные алгоритмы БПФ основываются на комплексном умножении отсчетов и поворачивающих коэффициентов W^k. Поэтому в состав цифровых вычислений необходимо включить умножитель комплексных чисел. Однако такой умножитель обладает большими возможностями, чем это необходимо для БПФ, так как с его помощью можно изменять и амплитуду, и фазу отсчетов. Вместе с тем для выполнения БПФ достаточно изменять фазу. Существуют алгоритмы, предназначенные только для поворота вектора. Использование их в алгоритмах БПФ упрощает аппаратную реализацию последних [9,10,6].

Кроме перечисленных существует множество различных алгоритмов БПФ, использование каждого из которых дает то или иное преимущество при решении конкретных задач. Например, широко распространенным является *алгоритм быстрого преобразования Уолша — Пэли* [11]. Основным преимуществом такого алгоритма БПФ, построенного в базисе разложения функций Уолша, упорядоченных различными методами (Пэли, Адамара и т. п.), является отсутствие операций умножения при вычислении ДПФ. Это позволяет построить более эффективные с вычислительной точки зрения алгоритмы БПФ.

1.4 Преобразование Лапласа и его применение в системах обработки данных

Из предыдущего материала известно, что Фурье-образ функции $f(t)$ задается формулой

$$F(v) = \int_{-\infty}^{\infty} f(t)e^{-2\pi jvt}dt. \qquad (1.25)$$

и $F(v)$ существует только в случае сходимости интеграла в правой части равенства (1.25). Если же интеграл расходится, то его можно сделать сходящимся, заменив показатель экспоненты ($-2\pi j\,v$) комплексным числом

$$p = -\delta_0 - 2\pi jv, \qquad (1.26)$$

где $\delta_0>0$. Показатель p называется комплексной частотой или, как принято в операционном исчислении, в теории автоматического управления, - оператором Лапласа. Если же функция $f(t)$ равна нулю при $t<0$, то получаем преобразование Лапласа *(Transform Laplace)*, определяемое формулой

$$TL[f(t)] = \int_0^{\infty} f(t)e^{-pt}dt. \qquad (1.27)$$

Число $\delta_0>0$ можно выбрать сколь угодно большим для того, чтобы интеграл в правой части равенства (1.27) всегда сходится. Это число $\delta_0>0$ называется показателем сходимости.

Замечания:

1. Такое определение показателя сходимости не совсем корректно. Показателем сходимости функции f(t) называется наименьшее число $\delta_0>0$*, для которого* $|f(t)|< Me^{-\delta_0 t}$*, t>0. Вместо названия показатель сходимости используют также характеристический показатель, или показатель Ляпунова.*

2. В этом разделе, в отличие от предыдущих, сигнал задается функцией f(t), а частота обозначается литерой ν.

Преобразование Лапласа представляет интерес для изучения переходных режимов, поскольку последние равны нулю для $t<0$ ($t=0$ соответствует моменту времени, при котором начинается возмущение, порождающее переходный процесс) в силу принципа причинности: следствие не может предшествовать причине.

Преобразование Лапласа – рабочий инструмент физиков и специалистов в области автоматического управления, систем связи и информационных технологий, так как изображение Лапласа импульсного отклика линейной системы обработки данных представляет собой передаточную функцию такой системы. С математической стороны вопроса важнейшей причиной такого широкого использования преобразования Лапласа является его применимость для тех функций (сигналов), у которых Фурье-образ не существует из-за расходимости интеграла (1.25).

После осуществления преобразования Лапласа над сверткой (см. подраздел 1.3.5)

$$s(t) = e(t) * h(t) \qquad (1.28)$$

получим выражение для изображения Лапласа

$$W_s(p) = W_e(p)W_h(p). \qquad (1.29)$$

Существуют таблицы изображений Лапласа (табл. 1.2), позволяющие по заданной функции $f(t)$ (равной нулю для $t<0$) находить ее изображение Лапласа, и наоборот [12]. В случае гармонического режима в изображении Лапласа можно заменить p на комплексную частоту $j\omega$ или $2\pi j\nu$.

1.4.1 Связь между Фурье - образом и изображением Лапласа

Рассмотрим функцию $f(t)$, для которой существует Фурье-образ

$$F(v) = \int_{-\infty}^{\infty} f(t)e^{-2\pi jvt}dt. \tag{1.30}$$

Имеем

$$F(v) = \int_{-\infty}^{0} f(t)e^{-2\pi jvt}dt + \int_{0}^{\infty} f(t)e^{-2\pi jvt}dt, \tag{1.31}$$

или

$$F(v) = \int_{0}^{\infty} f(-t)e^{-2\pi jv(-t)}dt + \int_{0}^{\infty} f(t)e^{-2\pi jvt}dt. \tag{1.32}$$

Определим функции $f_+(t)$ и f-(t) равенствами (рис. 1.20)

$$f_+(t) = \begin{cases} f(t), t \geq 0, \\ 0, \quad t < 0, \end{cases} \tag{1.33}$$

$$f_-(t) = \begin{cases} f(t), t \leq 0, \\ 0, \quad t > 0. \end{cases} \tag{1.34}$$

Будем предполагать, что для функций $f_+(t)$ и $f_-(t)$ существуют Фурье-образы, т.е. сходятся интегралы

$$\int_{-\infty}^{0} f(t)dt \text{ и } \int_{0}^{\infty} f(t)dt\ .$$

Полагая в интеграле

$$\int_{0}^{\infty} f_+(t)e^{-2\pi jvt}dt, \quad p = 2\pi jv$$

(поскольку показатель сходимости δ_0=0), переходим от Фурье-образа к изображению Лапласа

$$TF[f_+(t)]_{2\pi jv=p} = TL[f_+(t)] = W_1'(p), \quad t > 0. \tag{1.35}$$

Аналогично осуществляется переход для функции $f\_$ (t):

$$TF[f_-(t)]_{2\pi jv=p} = TL[f_-(-t)] = W_2'(p), \quad t < 0. \tag{1.36}$$

Имеем

$$\int_{0}^{\infty} f_-(-t)e^{-2\pi jvt}dt\,|_{p=2\pi jv} = W_2(-p). \tag{1.37}$$

Отсюда получаем

$$F(v) = W_1(2\pi jv) + W_2(-2\pi jv), \tag{1.38}$$

где $W_1(p) = TL[f_+(t)], W_2(p) = TL[f_-(-t)].$

Из предположения о сходимости интегралов $\int_0^{\infty} f(t)e^{-2\pi jvt}dt$ и $\int_{-\infty}^{0} f(t)e^{-2\pi jvt}dt$ следует, что функции $W_1(p)$ и $W_2(p)$ не имеют полюсов на мнимой оси и правее ее. Если $W_1(p)$и $W_2(p)$ имеют один или несколько полюсов (корней) на мнимой оси, то

$$F(v) = W_1(2\pi jv) + \frac{1}{2}\sum_K A_K\delta(v - v_K) + W_2(-2\pi jv) + \\ + \frac{1}{2}\sum_n A_n\delta(v - v_n). \tag{1.39}$$

Здесь A_K – вычет функции W_1 $(2\pi jv)$ в полюсе jv_K и

A_n—вычет функции W_2 $(2\pi jv)$ в полюсе jv_n. (1.40)

Если $W_1(p)$ или $W_2(p)$ имеют один или несколько полюсов правее мнимой оси, то Фурье-образа не существует.

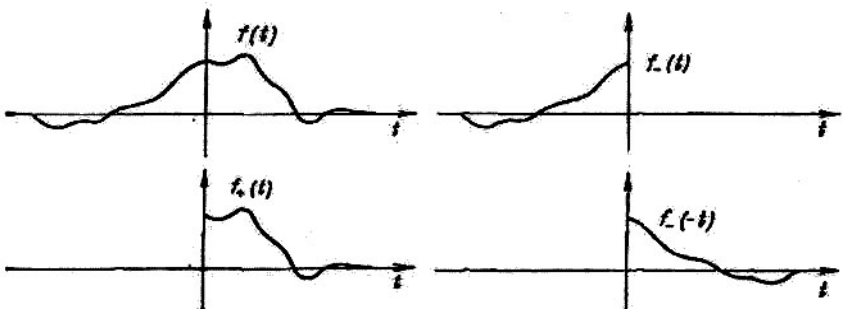


Рис. 1.20

Функции, для которых изображения Лапласа имеют полюсы на мнимой оси и не имеют их в правой полуплоскости, являются незатухающими (в качестве примера можно привести колебательную систему). Если же изображения Лапласа имеют полюсы с положительной вещественной частью, то соответствующие им оригиналы являются неустойчивыми функциями. Правильнее сказать неограниченными функциями на полуоси $t > 0$ (например, переходный процесс неустойчивой системы). Отметим, что на практике последние два случая встречаются редко. Поэтому для большинства функций $f(t)$ можно предполагать, что изображения Лапласа от $f_+(t)$ и $f_-(t)$ не имеют полюсов в правой полуплоскости и на мнимой оси. Тем самым предполагается справедливым равенство

$$F(v) = W_1(2\pi jv) + W_2(-2\pi jv). \tag{1.41}$$

Рассмотрим три важных частных случая:

1. $f(t) \equiv 0$ для $t < 0$. (1.42)

Этот случай соответствует δ-отклику физически реализуемых систем.

Имеем:

$f_-(-t) \equiv 0$ для $t > 0$. (1.43)

и поэтому:

$F(v) = W_1(2\pi jv)$. (1.44)

2. Функция $f(t)$ — четная. Имеем:

$$f_+(t) = f_-(-t). \quad (1.45)$$

Отсюда получаем

$$F(v) = W(2\pi jv) + W(-2\pi jv). \quad (1.46)$$

3. Функция $f(t)$ —нечетная. Имеем

$$f_+(t) = -f_-(-t). \quad (1.47)$$

Поэтому

$$F(v) = W(2\pi jv) - W(-2\pi jv). \quad (1.48)$$

Таблица 1.2 - Изображения функций по Лапласу и Z-преобразований [1]

№	$x(t)$ при $t \geq 0$	$X(p)$	Z-преобразование
1	$1_0(t)$	$\frac{1}{p}$	$\frac{z}{z-1}$
2	T	$\frac{1}{p^2}$	$\frac{Tz}{(z-1)^2}$
3	$e^{-\alpha t}$	$\frac{1}{p+\alpha}$	$\frac{z}{z-e^{-\alpha T}}$
4	$te^{-\alpha t}$	$\frac{1}{(p+\alpha)^2}$	$T\frac{e^{-\alpha T}z}{(z-e^{-\alpha T})^2}$

[1] *Holbrook G. Laplace transform for electronic engineers. - Pergamon Press, 1959.*

5	$\frac{1-e^{-\alpha t}}{\alpha}$	$\frac{1}{p(p+\alpha)}$	$\frac{(1-e^{-\alpha T})z}{\alpha(z-1)(z-e^{-\alpha T})}$
6	$\sin \beta t$	$\frac{\beta}{p^2+\beta^2}$	$\frac{z \sin \beta T}{z^2-2z\cos \beta T+1}$
7	$\cos \beta t$	$\frac{p}{p^2+\beta^2}$	$\frac{z(z-\cos \beta T)}{z^2-2z\cos \beta T+1}$
8	$e^{-\alpha t}\sin \beta t$	$\frac{\beta}{(p+\alpha)^2+\beta^2}$	$\frac{e^{-\alpha T}z \sin \beta T}{z^2-2ze^{-\alpha T}\cos \beta T+e^{-2\alpha T}}$
9	$e^{-\alpha t}\cos \beta t$	$\frac{p+\alpha}{(p+\alpha)^2+\beta^2}$	$\frac{z(z-e^{-\alpha T}\cos \beta T)}{z^2-2ze^{-\alpha T}\cos \beta T+e^{-2\alpha T}}$
10	$\delta(t)=\begin{cases}1, & t=0\\ 0, & t=kT, k=0\end{cases}$	1	1
11	$\delta(t-kT)=\begin{cases}1, & t=kT\\ 0, & t\neq kT, k=1,2,\ldots\end{cases}$	e^{-kTs}	z^{-k}

1.4.2 Применение преобразования Лапласа в системах обработки данных

Модели систем обработки данных

Рассмотрим основные виды моделей линейных непрерывных стационарных динамических систем обработки данных (СОД), к которым в общем виде относятся различного вида устройства преобразования сигналов, в частности: фильтры, наблюдающие и распознающие устройства и т.п. [1].

1. Дифференциальные модели. Наиболее универсальная модель, несущая в себе физику происходящих в системе явлений, описывается обыкновенным дифференциальным уравнением следующего вида:

$$\sum_{i=0}^{na} a_i y^{(i)}(t) = \sum_{j=0}^{nb} b_j u^{(j)}(t), \qquad (1.49)$$

где na – порядок модели ($na > nb$), a_i и b_j – постоянные коэффициенты (параметры модели), $u^{(i)}(t)$ и $y^{(j)}(t)$ – производные, соответственно, входного и выходного сигналов.

2. **Модели, характеризуемые передаточными функциями.** Данная характеристика определяется как отношение преобразований Лапласа

выходного и входного сигналов, что с учетом свойств данного преобразования и вышеприведенной формулы дает

$$W(p)=\frac{L\{y(t)\}}{L\{u(t)\}}=\frac{Y(p)}{U(p)}=\frac{\sum_{j=0}^{nb} b_j p^j}{\sum_{i=0}^{na} a_i p^i}, \qquad (1.50)$$

где ***L{·}*** — символ преобразования Лапласа (***L{.}=TL{. },*** *см.раздел 1.4*), $p = j\omega$ – комплексная частота (оператор Лапласа).

Следует иметь в виду, что класс аналоговых фильтров, т.е. фильтров для непрерывных сигналов, имеет передаточную функцию только типа (1.50). При этом любая передаточная функция типа (1.50) может быть представлена в виде комбинации следующих четырех элементарных передаточных функций:

$W_1(p)=\frac{1}{1+Tp}$ – низкочастотный фильтр 1-го рода;

$W_2(p)=\frac{Tp}{1+Tp}$ – высокочастотный фильтр 1-го рода;

$W_3(p)=\frac{1}{1+2\xi Tp+T^2p^2}$ –низкочастотный фильтр 2-го рода;

$W_3(p)=\frac{T^2p^2}{1+2\xi Tp+T^2p^2}$ –высокочастотный фильтр 2-го рода.

3. Модели, характеризуемые *ИХ w(t)*. Под *ИХ* - импульсной характеристикой *w(t) или, что тоже, импульсно - переходной характеристикой (ИПХ))* понимается реакция предварительно невозмущенного объекта (т. е. объекта с нулевыми начальными условиями) на входной сигнал/импульс в виде δ - функции.

4. Модели, характеризуемые переходными функциями *h(t).* Это реакция предварительно невозмущенного объекта на входной сигнал в виде единичного скачка. Из теории управления известны следующие соотношения между этими характеристиками:

$$L\{w(t)\}=W(p), \quad w'(t)=h(t), \quad L\{h(t)\}=\frac{W(p)}{p}. \qquad (1.51)$$

При нулевых начальных условиях связь между выходным и входным сигналами описывается интегралом свертки (интеграл Дюамеля):

$y(t)=\int_{-\infty}^{\infty} w(t-\tau)u(\tau)d\tau$, или, в операторной форме: $Y(p)=W(p)\cdot U(p)$.

5. Модели, характеризуемые частотными характеристиками.

Частотные характеристики объекта определяются его комплексным коэффициентом передачи $\dot{K}(j\omega)=W(j\omega) = W(p)|_{p=j\omega}$, который является Фурье-преобразованием ИХ и еще известен как амплитудно-фазочастотная характеристика (АФЧХ) СОД.

Модуль комплексного коэффициента передачи $|W(j\omega)| = A(\omega)$ представляет собой, как известно, амплитудно-частотную характеристику

(АЧХ) объекта с передаточной функцией $W(p)$, а аргумент $\arg(W(j\omega)) = \varphi(\omega)$ – фазочастотную характеристику (ФЧХ).

Графическое представление АФЧХ $W(j\omega)$ на комплексной плоскости при изменении частоты ω от 0 до ∞, то есть график АФЧХ в полярных координатах, в отечественной литературе, как известно, именуется *годографом,* а в англоязычной — *диаграммой Найквистпа.*

В то же время следует отметить, что часто для удобства в расчетах используется логарифмическая амплитудно-частотная характеристика (ЛАЧХ), равная $20 \lg |W(j\omega)|$.

6. Модель в пространстве состояний. Динамику системы (1.49) можно описать двумя уравнениями в пространстве состояний [1,12,13].

$$\begin{aligned} X'(t) &= AX(t) + Bu(t), \\ y(t) &= CX(t) + Du(t), \end{aligned} \qquad (1.52)$$

где $\boldsymbol{X}(t)$– вектор-столбец переменных состояния; $\boldsymbol{y}(t) = [y_1(t), y_2(t),....,y_p(t)]^T$ – вектор выходных координат; ***A, B, C и D***–матрицы соответствующих размерностей, в частности, при скалярных $u(t)$ *матрица координат входных воздействий* ***B*** будет иметь размерность *(1* x *n),* а *матрица наблюдений* ***C*** в общем случае будет иметь размерность *(p* x *n).*

Применение, при нулевых начальных условиях, к последним уравнениям преобразования Лапласа позволяет получить следующее матричное выражение для передаточной функции:

$$W(p) = C(pI - A)^{-1}B + D,$$

где I — единичная матрица, соответствующей размерности.

Отметим также, что *все приведенные модели являются эквивалентными друг другу, то есть, зная любую из них, можно при необходимости получить все остальные.*

Дискретные модели СОД. Z - преобразование

Для систем, функционирование которых по тем или иным причинам протекает в дискретном времени $t_k = kT$ (в данном случае T – интервал (шаг) дискретизации), наиболее общим видом описания является разностное уравнение (конечно-разностный аналог дифференциального уравнения (1.49))

$$y_k + a_1 y_{k-1} + \ldots + a_{na} y_{k-na} = b_1 u_k + b_2 u_{k-1} + b_3 u_{k-2} + \ldots + b_{nb} u_{k-nb-1},$$

где

$$y_{k-i} = y[(k-i)T], u_{k-j} = u[(k-j)T]. \quad y_k = \sum_{i=1}^{n} a_i y_{k-i} + \sum_{j=0}^{m} b_{j+1} u_{k-j}\,.$$

Связь между сигналами может быть отражена также через дискретную свертку (см. раздел 1.3.7):

$$y_k = \sum_{i=0}^{k} w_i u_{k-i},$$

где w_i – ординаты весовой решетчатой функции объекта, или, с использованием аппарата Z-преобразования [1,2]

$$Y(z)=\sum_{k=0}^{\infty} y_k z^{-k},\ \text{где}\ z=e^{pT}, \tag{1.53}$$

через дискретную передаточную функцию

$$W(z)=\frac{Y(z)}{u(z)}=\frac{B(z)}{A(z)}, \tag{1.54}$$

которая определяется на основании разностного уравнения после применения к обеим частям этого уравнения Z-преобразования (см.табл. 1.3) и, представляет, собственно, *передаточную функцию дискретного*, а в случае бинарного квантования значений сигналов, *передаточную функцию цифрового фильтра*:

$$(1+a_1z^{-1}+a_2z^{-2}+\ldots+a_{na}z^{-na})Y(z)=$$
$$=(b_1+b_2z^{-1}+b_3z^{-2}+\ldots+b_{nb}z^{-nb+1})U(z).$$

Заметим, *что Z-изображением решетчатой импульсной переходной характеристики является W(z), то есть* $Z\{w_i\}=W(z)$.

С другой стороны дискретные модели систем имеют место и в силу того, что на практике в большинстве случаев измерение непрерывных сигналов производится в дискретные моменты времени. В этой связи предложены и применяются следующие способы перехода от непрерывных моделей систем к их дискретным аналогам.

1. С применением Z-преобразования со следующей цепочкой переходов:

$$W(p)\to L^{-1}\{W(p)\}=w(t)\to w(kT)=w_k\to W(z)=Z\{w_k\}.$$

2. С заменой конечными разностями производных в дифференциальном уравнении вида (1.49), описывающем динамику непрерывных СОД:

$$py=\frac{dy(t)}{dt}\approx\frac{y_k-y_{k-1}}{T};\quad \frac{d^2y(t)}{dt^2}\approx\frac{y_k-2y_{k-1}+y_{k-2}}{T}\ \text{и т.д.}$$

Замечание. Данный подход дает приемлемую точность только при достаточно малых значениях T, удовлетворяющих теореме восстановления (теорема К. Шеннона) (эта теорема в отечественной литературе носит название *теоремы В.А. Котельникова* [4], а в западной – теоремы отсчетов Найквиста [3]):

Если для частоты дискретизации $F_e=1/T$ *справедливо неравенство* $F_e\ge 2F_M$, *где* F_M *–наибольшая частота спектра функции* $x(t)$, *то функция* $x(t)$ *однозначно восстанавливается по дискретным значениям* $x(k/F_e)$, $k=0,\pm1,\pm2,\ldots$

Имеет место *формула для выбора частоты дискретизации по теореме Котельникова*, применяемая на практике:

$F_e\ge F\left(2,2/\sqrt{\varepsilon}\right)$, где ε - ошибка восстановления.

3. С заменой $p=2(z-1)/(z+1)/T$ (приближенный способ, предложенный А. Тастиным и называемый *билинейным преобразованием),* то есть

$$W(p)\big|_{p=\frac{2(z-1)}{T(z+1)}}\to W(z).$$

Для дискретных объектов также может быть использовано описание в пространстве состояний

$$X_k = AX_{k-1} + Bu_{k-1},$$
$$y_k = CX_k + Du_k, \tag{1.55}$$

через переходную функцию и частотные характеристики, – так же, как и для непрерывных систем.

Определим множитель $z^{-1} = e^{-pT}$ как *оператор задержки,* то есть $z^{-1}u_k = u_{k-1}$, $z^{-2}u_k = u_{k-2}$ и т. д. Тогда, обозначая моменты дискретного времени тем же символом t, что и непрерывное время (т.е. в данном случае $t = 0, 1, 2, ...$), приведем несколько распространенных моделей дискретных систем, учитывающих действие шума наблюдения.

1. Модель авторегрессии AR (*AutoRegressive*) — считается самым простым описанием:

$$A(z)y(t) = e(t), \tag{1.56}$$

где $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{na} z^{-na}$.

2. ARX-модель (*AutoRegressive with external input*) — более сложная:

$$A(z)y(t) = B(z)u(t) + e(t), \tag{1.57}$$

или, в развернутом виде,

$$y(t) + a_1 y(t-1) + \ldots + a_{na} y(t-n) =$$
$$= b_1 u(t) + b_2 u(t-1) + \ldots + b_{nb} u(t-m) + e(t).$$

Здесь и ниже $e(t)$ — дискретный белый шум,

$$B(z) = b_1 + b_2 z^{-1} + \ldots + b_{nb} z^{-nb+1}.$$

3. ARMAX-модель (*AutoRegressive-Moving Average with eXternal input* — модель авторегрессии скользящего среднего, в русскоязычных изданиях - АРСС):

$$A(z)y(t) = B(z)u(t-nk) + C(z)e(t), \tag{1.58}$$

где nk — величина задержки (динамического запаздывания),

$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{nc} z^{-nc}.$$

4. Модель «вход-выход» (в англоязычных источниках такая модель называется «Output-Error», то есть «выход-ошибка», сокращенно OR):

$$y(t) = \frac{B(z)}{F(z)} u(t-nk) + e(t), \tag{1.59}$$

где $F(z) = 1 + f_1 z^{-1} + f_2 z^{-2} + \ldots + f_{nf} z^{-nf}$.

5. Модель Бокса—Дженкинса (BJ) [8]:

$$y(t) = \frac{B(z)}{F(z)} u(t-nk) + \frac{C(z)}{D(z)} e(t), \tag{1.60}$$

где полиномы $B(z)$, $F(z)$, $C(z)$ определены ранее, а полином

$$D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + \ldots + d_{nd} z^{-nd}.$$

Данные модели можно рассматривать как частные случаи обобщенной параметрической линейной структуры

$$A(z)y(t) = \frac{B(z)}{F(z)} u(t-nk) + \frac{C(z)}{D(z)} e(t), \tag{1.61}$$

при этом все они допускают расширение для многомерных объектов, имеющих несколько входов и выходов.

3. Модель в пространстве состояний с дискретным временем (*State space with discrete time*):

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t) + v(t), \end{aligned} \tag{1.62}$$

где ***A, B, C, D*** – матрицы соответствующих размерностей, $v(t)$ – коррелированный шум наблюдений.

Возможна и другая (так называемая обновленная, или каноническая) форма представления данной модели:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + Ke(t), \\ y(t) &= Cx(t) + Du(t) + e(t), \end{aligned} \tag{1.63}$$

где ***K*** – некоторая матрица (вектор-столбец), $e(t)$ – дискретный белый шум (скаляр).

В качестве примеров применения систем обработки дискретных данных приведем дискретные фильтры с конечно- и бесконечно импульсными характеристиками, так называемые КИХ (нерекурсивные) и БИХ (рекурсивные) дисретные фильтры.

Рекурсивные и нерекурсивные дискретные фильтры

Рассмотрим структурные схемы устройств, реализующих уравнение

$$y_k = \sum_{j=1}^{n} a_j y_{k-j} + \sum_{i=0}^{m} b_i x_{k-i} \ , \tag{1.64}$$

в общем виде описывающее процесс обработки сигнала дискретной системой (см. также уравнение (1.49)).

Нерекурсивные фильтры

Прежде всего, следует отметить, что в общем случае при вычислении очередного выходного отсчета $y(k)$ используется информация двух типов: некоторое количество отсчетов *входного* сигнала и некоторое количество предыдущих отсчетов *выходного* сигнала. Ясно, что хотя бы один отсчет входного сигнала должен участвовать в вычислениях; в противном случае выходной сигнал не будет зависеть от входного. В противоположность этому, предыдущие отсчеты выходного сигнала могут и не использоваться. Уравнение фильтрации (1.64) в этом случае приобретает следующий вид:

$$y(k) = \sum_{i=0}^{m} b_i x(k-i). \tag{1.65}$$

Количество используемых предыдущих отсчетов m называется *порядком фильтра.*

Структурная схема, реализующая алгоритм (1.65), приведена на рис. 1.23. Некоторое количество предыдущих отсчетов и входного сигнала хранится в ячейках памяти, которые образуют дискретную линию задержки. Эти отсчеты умножаются, на коэффициенты b_i и суммируются, формируя выходной отсчет $y(k)$.

Замечание. *Согласно свойствам Z-преобразования, задержка дискретной последовательности на один такт соответствует умножению ее Z-преобразования на Z^{-1}. Поэтому элементы памяти, осуществляющие такую задержку, обозначены на структурной схеме как «Z^{-1}».*

Так как при вычислениях не используются предыдущие отсчеты выходного сигнала, в схеме отсутствуют обратные связи. Поэтому такие фильтры называются *нерекурсивными* (nonrecursive). Применяется также термин «трансверсальный фильтр» (от английского *transversal* — поперечный).

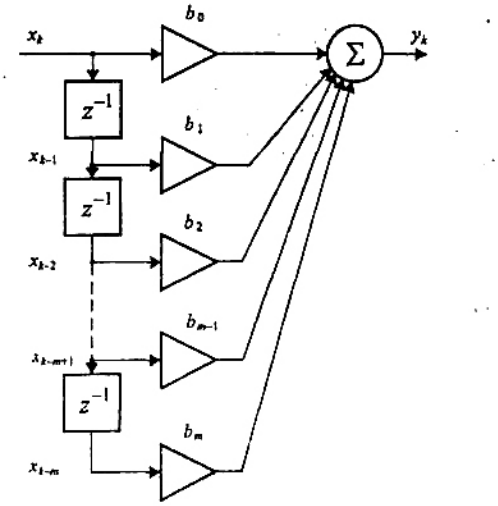


Рис. 1.23 - Нерекурсивный фильтр

Импульсная характеристика нерекурсивного фильтра определяется следующим образом. Подставим в уравнение (1.65) единичный импульс $x_0(k)$ в качества входного сигнала:

$$h(k) = \sum_{i=0}^{m} b_i x_0(k-i).$$

При этом отсчет $x_0(k - i)$ равен нулю для всех k, кроме $k = i$, когда этот отсчет равен единице. Отсюда имеем очень простой результат: $h(k) = b_k$, т. е. коэффициенты b_i являются отсчетами импульсной характеристики фильтра. Это можно наглядно пояснить с помощью рис. 1.21. При подаче на вход единичного импульса он будет перемещаться по линии задержки, умножаться на коэффициенты b_0, b_1, b_2, ... и проходить на выход устройства (ведь все остальные входные сигналы сумматора при этом равны нулю). Очевидно, что *в реальном устройстве линия задержки содержит конечное число элементов, поэтому импульсная характеристика нерекурсивного фильтра также является конечной* по длительности. Это обусловило еще одно название таких фильтров — фильтры с *конечной импульсной характеристикой* (КИХ-фильтры; английский термин — finite impulse response, FIR).

Замечание. *Вследствие отсутствия обратных связей любой нерекурсивный фильтр является устойчивым – ведь каковы бы ни были начальные условия (то есть отсчеты, хранящиеся в линии задержки), при отсутствии сигнала на входе (x(к) = 0) выходной сигнал (свободные колебания) будет отличен от нуля в течение не более чем m тактов, необходимых для очистки линии задержки.*

Продолжив рассмотрение прохождения входного единичного импульса во входной линии задержки и заполнения выходными отсчетами выходной линии задержки, для второго отсчета можно получить

$$h(2) = b_2 + a_2 h(0) + a_1 h(1) = b_2 + b_0 a_2 + a_1(b_1 + b_0 a_1) = b_2 + b_1 a_1 + b_0 a_2 + b_0 a_1^2.$$

Очевидно, что по мере того, как выходная линия задержки заполняется отсчетами импульсной характеристики, сложность аналитических формул быстро возрастает.

Несмотря на это простота анализа и реализации, а также наглядная связь коэффициентов фильтра с отсчетами его импульсной характеристики и абсолютная устойчивость привели к тому, что нерекурсивные фильтры широко применяются на практике. Однако для получения хороших частотных характеристик (например, полосовых фильтров с высокой прямоугольностью АЧХ) необходимы нерекурсивные фильтры высокого порядка — до нескольких сотен и даже тысяч, что увеличивает вычислительную сложность их программных реализаций и понижает быстродействие процесса фильтрации в целом.

Рекурсивные фильтры

Уравнение фильтрации имеет общий вид (1.64), в котором содержатся как входные, так и выходные отсчеты, поэтому для схемной реализации такого фильтра необходимо в схему представленную на рис. 1.21 добавить вторую линию задержки для хранения выходных отсчетов $y(k - i)$. Получающаяся при этом структура показана на рис. 1.24

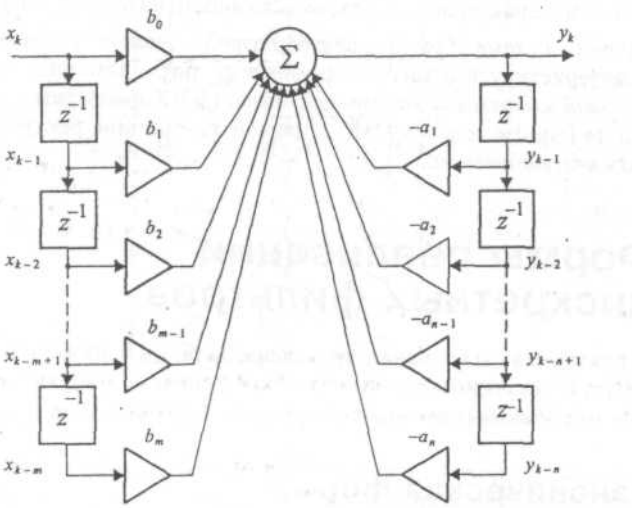


Рис. 1.24 - Рекурсивный фильтр — прямая реализация

Так как при вычислениях используются предыдущие отсчеты выходного сигнала, в схеме присутствуют обратные связи. В этой связи такие фильтры и называют *рекурсивными* (recursive).

Наличие в схеме обратных связей позволяет получить бесконечную импульсную характеристику, поэтому рекурсивные фильтры называют также фильтрами с *бесконечной импульсной характеристикой* (БИХ-фильтрами; английский термин — infinite impulse response, IIR). При некорректном выборе коэффициентов a_j рекурсивные фильтры могут быть *неустойчивыми.*

Замечание. *Количество предыдущих входных и выходных отсчетов, используемых для вычислений, может не совпадать. В таком случае порядком фильтра считается максимальное из чисел m и n.*

Импульсная характеристика (ИХ) рекурсивного фильтра рассчитывается значительно сложнее, чем для нерекурсивного. Рассмотрим формирование лишь нескольких первых ее отсчетов. При поступлении на вход единичного импульса он умножается на b_0 и проходит на выход, т.е. $h(0) = b_0$.

Далее входной единичный импульс попадает во входную линию задержки, а выходной отсчет, равный b_0, – в выходную линию задержки. В результате второй отсчет импульсной характеристики будет формироваться как

$$h(1) = b_1 + a_1 h(0) = b_1 + b_0 a_1.$$

Аналогично можно показать k – отсчет ИХ. При этом дисперсия выходного шума составит

$$D_y = D_x B_h(0) = D_x \sum_{k=0}^{\infty} h^2(k).$$

Таким образом, при воздействии на вход системы дискретного белого шума дисперсия выходного сигнала пропорциональна сумме квадратов отсчетов импульсной характеристики системы.

1.5 Вейвлет-преобразование сигналов

В последнее время наметилась тенденция к использованию широкополосных импульсных и цифровых сигналов (локация прямоугольными импульсами, видеосредства компьютеров и т. д.). Общепринятым подходом к анализу таких сигналов $s(t)$ является их представление в виде взвешенной суммы простых составляющих — базисных функций $\psi_k(t)$, помноженных на коэффициенты C_k:

$$s(t) = \sum_k C_k \psi_k(t). \qquad (1.66)$$

Так как базисные функции $\psi_k(t)$ зафиксированы как функции определенного типа, только коэффициенты C_k содержат информацию о конкретном сигнале. Таким образом, можно говорить о возможности представления произвольных сигналов на основе рядов с различными базисными функциями.

Ряд Фурье (1.66) использует в качестве базисных функций синусоиды. Они предельно локализованы в частотной области (вырождаясь на спектрограмме в вертикальную линию), но очень плохо локализованы (точнее, вообще не локализованы) во временно́й области. Противоположный пример – импульсная базисная функция

$$\psi_k(t) = \delta_k(t) = \begin{cases} 1, k = t, \\ 0, k \neq t. \end{cases} \qquad (1.67)$$

Она четко локализована во временной области и потому идеально подходит для представления разрывов сигнала. Однако эта базисная функция не несет информации о частоте сигнала и потому плохо приспособлена для представления сигналов на заданном отрезке времени и тем более периодических сигналов.

Термин *«вейвлет»,* введенный впервые Морлетом (J. Morlet), в переводе с английского wavelet означает «короткая волна». Изначально его переводили как «всплеск», «выброс» и т. д., что менее удачно. Вейвлеты занимают промежуточное положение между рассмотренными нами крайними случаями (синусоидой и импульсной функцией) и образуют некоторый набор функций, удовлетворяющих сформулированным далее условиям и основанных на использовании представления сигнала в виде (1.67).

Следует сразу отметить, что пока нет исчерпывающе полных и точных теоретических критериев, по которым те или иные базовые функции можно однозначно отнести к вейвлетам. Нередко такое отнесение оказывается попросту данью моде. В этой связи здесь и далее намеренно опущены детали определения условий существования вейвлет-функций.

Базисными функциями вейвлетов могут быть различные функции, в том числе напоминающие модулированные импульсами синусоиды, функции со скачками уровня и т. д. Это обеспечивает легкое представление сигналов с локальными скачками и разрывами наборами вейвлетов того или иного типа. Почти все вейвлеты не имеют аналитического представления в виде одной формулы и могут задаваться итерационными выражениями.

Вейвлеты характеризуются своим временным и частотным образами – рис. 1.26. Временно́й образ определяется некоторой *psi-функцией* времени $\psi(t)$. Частотный образ при этом задается ее *Фурье-образом* $\hat{\psi}(\omega)$, который задает огибающую спектра вейвлета. Если вейвлет в пространстве сужается, его «средняя частота» повышается, спектр вейвлета перемещается в область более высоких частот и расширяется. Этот процесс можно считать линейным – если вейвлет сужается вдвое, то его средняя частота и ширина спектра возрастают также вдвое.

Даже интуитивно ясно, что совокупность волновых пакетов, напоминающих модулированную импульсами синусоиду или подобных приведенному на рис1.25 вейвлету, способна хорошо отражать *локальные изменения* сигналов – рис. 1.26. Однако вопрос о представлении произвольного сигнала в произвольно заданном промежутке времени пока остается открытым. Он будет решен с введением понятия кратномасштабного анализа.

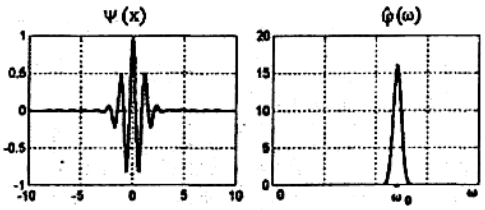


Рис. 1.25 - Временно́й и частотный образы вейвлета

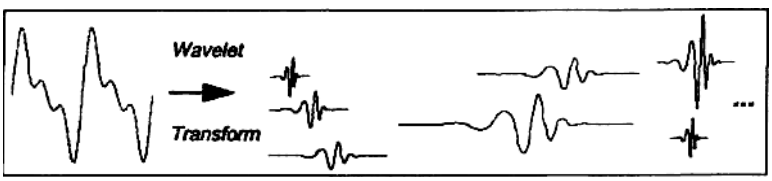


Рис. 1.26 - Иллюстрация к вейвлет-синтезу сигнала

Итак, с помощью вейвлетов сигнал представляется совокупностью волновых пакетов – вейвлетов, образованных на основе некоторой исходной (базисной, образующей и т. д.) функции $\psi_0(t)$. Эта совокупность разная в разных частях временно́го интервала определения сигнала и представляет последний с той или иной степенью детальности (см. рис. 1.24). Такой подход называют *вейвлет-анализом сигналов.*

Число используемых при разложении сигнала вейвлетов *задает уровень декомпозиции* сигнала. При этом за нулевой уровень декомпозиции принимается сам сигнал, а уровни декомпозиции образуют ниспадающее *вейвлет-дерево* того или иного вида. Точность представления сигнала по мере перехода на более низкие уровни декомпозиции снижается, но зато появляется возможность вейвлет-фильтрации сигналов, удаления из сигналов шумов и эффективной компрессии сигналов.

Замечание. *Вейвлет-составляющие сигнала даже внешне не имеют ничего общего с синусоидами, и они представлены сигналами подчас весьма сложного и порою не вполне понятного вида. Это, кстати, существенный недостаток вейвлетов с позиции наглядного их понимания и представления. Он ликвидируется соответствующими инструментальными средствами, вошедшими в пакет расширения Wavelet Toolbox системы MATLAB.*

Вполне очевидно, что для представления сигналов как в локальных областях их возмущений, так и во всем временно́м интервале изменения сигналов, надо иметь возможность сжимать или растягивать вейвлеты и перемещать их по временной оси.

1.5.1 Основы теории вейвлет-преобразований

Прямое вейвлет-преобразование (ПВП) означает разложение произвольного входного сигнала на составляющие с использованием базиса в виде совокупности волновых пакетов – вейвлетов, которые характеризуются четырьмя принципиально важными свойствами:

- имеют вид коротких, локализованных во времени (или в пространстве) волновых пакетов с нулевым значением интеграла вейвлет-функции;
- обладают возможностью сдвига по оси времени;
- способны к масштабированию (сжатию-растяжению);
- имеют ограниченный (или локальный) частотный спектр.

Этот базис может быть ортогональным (см. ниже), что заметно облегчает анализ, дает возможность реконструкции сигналов и позволяет реализовать алгоритмы быстрых вейвлет-преобразований. Однако есть ряд вейвлетов, которые свойствами ортогональности не обладают, но которые, тем не менее, практически полезны — например, в задачах анализа и идентификации локальных особенностей сигналов и функций.

1.5.2 Аппроксимирующая и детализирующая компоненты вейвлетов

Одна из основополагающих идей вейвлет-представления сигналов заключается в разбивке приближения к сигналу на две составляющие – грубую (аппроксимирующую) и утонченную (детализирующую) – с после-

дующим их дроблением с целью изменения уровня декомпозиции сигнала. Это возможно как во временной, так и в частотной областях представления сигналов вейвлетами.

В основе непрерывного вейвлет-преобразования НВП (или CWT — Continue Wawelet Transform) лежит использование двух непрерывных и интегрируемых по всей оси t (или x) функций:

- *вейвлет-функция psi* $\psi(t)$ с нулевым значением интеграла $\int_{-\infty}^{\infty} \psi(t)dt = 0$, определяющая детали сигнала и порождающая детализирующие коэффициенты;

- масштабирующая, или *скейлинг-функция phi* $\varphi(t)$ с единичным значением интеграла $\int_{-\infty}^{\infty} \varphi(t)dt = 1$, определяющая грубое приближение (аппроксимацию) сигнала и порождающая коэффициенты аппроксимации.

Phi-функции $\varphi(t)$ присущи далеко не всем вейвлетам, а только тем, которые относятся к ортогональным. Такие вейвлеты мы рассмотрим в дальнейшем, а пока остановимся только на свойствах psi-функции $\psi(t)$ и на приближении ими локальных участков сигналов $s(t)$.

Psi-функция $\psi(t)$ создается на основе той или иной *базисной функции* - $\psi_0(t)$, которая, как и $\psi(t)$, определяет тип вейвлета. Базисная функция должна удовлетворять всем тем требованиям, которые были отмечены для *psi*-функции $\psi(t)$. Она должна обеспечивать выполнение двух основных операций:

- смещение по оси времени t – $\psi_0(t\text{-}b)$ при $b \in R$;

- масштабирование – $a^{-1/2}\psi_0\left(\frac{t}{a}\right)$ при $a > 0$ и $a \in R^+ \text{-}\{0\}$.

Параметр a задает ширину этого пакета, а b – его положение. В ряде литературных источников вместо явного указания времени t используется аргумент x, а вместо параметров a и b используются имеющие тот же смысл иные обозначения. Нетрудно убедиться в том, что следующее выражение задает сразу два этих свойства функции $\psi(t)$:

$$\psi(t) = a^{-1/2}\psi_0\left(\frac{t-b}{a}\right). \qquad (1.68)$$

Итак, для заданных a и b функция $\psi(t)$ и есть *вейвлет.* Вейвлеты являются вещественными функциями времени t и колеблются вокруг оси t (или x и т. д.). Параметр b задает положение вейвлетов, а параметр a -их масштаб. О вейвлетах, четко локализованных в пространстве, говорят, что они имеют компактный носитель.

1.5.3 Непрерывное прямое вейвлет-преобразование

Пусть энергия сигнала $s(t)$, равная $\int_R s^2(t)dt$, конечна в пространстве V сигнала с областью ограничения R. *Прямое непрерывное вейвлет-преобразование (ПНВП)* сигнала $s(t)$ задается, по аналогии с преобразованием Фурье, путем вычисления *вейвлет-коэффициентов* по формуле:

$$C(a,b) = \int_{-\infty}^{\infty} s(t) a^{-1/2} \psi\left(\frac{t-b}{a}\right) dt, \qquad (1.69a)$$

или с учетом области ограничения сигналов:

$$C(a,b) = \int_{R} s(t) a^{-1/2} \psi\left(\frac{t-b}{a}\right) dt. \qquad (1.69b)$$

Итак, вейвлет-коэффициенты определяются интегральным значением скалярного произведения сигнала на вейвлет-функцию заданного вида. Выражение (1.69b) используется как основное для функции прямого непрерывного вейвлет-преобразования в пакете Wavelet Toolbox.

1.5.4 Вейвлет-анализ сигналов с помощью спектрограмм

Пакет Wavelet Toolbox имеет специальные средства для построения спектрограмм сигналов, синтезированных вейвлетами. Эти спектрограммы представляют значения коэффициентов вейвлетов в плоскости масштаб (номера коэффициентов) – время. Внизу вейвлет-спектрограммы расположены коэффициенты с малыми номерами, дающие детальную картину сигнала, а сверху — с большими номерами, дающие огрубленную картину сигнала. При этом их значения определяют цвет соответствующей (обычно достаточно малой) области спектрограммы.

Вейвлет-спектрограммы являются важнейшим продуктом вейвлет-анализа сигналов и прекрасным дополнением к обычным спектрограммам на основе оконного преобразования Фурье, которые мы уже рассмотрели в разделах, посвященных пакету Signal Processing. Вейвлет-спектрограммы сигналов (рис. 1.27) порой выделяют такие особенности сигналов, которые просто незаметны на графиках сигналов и на Фурье-спектрограммах.

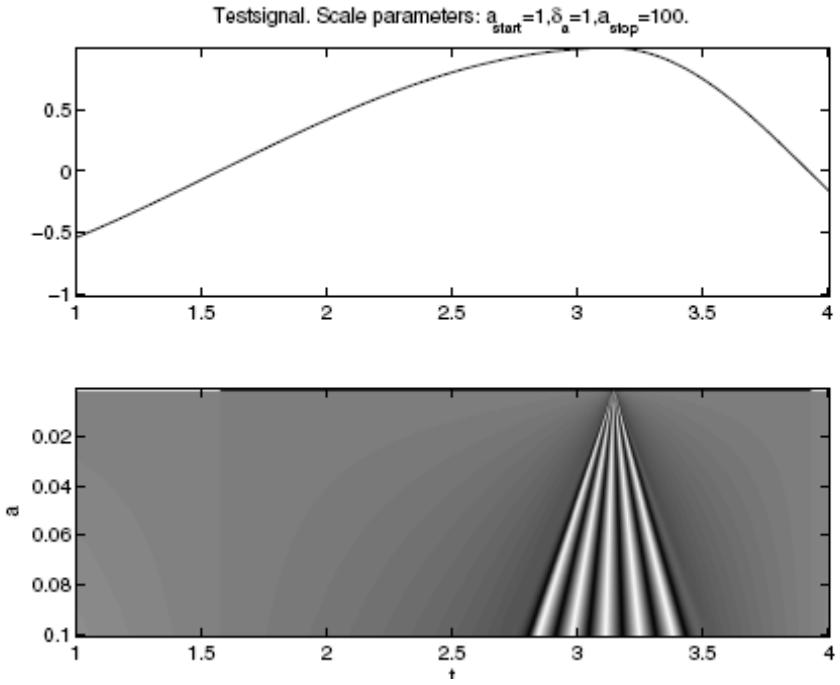


Рис. 1.27 - Сигнал с особенностями и его вейвлет-спектрограмма

Чистым гармоническим сигналам соответствуют яркие горизонтальные полосы, где модуль некоторого коэффициента вейвлета велик. Локальным особенностям (нарушениям гладкости) отвечают вертикальные

полосы, выходящие из точки, где находится особенность. Пикам сигналов соответствует сгущение светлых областей вейвлет-спектрограмм, а впадинам – сгущение темных областей.

Чем резче выражена особенность сигнала, тем сильнее она выделяется на спектрограмме. Вейвлет-спектрограммы отчетливо выделяют такие особенности сигнала, как небольшие разрывы, изменение знаков первой и второй производных и т. д. Словом, именно те особенности сигнала, которые плохо выделяются на спектре Фурье-сигнала, но прекрасно видны на вейвлет-спектрограммах.

Замечание. *Вейвлет-анализ сигналов открывает принципиально новые возможности в детальном анализе тонких особенностей сигналов. Это особенно важно для звуковых сигналов и сигналов изображения, где именно такие особенности подчас определяют качество их воспроизведения. Технические науки, медицина, астрономия и космос – все это именно те области, где применение вейвлетов способно привести к новым открытиям путем выявления характерных особенностей сигналов и изображений, мало заметных на временных зависимостях сигналов и на их спектрах Фурье.*

1.5.5 Вейвлеты в частотной области

Вейвлеты, будучи функциями времени, имеют свое *частотное представление,* или Фурье-образ $\hat{\psi}(\omega)$. Налагаемое на функцию $\psi(t)$ условие (нулевое значение интеграла) означает, что $\hat{\psi}(0) = 0.$ Последнее указывает на то, что Фурье-образ смещен по оси времени и будет расположен вокруг некоторой ненулевой частоты ω_0, которую можно рассматривать как *среднюю круговую частоту* вейвлета.

В частотной области спектры многих вейвлетов напоминают всплеск, пик которого приходится на частоту ω_0 (рис.1.23). Если приближенно трактовать вейвлет как модулированную синусоиду, то ее частота и будет средней частотой вейвлета. В общем же случае, когда временная зависимость вейвлетов далека от синусоидальной, определение средней частоты требует обработки сигнала и реализуется итерационными методами [14].

Частотное (спектральное) представление вейвлетов имеет важное значение в определении фильтрующих свойств вейвлет-преобразований и основанном на них алгоритме *быстрого вейвлет-преобразования (БВП).* Нетрудно заметить, что есть прямая связь между временным и частотным представлением вейвлетов. Так, малые значения параметра *a,* характеризующие быстрые процессы в сигналах, соответствуют высоким частотам, а большие значения (соответствующие медленным изменениям сигнала) – низким частотам.

Замечание. *Временное и частотное представление вейвлетов – это две стороны одной медали, имя которой – вейвлет. Они образуют неразлучную пару и могут легко преобразовываться друг в друга. И каждое такое преобразование имеет свои достоинства и недостатки.*

Основанные на частотном подходе вейвлет-преобразования с помощью фильтров будут описаны далее.

1.5.6 Непрерывное обратное вейвлет-преобразование

Обратное непрерывное вейвлет-преобразование (ОНВП) осуществляется по формуле реконструкции во временной области, которая имеет ряд форм. Ниже представлена эта формула в виде, использованном в пакете расширения системы MATLAB 7.0/7.x — Wavelet Toolbox:

$$s(t) = \frac{1}{K_\psi} \int_{R^+} \int_R C(a,b) a^{-1/2} \psi\left(\frac{t-b}{a}\right) \frac{dadb}{a^2}, \qquad (1.70)$$

где K_v – константа, определяемая функцией ψ.

Основной задачей теории вейвлет-преобразований является доказательство того, что прямое и обратное вейвлет-преобразования способны обеспечить *реконструкцию* сигнала, причем точную или хотя бы приближенную, локальную или для сигнала в целом на заданном промежутке времени. Учитывая нулевое значение интеграла для функции $\psi(t)$, очевидно, что эти преобразования не всегда способны восстановить любой сигнал в целом.

Итак, вейвлет-преобразование на основе только детализирующей ортогональной вейвлет-функции $\psi(t)$ способно восстановить (реконструировать) лишь тонкие детали временной зависимости сигнала $s(t)$. Для восстановления полной формы сигнала приходится прибегать к применению еще одной временной функции $\varphi(t)$, называемой аппроксимирующей. Причины, порождающие необходимость в использовании этой функции, и ее роль будут рассмотрены чуть ниже – при описании кратномасштабного анализа.

Замечание. *Далеко не все типы вейвлетов гарантируют точную реконструкцию сигналов в целом и даже таковую вообще. Тем не менее применение и таких вейвлетов может быть полезно для выявления тонких особенностей сигналов или изображений, которые хорошо согласуются с определенными типами вейвлетов.*

1.5.7 Сравнение различных представлений сигналов

Следует отметить, что вейвлет-анализ не использует амплитудно-частотную область для визуального представления спектров сигналов, как это имеет место при спектральном анализе Фурье. Вместо нее используете область масштаб – время . Это схематично показывает рис. 1.28 [15].

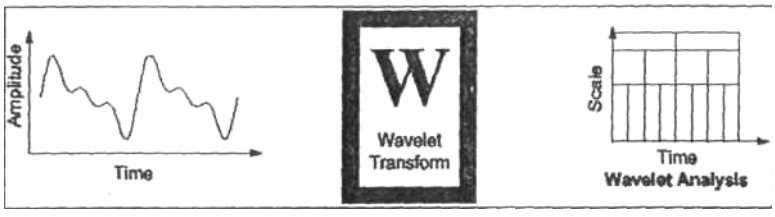


Рис.1.28 - Структура вейвлет-преобразования

Теперь мы можем наглядно отобразить различные виды представлений сигналов в ходе тех или иных их преобразований – рис. 1.29 [15].

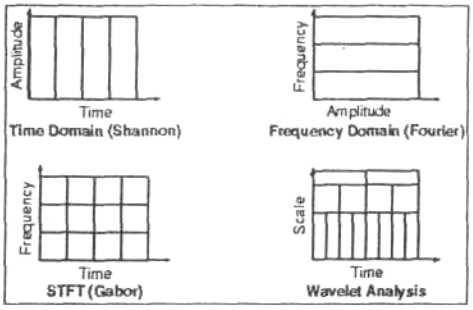


Рис 1.29 - Различные представления сигналов

На рис. 1.29 представлены следующие наиболее известные формы представления сигналов:

- Time Domain – временное представление (по Шеннону);
- Frequency Domain – частотное представление (по Фурье);
- STFT – кратковременное (оконное) быстрое преобразование Фурье;
- Wavelet – вейвлет-преобразование.

Нетрудно заметить, что вейвлет-преобразование отличается наиболее сложной и гибкой структурой представления сигналов в плоскости масштаб-время (Scale-Time).

1.5.8 О скорости вычислений при вейвлет-преобразованиях

В чем же, с точки зрения затрат машинного времени, вейвлет-преобразование лучше, чем преобразование Фурье? На этот вопрос ответить однозначно нельзя. Тем не менее, важно учитывать следующие обстоятельства:

- вейвлет-преобразование открывает принципиально новые возможности в обработке сигналов и изображений, и в этом случае затраты времени на вычисления отходят на второй план (особенно с учетом постоянного роста производительности компьютеров);

- некоторые вейвлеты (например, Хаара) являются намного более простыми функциями, чем синусоидальная функция, и в этом случае затраты времени на вейвлет-преобразования могут быть заметно ниже, чем на преобразования Фурье, где вычисления множества трансцендентных тригонометрических функций требуют значительных затрат времени;

a — их масштаб. О вейвлетах, четко локализованных в пространстве (или во времени), говорят, что они имеют *компактный носитель*.

Применительно к сигналам, как функциям времени, параметр $b \in R$ задает положение вейвлета на временной оси, а параметр *a* задает его масштабирование по времени. Поскольку параметр масштаба *a* реально может быть только положительным и его нельзя брать равным нулю, то считается, что $a \in R^+ - \{0\}$. В дальнейшем мы будем опускать выражение - {0}, означающее исключение значения $a = 0$.

На рис. 1.26 показано построение вейвлета, уже известного нам под названием «мексиканская шляпа». Для вычисления и построения графиков этого вейвлета использована популярная СКМ Mathcad.

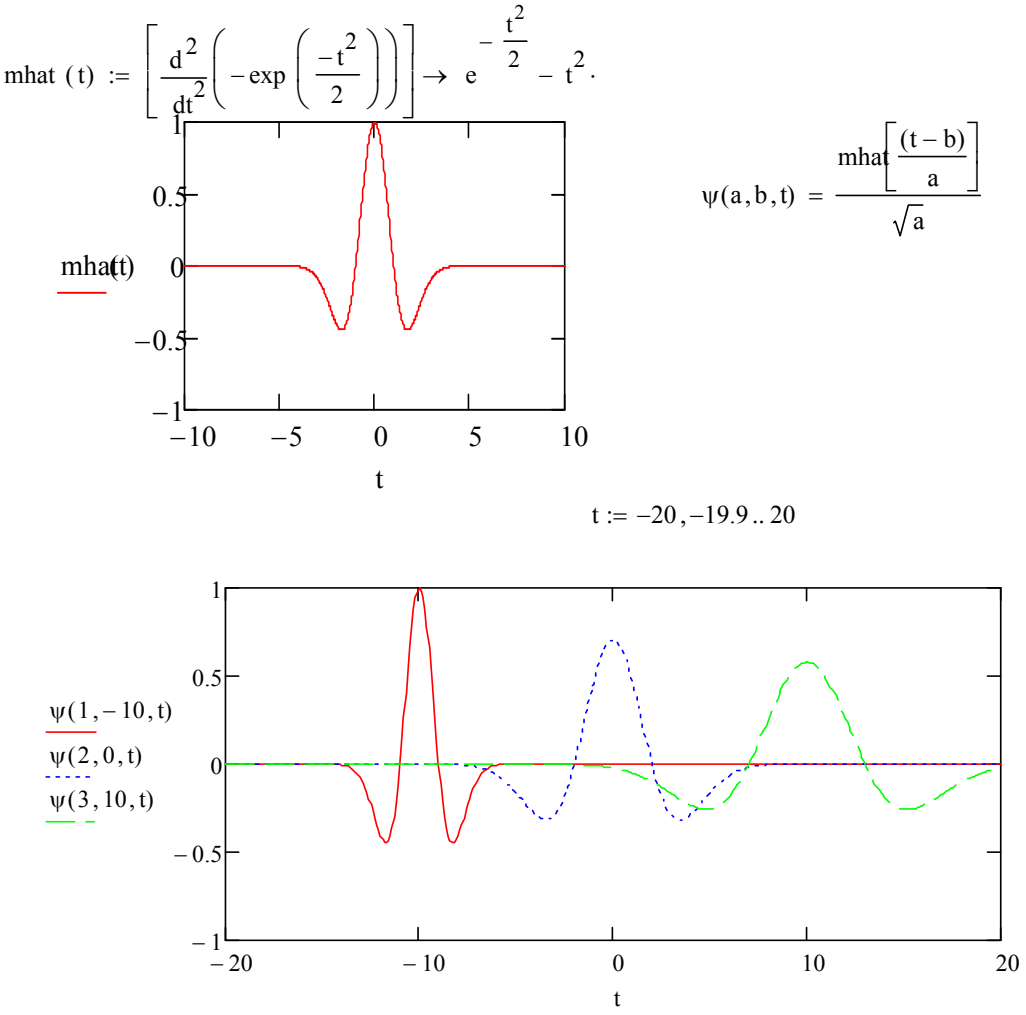


Рис. 1.30 - Иллюстрация сдвига и масштабирования вейвлета типа «мексиканская шляпа»

На рис. 1.30 представлена базисная функция данного вейвлета и функция $\psi(a, b, t)$ для разных a и b, что иллюстрирует сдвиг вейвлета и его масштабирование. В Mathcad для обеспечения изменений a и b функцию $\psi(a, b, t)$ приходится задавать в более полном виде, чем $\psi(t)$, в котором зависимость ψ от a и b лишь подразумевается

В частотной области малые значения a соответствуют высоким частотам, а большие – низким частотам. Таким образом, операция задания окна, используемая в оконном преобразовании Фурье, как бы заложена в самой базисной функции вейвлетов. Это создает предпосылки их приспособления (адаптации) к сигналам, которые могут быть представлены совокупностью вейвлетов.

1.5.9 Кратномасштабный вейвлет-анализ

Рассмотрим $L^2(R)$ – гильбертово пространство функций x(t), для которых $\int_{-\infty}^{\infty} |x(t)|^2 \, dt < \infty$. В этом пространстве определено скалярное произведение $\langle x, y \rangle = \int_{-\infty}^{\infty} x(t)\bar{y}(t)dt$ и норма $\|x\| = \sqrt{\langle x, x \rangle}$. Базисом в пространстве $V \subset L^2(R)$ называют такую систему функций v_i(t), в которой любая функция этого пространства единым образом представляется в виде $v(t) = \sum_i c_i v_i(t)$.

В свою очередь, базис называют ортонормированным, если $\langle v_i(t), v_j(t) \rangle = \delta_{ij}$. В этом случае $c_i = \langle v(t), v_i(t) \rangle$.

Под ортогональным кратномасштабным анализом понимают рассмотрение пространства $L^2(R)$ в виде подпространств $V_j \subset L^2(R)$, удовлетворяющих следующим условиям:

1) подпространства замкнуты и вложены друг в друга: $V_j \subset V_{j+1}$;

2) подпространства не пересекаются, а замыкание их объединения совпадает с $L^2(R) : \cap V_j = \{0\}, \overline{\cup V_j} = L^2(R)$;

3) сдвиг функции сохраняет ее в подпространстве:

$$v(t) \in V_0 \Leftrightarrow v(t+1) \in V_0;$$

4) масштабирование аргумента в два раза перемещает функцию в соседнее подпространство: $v(t) \in V_j \Leftrightarrow v(2t) \in V_{j+1}$;

5) существует масштабирующая функция $\varphi(t) \in V_0$, сдвиги которой $\varphi_{0,m}(t) = \varphi(t-m), m \in Z,$ образуют ортонормированный базис пространства V_0. Поскольку функции $\varphi_{0,m}(t)$ составляют ортонормированный базис пространства V_0, то функции $\varphi_{j,m}(t) = 2^{j/2}\varphi(2^j t - m),\ j, m \in Z$ порождают ортонормированный базис пространства V_j.

Если последовательность V_j удовлетворяет указанным условиям, то для каждого $j \in Z$ ортогональное дополнение W_j к пространству V_j в пространстве V_{j+1} $(v_{j+1} = V_j \oplus W_j)$ называют пространством вейвлетов, а его элементы – вейвлетами. При этом вводится также функция $\psi(t) \in W_0$, называемая материнским вейвлетом, множество сдвигов которой ψ(t-m) образует ортонормированный базис пространства W_0, а функции $\psi_{j,m}(t) = 2^{j/2}\psi(2^j t - m)$ образуют ортонормированные базисы пространств W_j для каждого $j \in Z$.

Сигнал $x(t)$ на j-м уровне кратномасштабного вейвлет-разложения представляется в виде

$$x(t) = \sum_{m=0}^{N/2^j-1} a_{j,m}\varphi_{j,m}(t) + \sum_{k=1}^{j}\sum_{m=0}^{N/2^k-1} d_{k,m}\psi_{k,m}(t), \qquad (1.71)$$

где N – общее количество отсчетов сигнала.

Здесь первое слагаемое выражения (1.71) – это сглаженные, аппроксимирующие значения сигнала $x(t)$ при некотором масштабе. Второе слагаемое добавляет к «грубой» аппроксимации сигнала все более уточняющие детали на все меньших масштабных интервалах.

Определение коэффициентов $a_{\text{j,m}}$ и $d_{\text{k,m}}$ влечет за собой проблему вычисления большого количества интегралов с необходимой точностью. Эту проблему решает алгоритм быстрого вейвлет-преобразования (БВП), предложенный Малла [14]. Он тесно связан с методами субполосного кодирования и дает возможность вычислять коэффициенты вейвлет-разложения без интегрирования, используя алгебраические операции на основе свертки. При вычислении коэффициентов БВП используют не вейвлеты, а образованные ими низкочастотные (НЧ) и высокочастотные (ВЧ) фильтры. Каждому ортогональному вейвлету соответствует четыре фильтра: h_{n} – НЧ фильтр декомпозиции, g_{n} – ВЧ фильтр декомпозиции, $\tilde{h}_n$ – НЧ фильтр реконструкции, $\tilde{g}_n$ – ВЧ фильтр реконструкции [15, 139]. Все они взаимосвязаны; h_n и g_n, $\tilde{g}_n$ и $\tilde{h}_n$ являются квадратурно-зеркальными фильтрами (КЗФ), т.е. фильтрами, частотные характеристики которых есть зеркальным отражением друг друга относительно средней частоты.

В общем случае итерационные формулы БВП имеют вид:

$$a_{j+1,m} = \sum_n h_n a_{j,2m+1},\ d_{j+1,m} = \sum_n g_n a_{j,2m+1},$$

где $a_{0,m} = \int x(t)\varphi(t-m)dt.$

Для сигнала, заданного массивом своих отсчетов, начальные коэффициенты разложения, как правило, выбирают равными значениям отсчетов: $a_{0,m} = x(t_n)$.

Применив к исходному сигналу фильтр h_{n} с дальнейшей децимацией $\downarrow 2$, состоящей в отбрасывании каждого второго из полученных коэффициентов, находим сглаженную составляющую сигнала $A_1 = \{a_{1,\text{m}}\}$. Далее, используя к $x(t_{\text{n}})$ фильтр g_{n} с децимацией $\downarrow 2$, получим $D_1 = \{d_{1,\text{m}}\}$ – детали, потерянные при сглаживании. Затем проведем декомпозицию для субполосы A_1, результатом которой являются коэффициенты второго уровня разложения $A_2 = \{a_{2,\text{m}}\}$ и $D_2 = \{d_{2,\text{m}}\}$. Повторяя процедуру декомпозиции нужное количество раз, вместо сигнала $x(t_{\text{n}})$ получаем серию его вейвлет-коэффициентов (субполос) $X=\{A_{\text{j}}, D_{\text{j}}, D_{\text{j-1}}, \ldots, D_2, D_1\}$.

В частотной области вейвлет-декомпозиция приводит к октавополосному разбиению сигнала (рис. 1.31). Ширина субполос, соответствующих

каждому новому уровню разложения, вдвое меньше по сравнению с предыдущим уровнем.

Таким образом, кратномасштабный вейвлет-анализ сигнала состоит в изучении и обработке коэффициентов разложения разных уровней. К примеру, анализируя полученное частотно-временное представление аудиосигнала с учетом психоакустических особенностей системы человеческого слуха и влияния на спектр типичных операций обработки, можно определить области, наиболее подходящие для неощутимого стойкого внедрения дополнительной информации (стеганографическая передача информации аудиосигналом, т.н. цифровые «водяные» знаки – ЦВЗ).

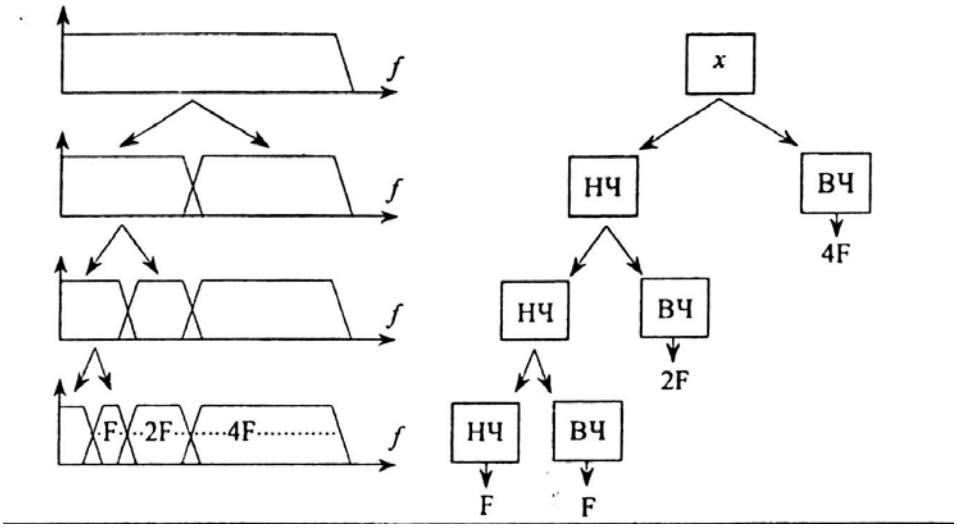


Рис. 1.31

После модификации коэффициентов декомпозиции процедурой внедрения битов ЦВЗ выполняется реконструкция сигнала, т.е. его восстановление из субполос в единое целое. На данном шаге используется интерполяция $\uparrow 2$ (вставка нулевого коэффициента между каждыми соседними из имеющихся) и фильтрация с помощью $\tilde{h}_n$ и $\tilde{g}_n$.

В свою очередь, следует отметить, что процедура декомпозиции может быть применена не только к аппроксимирующим, но и к детализирующим коэффициентам. Такой метод анализа называют *вейвлет-пакетной декомпозицией сигнала*. Применение такого преобразование способствует лучшей частотной локализации сигналов и соответственно более эффективному представлению области внедрения битов ЦВЗ по сравнению с обычным БВП. При этом следует отметить, что в отличие от БВП, осуществляющего октавополостное частотное разбиение сигнала, пакетное преобразование позволяет выполнить частотное разбиение согласно критическим полосам (например, слуха), что дает возможность определять свои, оптимальные параметры внедрения для коэффициентов каждой субполосы.

Контрольные вопросы к разделу I:

1. Раскрыть термин сигнал. Какие сигналы Вы знаете?
2. Привести функции Дирака и Хевисайда с указанием их связи и свойств.
3. Привести примеры сигналов с конечной и бесконечной энергией.
4. Ряд Фурье и условия Дирихле.
5. Четные и нечетные функции: их представление в виде ряда Фурье.
6. Свойства преобразования Фурье.
7. Прямое и обратное ДПФ.
8. Суть вейвлет-преобразования сигнала: прямого и обратного.
9. В чём заключается кратномасштабность вейвлет-преобразования?
10. Для чего применяется вейвлет-пакетная декомпозиция сигнала?

РАЗДЕЛ II
ВЫЯВЛЕНИЕ АССОЦИАЦИЙ И ЗАКОНОМЕРНОСТЕЙ*

2.1 Классические методы выявления закономерностей

2.1.1. Метод наименьших квадратов

Пусть задана система точек:

Таблица 2.1 – Экспериментальные данные

x_i	x_0	x_1	…	x_N
t_i	t_0	t_1	…	t_N

Число точек N велико и данные получены с ошибкой, что собственно является вполне типичным при обработке экспериментальных данных. В этом случае использование интерполяционных методов нецелесообразно. Кроме того, возможна ситуация, когда известна априорная информация об исследуемом процессе, тогда вид аппроксимирующей функции определяется технологическими условиями или природой явления, а выбор коэффициентов этой функции обусловлен требованиями получения адекватной модели самого процесса, т.е. выполнения главного требования – ошибка моделирования должна быть минимальной. При этом понятно, что выбор критерия близости модели к объекту (*критерия адекватности модели*) является принципиальным и наиболее ответственным этапом обработки данных. Как правило, при решении такого рода задач в качестве критерия используют среднеквадратическое расстояние, описываемое квадратичной функцией – параболоидом, имеющим в силу своей выпуклости единственный экстремум. Это существенно упрощает решение задачи минимизации критерия, поскольку в этом случае необходимое условие существования экстремума совпадает с достаточным.

Метод нахождения экстремума среднеквадратической функции цели принято называть **М**етодом **Н**аименьших **К**вадратов (МНК) [18,19]. При этом идея среднеквадратичного приближения принадлежит Адриену Мари Лежандру (1806 г.), а сам метод – Карлу Фридриху Гауссу (1809 г.). Отсюда название метода – МНК или метод Гаусса.

Поставим в соответствие исходным данным, заданным в виде таблицы 2.1, функцию вида

$$F\left(\{a_i\}_{i=0}^{n}, t\right) = \sum_{i=0}^{n} a_i \varphi_i(t) = a_0\varphi_0(t) + a_1\varphi_1(t) + \ldots + a_n\varphi_n(t), \qquad (2.1)$$

* Данный раздел существенно опирается на книгу Шумейко А.А., Сотник С.Л. [95], на сайте https://cmidpbook.codeplex.com/ размещен программный материал данного раздела.

где $\varphi_i(t)$, $i=0,...,n$ - базисные функции, a_i – неизвестные коэффициенты, подлежащие определению. В частности, если в качестве базисных функций использовать степенные мономы $\varphi_i(t)=t^i$, задача сводится к поиску полинома

$$F\left(\{a_i\}_{i=0}^n,t\right)=\sum_{i=0}^{n}a_i t^i=a_0+a_1t+...+a_nt^n$$

степени *n,* приближающего значения исходной таблицы.

Для определения коэффициентов a_i будем искать функцию $F\left(\{a_i\}_{i=0}^n,t\right)$, отклонение значений которой от заданных таблицей значений x_i минимально в некотором средне интегральном смысле.

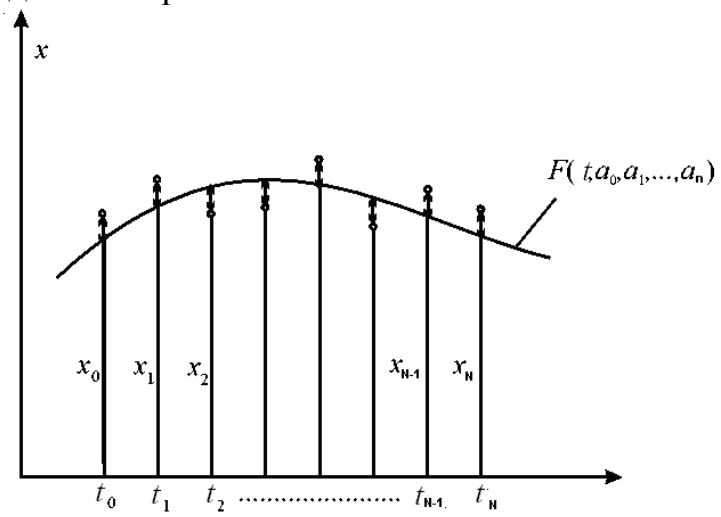


Рис. 2.1- Иллюстрация МНК

В частности, в дискретном методе наименьших квадратов строится функционал цели

$$\begin{aligned}S(a_0,a_1,...,a_n)&=\sum_{i=0}^{N}\left(F(a_0,a_1,...,a_n,t_i)-x_i\right)^2\rho_i^2=\\&=\sum_{i=0}^{N}\left(a_0\phi_0(t_i)+a_1\phi_1(t_i)+...+a_n\phi_n(t_i)-x_i\right)^2\rho_i^2,\end{aligned}\qquad(2.2)$$

где ρ_i – некоторые неотрицательные числа (весовые коэффициенты).

Если все значения равноправны, то весовые коэффициенты берутся равными единице.

Геометрически функционал (2.2) представляет собой сумму квадратов отклонений с весом ρ_i экспериментальных данных x_i от значений аппроксимирующей функции $F(a_0,a_1,...,a_n,t)$ в точках t_i *(i=0,...,N).*

Необходимым (а в данном случае, в силу выпуклости функционала цели, и достаточным) условием минимума функции многих переменных

$$S(a_0,a_1,...,a_n)\to\min_{a_0,a_1,...,a_n}$$

является равенство нулю ее частных производных первого порядка по независимым переменным:

$$\begin{cases} \dfrac{\partial S}{\partial a_0} = 2\sum\limits_{i=0}^{N}\left(F\left(a_0,a_1,...,a_n,t_i\right)-x_i\right)\varphi_0\left(t_i\right)\rho_i^2 = 0, \\ \dfrac{\partial S}{\partial a_1} = 2\sum\limits_{i=0}^{N}\left(F\left(a_0,a_1,...,a_n,t_i\right)-x_i\right)\varphi_1\left(t_i\right)\rho_i^2 = 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \dfrac{\partial S}{\partial a_n} = 2\sum\limits_{i=0}^{N}\left(F\left(a_0,a_1,...,a_n,t_i\right)-x_i\right)\varphi_n\left(t_i\right)\rho_i^2 = 0. \end{cases} \qquad (2.3)$$

Полученная система представляет собой систему линейных алгебраических уравнений порядка *n + 1* относительно неизвестных $a_0, a_1, ..., a_n$. Система разрешима при условии $n \le N$. Ее матрица является симметрической и положительно определенной. Решения $a_0, a_1, ..., a_n$ доставляют *минимум* функционалу (2.2).

Решение данной системы может быть осуществлено любым из известных методов (например, методом Гаусса, Крамера и др.). Подставляя найденные в результате решения системы (2.3) значения $a_0, a_1, ..., a_n$ в (2.1), получаем функцию *F(t),* наилучшим образом приближающую исходные данные в среднеквадратическом смысле. Качество такого приближения может быть оценено величиной среднеквадратичного отклонения

$$\sigma = \sqrt{\frac{1}{N+1}\sum_{i=0}^{N}\left(x_i - F\left(t_i\right)\right)^2 \rho_i^2}.$$

Достаточно часто естественным условием является требование описания исходных данных прямой или параболой. В этом случае говорят, что используется линейная или квадратичная регрессия.

Рассмотрим случай описания априорных данных прямой, то есть используем метод линейной регрессии. Опишем исходные данные $\left(\mathrm{t_i}, \mathrm{x_i}\right), \mathrm{i} = 0,1,...,\mathrm{N}$ прямой $x = at + b.$ В этом случае функция цели (2.2) примет вид

$$S\left(a,b\right) = \sum_{i=0}^{N}\left(at_i + b - x_i\right)^2 \to \min_{a,b}.$$

Необходимое (и, в данном квадратичном случае, достаточное) условие экстремума имеет вид

$$\begin{cases} \dfrac{\partial}{\partial a}S\left(a,b\right) = 2\sum\limits_{i=0}^{N}t_i\left(at_i + b - x_i\right) = 0, \\ \dfrac{\partial}{\partial b}S\left(a,b\right) = 2\sum\limits_{i=0}^{N}\left(at_i + b - x_i\right) = 0, \end{cases}$$

или, что, то же,

$$\begin{cases} a\sum\limits_{i=0}^{N}t_i^2 + b\sum\limits_{i=0}^{N}t_i = \sum\limits_{i=0}^{N}x_i t_i, \\ a\sum\limits_{i=0}^{N}t_i + b(N+1) = \sum\limits_{i=0}^{N}x_i. \end{cases}$$

Отсюда, применяя метод Крамера решения систем линейных уравнений, сразу получаем коэффициенты прямой (линейной регрессии)

$$a = \frac{(N+1)\sum_{i=0}^{N} x_i t_i - \sum_{i=0}^{N} t_i \sum_{i=0}^{N} x_i}{(N+1)\sum_{i=0}^{N} t_i^2 - \left(\sum_{i=0}^{N} t_i\right)^2}, b = \frac{\sum_{i=0}^{N} t_i^2 \sum_{i=0}^{N} x_i - \sum_{i=0}^{N} t_i \sum_{i=0}^{N} x_i t_i}{(N+1)\sum_{i=0}^{N} t_i^2 - \left(\sum_{i=0}^{N} t_i\right)^2}.$$

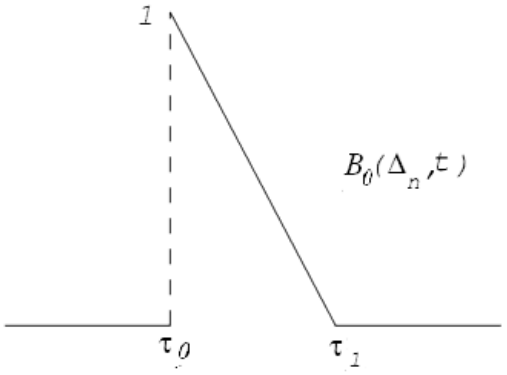
Достаточно распространенным аппаратом приближения являются кусочно-полиномиальные функции или сплайны. Наиболее распространенным видом сплайнов, как известно, являются ломаные или полигональные функции. Рассмотрим в качестве регрессионной модели ломаную с весовой функцией, равной единице. Пусть Δ_n фиксированное разбиение отрезка $[t_0, T]$ точками $\tau_i (i = 0,1,2,...,n)$, и $\Re(\Delta_n)$ - множество ломаных $P(\Delta_n, t) = P\left(\{a_i\}_{i=0}^n, \Delta_n, t\right)$ с узлами в точках разбиения Δ_n. Тогда задача нахождения кусочно-линейной регрессионной модели с фиксированными узлами имеет вид

$$\inf\left\{\sum_{i=0}^{N} (x_i - P(\Delta_n, t_i))^2 \middle| P(\Delta_n) \in \Re(\Delta_n)\right\}.$$

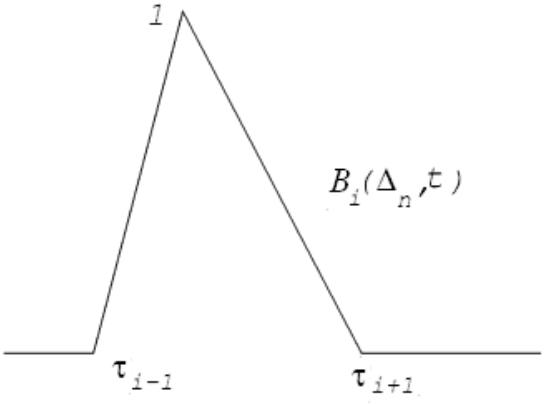
Нетрудно видеть, что

$$P(\Delta_n, t) = P\left(\{a_i\}_{i=0}^n, \Delta_n, t\right) = \sum_{i=0}^{n} a_i B_i(\Delta_n, t),$$

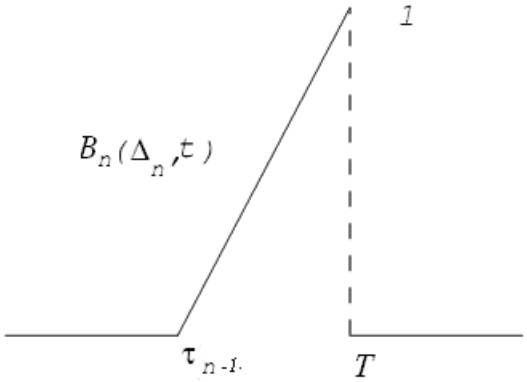
где $B_i(\Delta_n, t)$ *(i = 0,...,n)* базисные функции, которые можно записать в следующем виде:



$$B_0(\Delta_n, t) = \begin{cases} (\tau_1 - t)(\tau_1 - \tau_0)^{-1}, & t \in [\tau_0, \tau_1], \\ 0, & \textit{иначе}, \end{cases}$$



$$B_i(\Delta_n,t)=\begin{cases}(\tau_{i-1}-t)(\tau_{i-1}-\tau_i)^{-1}, & t\in[\tau_{i-1},\tau_i],\\ (\tau_{i+1}-t)(\tau_{i+1}-\tau_i)^{-1}, & t\in[\tau_i,\tau_{i+1}],\\ 0, & \textit{иначе},\end{cases}\quad (i=2,...,n-1).$$



$$\mathrm{B_n}\left(\Delta_{\mathrm{n}},\mathrm{t}\right)=\begin{cases}\left(\tau_{\mathrm{n}-1}-\mathrm{t}\right)\left(\tau_{\mathrm{n}-1}-\mathrm{T}\right)^{-1}, & \mathrm{t}\in\left[\tau_{\mathrm{n}-1},\mathrm{T}\right],\\ 0, & \text{иначе.}\end{cases}$$

Запишем функцию цели

$$S(a_0,a_1,...,a_n)=\sum_{i=0}^{N}\left(\sum_{j=0}^{n}a_jB_j(\Delta_n,t_i)-x_i\right)^2,$$

и найдем решение задачи $S(a_0,a_1,...,a_n)\to\min\limits_{a_0,a_1,...,a_n}$. Необходимое условие экстремума при этом будет иметь вид

$$\frac{\partial}{\partial a_i}S(a_0,a_1,...,a_n)=0,i=0,1,...,n.$$

Нахождение экстремума сводится к решению системы уравнений

$$\begin{pmatrix}\langle B_0,B_0\rangle & \langle B_0,B_1\rangle & \cdots & \langle B_0,B_n\rangle\\ \langle B_1,B_0\rangle & \langle B_1,B_1\rangle & \cdots & \langle B_1,B_n\rangle\\ \vdots & \vdots & \ddots & \vdots\\ \langle B_n,B_0\rangle & \langle B_n,B_1\rangle & \cdots & \langle B_n,B_n\rangle\end{pmatrix}\begin{pmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{pmatrix}=\begin{pmatrix}\langle x,B_0\rangle\\ \langle x,B_1\rangle\\ \vdots\\ \langle x,B_n\rangle\end{pmatrix},$$

где

$$\langle x,B_j\rangle=\sum_{i=0}^{N}x_iB_j(\Delta_n,t_i),j=0,1,...,n,$$

а замечая, что $\langle B_i,B_j\rangle=0,\forall i,j:|i-j|\geq 2,$ получаем систему уравнений с трехдиагональной матрицей

$$\mathbf{A}=\begin{pmatrix}\langle \mathrm{B}_0,\mathrm{B}_0\rangle & \langle \mathrm{B}_0,\mathrm{B}_1\rangle & 0 & \cdots & 0\\ \langle \mathrm{B}_1,\mathrm{B}_0\rangle & \langle \mathrm{B}_1,\mathrm{B}_1\rangle & \langle \mathrm{B}_1,\mathrm{B}_2\rangle & \cdots & 0\\ 0 & \langle \mathrm{B}_2,\mathrm{B}_1\rangle & \langle \mathrm{B}_2,\mathrm{B}_2\rangle & \cdots & 0\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ 0 & 0 & 0 & \cdots & \langle \mathrm{B}_n,\mathrm{B}_n\rangle\end{pmatrix}.$$

Применяя метод прогонки, получаем эффективный алгоритм нахождения уравнения кусочно-линейной регрессии с фиксированными узлами.

Приведем пример построения ломаной методом наименьших квадратов.

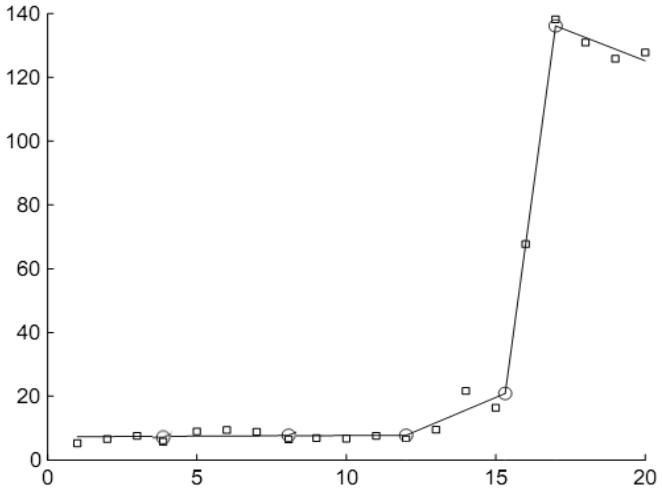


Рис. 2.2 - Приближение дискретных данных ломаной по МНК

Линеаризация при использовании метода наименьших квадратов

Приведенная методология определения аппроксимирующих функций методом наименьших квадратов годится лишь для функций, у которых неопределенные коэффициенты заданы линейно. Если же это условие не выполняется, то прямое использование метода наименьших квадратов невозможно.

В этом случае для применения МНК необходимо провести некоторые дополнительные построения, линеаризующие (по коэффициентам) приближающую функцию (см. [18]).

Проиллюстрируем это на нескольких примерах.

1. Пусть приближающая функция имеет вид $x = \dfrac{1}{\alpha t + \beta}$.

Тогда для x_i ошибка будет равна

$$\delta_i = x_i - \frac{1}{\alpha t_i + \beta}. \tag{2.4}$$

Прямое использование метода наименьших квадратов приводит к необходимости минимизации величины

$$\sum_{i=1}^{n} \delta_i^{\,2} = \sum_{i=1}^{n} \left(x_i - \frac{1}{\alpha t_i + \beta} \right)^2. \tag{2.5}$$

Взяв частные производные по α и β, и приравняв их к нулю, получим систему из двух нелинейных уравнений, которая не подлежит точному решению. В связи с этим, проведем некоторые построения.

Рассмотрим величины

$$\Delta_i = x_i(\alpha t_i + \beta) - 1, (i = 1,2,...,n).$$

Установим зависимость между Δ_i и δ_i. Из (2.4) получаем

$$\alpha t_i + \beta = \frac{1}{x_i - \delta_i}.$$

Тогда

$$\Delta_i = \frac{x_i}{x_i - \delta_i} - 1 = \frac{\delta_i}{x_i - \delta_i}, (i = 1,2,...,n),$$

и, следовательно, при малых Δ_i

$$\delta_i = \frac{x_i \Delta_i}{\Delta_i + 1} \approx x_i \Delta_i.$$

Отсюда задача (2.5) сводится к задаче определения коэффициентов α и β так, чтобы величина

$$\sum_{i=1}^{n} \left(x_i \Delta_i \right)^2 = \sum_{i=1}^{n} \left(1 - \alpha t_i x_i - \beta x_i \right)^2 x_i^2$$

была минимальной.

Таким образом, мы пришли к задаче (2.1) при условии, что приближаемая функция тождественно равна единице и $\phi_0(t) = tx(t), \phi_1(t) = x(t)$ с весом $\rho_i = x_i$.

При этом погрешность приближения имеет вид

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(x_i - \frac{1}{\alpha t_i + \beta} \right)^2}.$$

2. Рассмотрим ещё один пример. Пусть приближающая функция имеет вид

$$x = \frac{t}{\alpha t + \beta}.$$

Для x_i ошибка будет равна

$$\delta_i = x_i - \frac{t_i}{\alpha t_i + \beta} \qquad (2.6)$$

и

$$\Delta_i = x_i \left(\alpha t_i + \beta \right) - t_i, (i = 1,2,...,n).$$

Установим связь между Δ_i и δ_i. Из (2.6) имеем

$$\alpha t_i + \beta = \frac{t_i}{x_i - \delta_i}.$$

Тогда

$$\Delta_i = \frac{t_i x_i}{x_i - \delta_i} - t_i = \frac{t_i \delta_i}{x_i - \delta_i}, (i = 1,2,...,n).$$

Отсюда при малых Δ_i

$$\delta_i = \frac{x_i \Delta_i}{\Delta_i + t_i} \approx \frac{x_i}{t_i} \Delta_i.$$

Для малых δ_i $\sum_{i=1}^{n} \delta_i^2 \approx \sum_{i=1}^{n} \left(\frac{x_i}{t_i} \Delta_i \right)^2$ и $\sum_{i=1}^{n} \left(\frac{x_i}{t_i} \Delta_i \right)^2 = \sum_{i=1}^{n} \left(t_i - \alpha t_i x_i - \beta x_i \right)^2 \left(\frac{x_i}{t_i} \right)^2$.

Таким образом, мы пришли к задаче (2.1) при условии, что приближаемая функция тождественно равна t и $\phi_0(t) = tx(t), \phi_1(t) = x(t)$ и $\rho_i = x_i / t_i$.

При этом для данного примера погрешность приближения имеет вид

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \frac{t_i}{\alpha t_i + \beta}\right)^2}.$$

3. Наконец, пусть приближающая функция имеет вид

$$x = \frac{\alpha t + \beta}{\gamma t + 1}.$$

Задача состоит в нахождении коэффициентов α, β, γ, при которых величина

$$\sum_{i=1}^{n}\delta_i^2 = \sum_{i=1}^{n}\left(x_i - \frac{\alpha t_i + \beta}{\gamma t_i + 1}\right)^2$$

будет минимальной.

Линеаризуем эту задачу.

Пусть

$$\Delta_i = \gamma t_i x_i + x_i - \alpha t_i - \beta, (i = 1,2,...,n).$$

Установим связь между Δ_i и δ_i. Из предыдущего имеем

$$\frac{\Delta_i}{\gamma t_i + 1} = x_i - \frac{\alpha t_i + \beta}{\gamma t_i + 1}.$$

Таким образом, имеем

$$\frac{\Delta_i}{\gamma t_i + 1} = \delta_i.$$

Для малых δ_i получаем задачу, эквивалентную искомой

$$\sum_{i=1}^{n}\left(\frac{\Delta_i}{\gamma t_i + 1}\right)^2 = \sum_{i=1}^{n}\left(x_i - \alpha t_i - \beta + \gamma t_i x_i\right)^2\left(\frac{1}{\gamma t_i + 1}\right)^2 \to \min.$$

Однако использовать метод наименьших квадратов не представляется возможным, так как в "вес" входит неизвестный параметр γ. Поэтому рассмотрим итерационный метод пошагового уточнения весовых коэффициентов.

Для задачи (2.2) положим $\varphi_0(t) = t, \varphi_1(t) = 1, \varphi_2(t) = -tx(t)$ и $\rho_i = 1.$ Решая эту задачу, получаем первое приближение $\alpha_1, \beta_1, \gamma_1$.

Полагая теперь $\varphi_0(t) = t, \varphi_1(t) = 1, \varphi_2(t) = -tx(t)$, $\rho_i = \frac{1}{\gamma_1 t_i + 1}.$ и снова решая эту задачу, получаем $\alpha_2, \beta_2, \gamma_2$. Продолжая этот процесс при $\varphi_0(t) = t, \varphi_1(t) = 1, \varphi_2(t) = -tx(t)$, $\rho_i = \frac{1}{\gamma_2 t_i + 1},$ получим следующее приближение значений α, β, γ. Итерацию будем продолжать до тех пор, пока не выполнятся соотношения

$$\begin{cases} |\alpha_k - \alpha_{k-1}| < \varepsilon, \\ |\beta_k - \beta_{k-1}| < \varepsilon, \\ |\gamma_k - \gamma_{k-1}| < \varepsilon, \end{cases}$$

где ε - заданная погрешность.

Естественно, все множество используемых регрессионных моделей не исчерпывается дробно-линейными функциями, достаточно часто используются степенные и показательные функции.

4. Будем искать приближающую функцию в виде $x = \alpha t^{\beta}$. Для всех $i = 1,2,...,n$ положим $\delta_i = y_i - \alpha t_i^{\beta}$ и

$$\Delta_i = \ln x_i - \ln \alpha - \beta \ln t_i = \ln \frac{x_i}{\alpha t_i^{\beta}}.$$

Как и ранее, установим связь между этими величинами. Из первого равенства найдем $\alpha t_i^{\beta} = x_i - \delta_i$ и подставим во второе

$$\Delta_i = \ln \frac{x_i}{x_i - \delta_i}.$$

Отсюда $x_i - \delta_i = x_i \exp(-\Delta_i)$ при малых Δ_i можем записать $\delta_i = x_i\left(1 - \exp(-\Delta_i)\right) \approx x_i \Delta_i$.

Таким образом, задачу минимизации величины $\sum_{i=1}^{n} \delta_i^2$ можно заменить задачей минимизации величины

$$\sum_{i=1}^{n} (x_i \Delta_i)^2 = \sum_{i=1}^{n} (\ln x_i - \ln \alpha - \beta \ln t_i)^2 x_i^2,$$

которая является задачей (2.2).

При этом погрешность приближения будет иметь вид

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(x_i - \alpha t_i^{\beta}\right)^2}.$$

5. Пусть приближающая функция задана в виде $x = \alpha \beta^{t}$. Для всех $i = 1,2,...,n$ положим $\delta_i = x_i - \alpha \beta^{t_i}$ и $\Delta_i = \ln x_i - \ln \alpha - t_i \ln \beta = \ln \frac{x_i}{\alpha \beta^{t_i}}$.

Найдем связь между этими величинами. Выражая из первого равенства $\alpha \beta^{t_i} = x_i - \delta_i$ и подставляя во второе, получаем

$$\Delta_i = \ln \frac{x_i}{x_i - \delta_i}.$$

Следовательно, $\delta_i = x_i\left(1 - \exp(-\Delta_i)\right) \approx x_i \Delta_i$. Таким образом, по аналогии, задачу минимизации величины $\sum_{i=1}^{n} \delta_i^{\,2}$ можно заменить задачей минимизации величины $\sum_{i=1}^{n} (x_i \Delta_i)^2 = \sum_{i=1}^{n} (\ln x_i - \ln \alpha - t_i \ln \beta)^2 x_i^2$, которая является задачей (2.2). Погрешность приближения при этом будет иметь вид $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(x_i - \alpha \beta^{t_i}\right)^2}$.

2.1.2. Метод главных компонент

(Principle Component Analysis)

В рассмотренных ранее случаях нам был известен вид регрессионной модели и требовалось определить те или иные количественные характеристики, определяющие эту модель. А что, если нет информации ни о качественных, ни о количественных характеристиках регрессии? Как быть в этом случае? Эффективным методом решения такого рода задач является метод главных компонент (Principle Component Analysis - PCA).

Метод главных компонент (МГК) - один из основных способов уменьшения размерности данных с потерей минимального количества информации, разработан Карлом Пирсоном (Karl Pearson) в 1901 г. Применяется во многих областях, таких как распознавание образов, компьютерное зрение, сжатие данных и т. п. Нахождение главных компонент сводится к вычислению собственных векторов и собственных значений ковариационной матрицы исходных данных. Иногда метод главных компонент называют преобразованием Кархунена-Лоэва (Karhunen-Loeve, часто переводят Карунена-Лоэва) или преобразованием Хотеллинга (Hotelling transform) (см., например, [23],[24]).

Перейдем к рассмотрению метода МГК (PCA). Вначале проведем центрирование данных (удаление тренда), то есть, переопределим исходные значения $x_{new} = x_{old} - \mu$, вычтя из каждого значение математического ожидания

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \ .$$

Ясно, что новые данные будут иметь математическое ожидание равное нулю:

$$E\big(X - E(X)\big) = E(X) - E(X) = 0.$$

По сути был осуществлён параллельный перенос в существующей системе координат.

Требуется найти наиболее точное представление данных $D = \{x_1,...,x_n\}$ в некотором подпространстве *W*, которое имеет размерность $k < n$.

Пусть $\{e_1,...,e_k\}$ ортонормированный базис *W*. Любой вектор из *W* может быть записан в виде $\sum_{i=1}^{k} \alpha_i e_i$, поэтому x_1 можно поставить в соответствие некоторый вектор $\sum_{i=1}^{k} \alpha_{1,i} e_i$ из *W*. При этом ошибка между ними вычислится следующим образом

$$\varepsilon_1 = \left\| x_1 - \sum_{i=1}^{k} \alpha_{1,i} e_i \right\|_2^2 = \left\langle x_1 - \sum_{i=1}^{k} \alpha_{1,i} e_i , x_1 - \sum_{i=1}^{k} \alpha_{1,i} e_i \right\rangle.$$

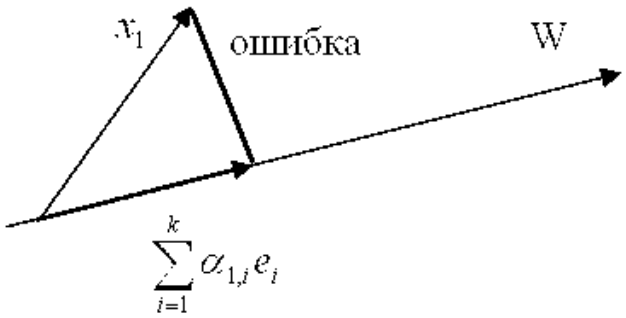


Рис. 2.3- Иллюстрация ошибки восстановления вектора

Чтобы найти полную ошибку, нам необходимо просуммировать величины ошибок по всем x_j, поэтому полная ошибка равна

$$\varepsilon\underbrace{\left(e_1,...,e_k,\alpha_{1,1},...,\alpha_{n,k}\right)}_{unknowns}=\sum_{j=1}^{n}\varepsilon_j^2=\sum_{j=1}^{n}\left\|x_j-\sum_{i=1}^{k}\alpha_{j,i}e_i\right\|_2^2. \qquad (2.7)$$

Для минимизации ошибки, необходимо определить частные производные и учесть ограничения на ортогональность $\{e_1,...,e_k\}$. Вначале упростим соотношение (2.7)

$$\varepsilon\left(e_1,...,e_k,\alpha_{1,1},...,\alpha_{n,k}\right)=\sum_{j=1}^{n}\left\|x_j-\sum_{i=1}^{k}\alpha_{j,i}e_i\right\|_2^2=$$
$$=\sum_{j=1}^{n}\left\|x_j\right\|_2^2-2\sum_{j=1}^{n}x_j^T\sum_{i=1}^{k}\alpha_{j,i}e_i+\sum_{j=1}^{n}\sum_{i=1}^{k}\alpha_{j,i}^2=$$
$$=\sum_{j=1}^{n}\left\|x_j\right\|_2^2-2\sum_{j=1}^{n}\sum_{i=1}^{k}\alpha_{j,i}x_j^Te_i+\sum_{j=1}^{n}\sum_{i=1}^{k}\alpha_{j,i}^2.$$

Тогда

$$\frac{\partial}{\partial\alpha_{m,l}}\varepsilon\left(e_1,...,e_k,\alpha_{1,1},...,\alpha_{n,k}\right)=-2x_m^Te_l+2\alpha_{m,l}.$$

Необходимое и достаточное условие (выполняется только для квадратичной функции) существования экстремума будет иметь вид

$-2x_m^Te_l+2\alpha_{m,l}=0\Rightarrow\alpha_{m,l}=2x_m^Te_l.$

Таким образом, ошибка (2.7) примет вид

$$\varepsilon\left(e_1,...,e_k\right)=\sum_{j=1}^{n}\left\|x_j\right\|_2^2-2\sum_{j=1}^{n}\sum_{i=1}^{k}\left(x_j^Te_i\right)x_j^Te_i+\sum_{j=1}^{n}\sum_{i=1}^{k}\left(x_j^Te_i\right)^2.$$

Упрощая это соотношение, получаем

$$\varepsilon\left(e_1,...,e_k\right)=\sum_{j=1}^{n}\left\|x_j\right\|_2^2-\sum_{j=1}^{n}\sum_{i=1}^{k}\left(x_j^Te_i\right)^2. \qquad (2.8)$$

Замечая, что $\left(a^Tb\right)^2=\left(a^Tb\right)\left(a^Tb\right)=\left(b^Ta\right)\left(a^Tb\right)=b^T\left(aa^T\right)b$, имеем

$$\varepsilon\left(e_1,...,e_k\right)=\sum_{j=1}^{n}\left\|x_j\right\|_2^2-\sum_{i=1}^{k}e_i^T\left(\sum_{j=1}^{n}\left(x_jx_j^T\right)\right)e_i=\sum_{j=1}^{n}\left\|x_j\right\|_2^2-\sum_{i=1}^{k}e_i^TSe_i,$$

где $S=\sum_{j=1}^{n}\left(x_jx_j^T\right)$ является ковариационной матрицей.

Следующим шагом требуется минимизировать функцию ошибки $\varepsilon(e_1,...,e_k)=\sum_{j=1}^{n}\|x_j\|_2^2-\sum_{i=1}^{k}e_i^T Se_i$ при условии $e_i^T e_i=1$ для всех *i*. Используя метод неопределенных множителей Лагранжа (Lagrange), введем множители $\lambda_1,...,\lambda_k$ и, замечая, что $\sum_{j=1}^{n}\|x_j\|_2^2\equiv Const$, выпишем функцию цели (Лагранжиан)

$$\ell(e_1,...,e_k)=\sum_{i=1}^{k}e_i^T Se_i-\sum_{i=1}^{k}\lambda_i\left(e_i^T e_i-1\right)$$

При этом, поскольку градиент функции $f(X)=f(x_1,...,x_m)$:

$$\frac{d}{dX}f(X)=\begin{bmatrix}\frac{\partial f}{\partial x_1}\\ \vdots \\ \frac{\partial f}{\partial x_m}\end{bmatrix},$$

и: $\frac{d}{dX}\left(X^T X\right)=2X$, а при симметрической матрице A: $\frac{d}{dX}\left(X^T AX\right)=2AX$, имеем

$$\frac{\partial}{\partial e_m}\ell(e_1,...,e_k)=2Se_m-2\lambda_m e_m=0\,.$$

Отсюда следует $Se_m=\lambda_m e_m$, что эквивалентно тому, что λ_m и e_m есть соответственно **собственные значения** и **собственные векторы** ковариационной матрицы *S*. При этом ошибка приобретает вид

$$\varepsilon(e_1,...,e_k)=\sum_{j=1}^{n}\|x_j\|_2^2-\sum_{i=1}^{k}\lambda_i\|e_i\|_2^2=\sum_{j=1}^{n}\|x_j\|_2^2-\sum_{i=1}^{k}\lambda_i\,. \qquad (2.9)$$

В свою очередь, минимизация (функция ошибки) (2.9) состоит в выборе базиса *W* из *k* собственных векторов матрицы *S*, которые соответствуют *k* наибольшим собственным значениям. Большее собственное значение *S* дает большую вариацию в направлении соответственного собственного вектора. Этот результат можно переформулировать следующим образом – проекция *X* на подпространство размерности *k*, которая обеспечивает наибольшую вариацию. Таким образом МГК может трактоваться следующим образом - берем ортогональный базис и вращаем его пока на одном из направлений не получим максимальную вариацию. Фиксируем это направление и вращаем остальные, пока не найдем второе направление и так далее.

Пусть $\{e_1,...,e_n\}$ все собственные векторы матрицы *S*, просортированы в порядке уменьшения соответственного собственного значения, тогда для любого

$$x_i=\sum_{j=1}^{n}\alpha_{i,j}e_j=\underbrace{\alpha_{i,1}e_1+...+\alpha_{i,k}e_k}_{approximation}+\overbrace{\alpha_{i,k+1}e_{k+1}+...+\alpha_{i,n}e_n}^{error}$$

коэффициенты $\alpha_{m,l}=2x_m^T e_l$ являются координатами главных компонент, и чем больше значение *k*, тем лучше аппроксимация. При этом главные ком-

поненты располагаются в порядке значимости, – более важные имеют меньший номер.

Приведем алгоритм МГК .

Пусть известны экспериментальные данные $D = \{x_1,...,x_n\}$, и каждый из этих векторов имеет размерность N:

1. Найдем среднее $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$.
2. Вычтем среднее из каждого вектора $z_i = x_i - \mu$.
3. Найдем ковариационную матрицу $S = \sum_{j=1}^{n} z_j z_j^T$.
4. Вычислим собственные векторы $\{e_1,...,e_k\}$, соответствующие k наибольшим собственным значениям S.
5. Пусть $\{e_1,...,e_k\}$ составляют матрицу $E = [e_1...e_k]$.
6. Тогда самой близкой аппроксимацией к x является $y = E^T z$.

Рассмотрим пример.

Пусть дано множество точек D, заданных таблицей:

Таблица 2.2 – Исходные данные

x	1	2	3	4	5	6	7	8
y	2	3	2	4	4	7	6	7

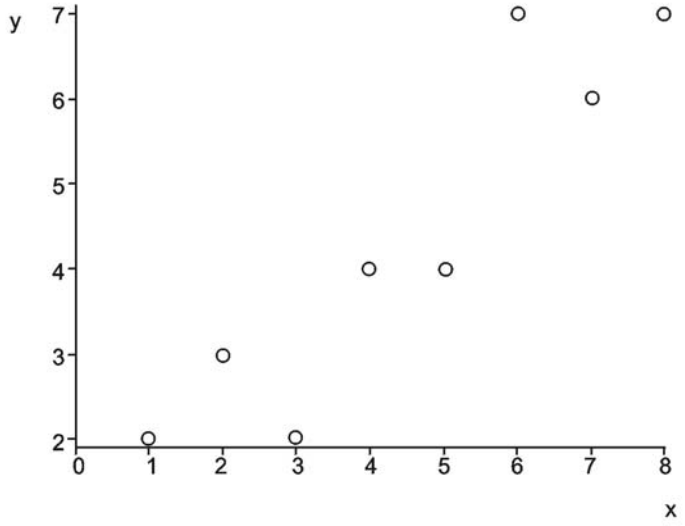


Рис. 2.4 - Исходные данные

Найдем среднее значение $\mu = (4.5,4.375)$, тогда после центрирования данные $\hat{D}$ примут вид

Таблица 2.3 – Центрированные данные

$\hat{x}$	-3,5	-2,5	-1,5	-0,5	0,5	1,5	2,5	3,5
$\hat{y}$	-2,375	-1,375	-2,375	-0,375	-0,375	2,625	1,625	2,625

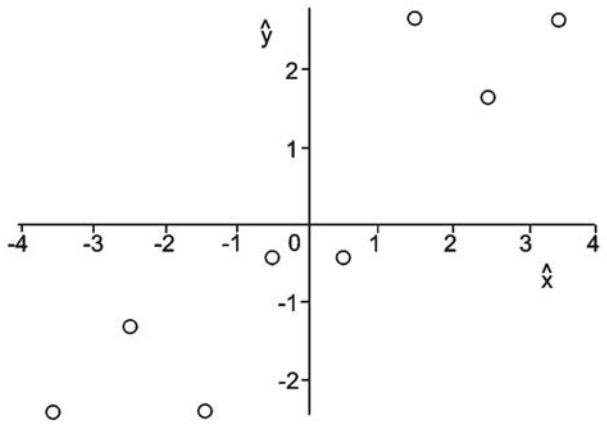


Рис. 2.5 - Параллельный сдвиг, совмещающий начало координат с математическим ожиданием

Тогда

$$s_{1,1} = \langle \hat{x}, \hat{x} \rangle = \sum_{i=1}^{8} \hat{x}_i \hat{x}_i = 42,$$

$$s_{2,1} = s_{1,2} = \langle \hat{x}, \hat{y} \rangle = \sum_{i=1}^{8} \hat{x}_i \hat{y}_i = 32{,}5,\ s_{2,2} = \langle \hat{y}, \hat{y} \rangle = \sum_{i=1}^{8} \hat{y}_i \hat{y}_i = 29{,}875,$$

и ковариационная матрица будет иметь вид

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} \\ s_{2,1} & s_{2,2} \end{pmatrix} = \begin{pmatrix} 42 & 32{,}5 \\ 32{,}5 & 29{,}875 \end{pmatrix}.$$

Решая уравнение

$$\begin{vmatrix} 42 - \lambda & 32{,}5 \\ 32{,}5 & 29{,}875 - \lambda \end{vmatrix} = 0 \Leftrightarrow (42 - \lambda)(29{,}875 - \lambda) - (32{,}5)^2 = 0,$$

получаем собственные значения $\lambda_1 = 68{,}998, \lambda_2 = 2{,}877.$

Для определения собственных векторов найдем любое нетривиальное решение системы

$$\begin{cases} (s_{1,1} - \lambda_1) X_1 + s_{1,2} X_2 = 0, \\ s_{1,2} X_1 + (s_{2,2} - \lambda_1) X_2 = 0, \end{cases}$$

например, $X_1 = 1, X_2 = 0{,}831,$ и, соответственно, системы

$$\begin{cases} (s_{1,1} - \lambda_2) X_1 + s_{1,2} X_2 = 0, \\ s_{1,2} X_1 + (s_{2,2} - \lambda_2) X_2 = 0, \end{cases}$$

например, $X_1 = 1, X_2 = -1{,}204.$

Таким образом, вектор $e_1 = (1, 0.831)^{\mathrm{T}}$, соответствует собственному значению $\lambda_1 = 68.998$, а значению $\lambda_2 = 2.877$ соответствует вектор $e_2 = (1, -1.203787987)^{\mathrm{T}}$. Большему собственному значению соответствует более главное направление. Нормируя собственные векторы единицей, получаем $e_1 = (0.769, 0.639)^{\mathrm{T}}$ и $e_2 = (0.639, -0.769)^{\mathrm{T}}$.

Остается выписать главную компоненту $Y_1 = e_1^T \hat{D}^T$:

Таблица 2.3 – Главная компонента

Y_1	-4,210	-2,802	-2,671	-0,624	0,145	2,831	2,961	4,370

Соответственно, вторая компонента $Y_2 = e_2^T \hat{D}^T$ будет иметь вид:

Таблица 2.4 –Вторая главная компонента

Y_2	-0,410	-0,540	0,868	-0,031	0,608	-1,06	0,347	0,217

Заметим, что для получения приближения исходных данных (нецентрированных) нужно прибавить соответствующее среднее значение.

Восстановление данных одной главной компонентой (то есть проекциями исходных данных на главное направление) будет иметь вид $x_i = e_{1,1} Y_{1,i} + \mu_1$, $y_i = e_{1,2} Y_{1,i} + \mu_2$:

Таблица 2.5 – Исходные данные, восстановленные по первой главной компоненте

$0{,}769 \times Y_1 + 4{,}5$	1,262	2,345	2,445	4,020	4,612	6,678	6,778	7,861
$0{,}639 \times Y_1 + 4{,}375$	1,685	2,585	2,668	3,976	4,468	6,184	6,267	7,417

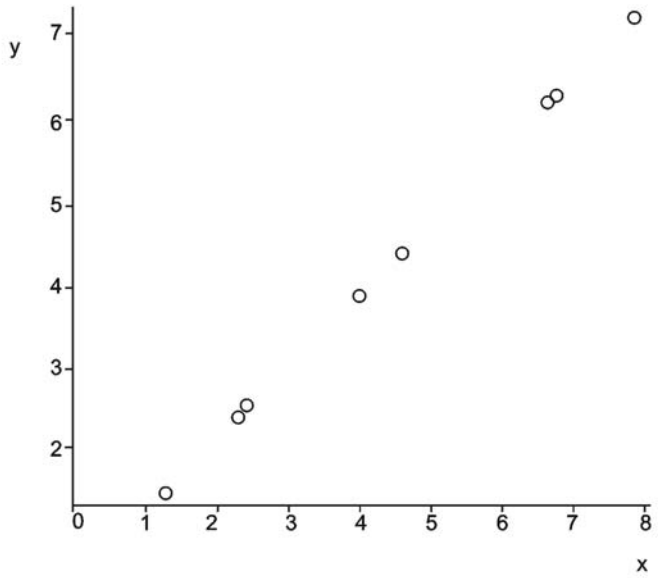


Рис. 2.6 - Представление данных одной главной компонентой

Итерационный алгоритм вычисления главных компонент

Описанный метод определения главных компонент является достаточно ресурсоемким и неустойчивым, особенно в случае, если собственные значения матрицы близки к нулю.

Более эффективным является использование итерационного метода определения главных компонент. Для этой цели рассмотрим задачу (2.1) с другой точки зрения.

Для случая *i=1* задача (2.1) сводится к определению одной компоненты e_1, которая наилучшим образом восстанавливает все исходные данные $\{x_1,...,x_n\}$

$$\varepsilon\left(e_1,\alpha_{1,1},...,\alpha_{n,1}\right)=\sum_{j=1}^{n}\left\|x_j-\alpha_{j,1}e_1\right\|_2^2\to\min \tag{2.10}$$

по всем e_1 и $\{\alpha_{i,1}\}_{i=1}^{n}$ при условии $\sum_{i=1}^{n}\alpha_{i,1}^2=1.$

Если $\{\widetilde{\alpha}_{i,1}\}_{i=1}^{n}$ и $\widetilde{e}_1$ есть решение этой задачи и $\Delta x_j=x_j-\widetilde{\alpha}_{j,1}\widetilde{e}_1$ - ошибка восстановления данных одной первой главной компонентой, то решая задачу

$$\sum_{j=1}^{n}\left\|\Delta x_j-\alpha_{j,2}e_2\right\|_2^2\to\min$$

по всем e_2 и $\{\alpha_{i,2}\}_{i=1}^{n}$ при условии $\sum_{i=1}^{n}\alpha_{i,1}^2=1$, получаем вторую главную компоненту $\widetilde{e}_2$ и соответствующий вектор $\{\widetilde{\alpha}_{i,2}\}_{i=1}^{n}$ и т.д.

При фиксированных $\{\alpha_{i,1}\}_{i=1}^{n}$ задача (2.10) решается методом наименьших квадратов. В силу того, что функция цели представляет собой квадратичный функционал, необходимое и достаточное условия экстремума совпадают. Таким образом, решение задачи сводится к поиску решения уравнения

$$\frac{\partial}{\partial e_1}\varepsilon\left(e_1,\alpha_{1,1},...,\alpha_{n,1}\right)=-2\sum_{j=1}^{n}\left(x_j-\alpha_{j,1}e_1\right)\alpha_{j,1}=-2\left(\sum_{j=1}^{n}x_j\alpha_{j,1}-\sum_{j=1}^{n}\alpha_{j,1}^2e_1\right).$$

Отсюда получаем

$$e_1=\frac{\sum_{j=1}^{n}x_j\alpha_{j,1}}{\sum_{j=1}^{n}\alpha_{j,1}^2},$$

учитывая условие нормирования единицей, имеем $e_1=\sum_{j=1}^{n}x_j\alpha_{j,1}$.

Следующий шаг будем делать исходя из предположения, что в задаче (2.10) нам известна компонента e_1 и требуется найти экстремум по $\{\alpha_{i,1}\}_{i=1}^{n}$

$$\frac{\partial}{\partial\alpha_{\nu,1}}\varepsilon\left(e_1,\alpha_{1,1},...,\alpha_{n,1}\right)=-2\left(x_\nu-\alpha_{\nu,1}e_1\right)=-2\left(\langle x_\nu,e_1\rangle-\alpha_{\nu,1}\langle e_1,e_1\rangle\right)=0,$$

то есть

$$\alpha_{\nu,1}=\frac{\langle x_\nu,e_1\rangle}{\langle e_1,e_1\rangle},$$

где, как обычно, $\langle x,y\rangle$ - скалярное произведение векторов x и y.

Далее, считая найденные $\{\alpha_{i,1}\}_{i=1}^{n}$ известными, повторяем весь процесс, пока не наступит стабилизация ошибки. Полученные e_1 будем считать первой главной компонентой $\tilde{e}_1$.

Применяя этот алгоритм к ошибке восстановления Δx_j, находим вторую главную компоненту e_2 вместе с коэффициентами $\alpha_{j,2}$,и т.д.

Приведем алгоритм.

Вначале центрируем данные, вычитая из исходных данных среднее значение и в дальнейшем считаем, что данные в среднем равны нулю.

1. Положим $\nu = 1$.
2. Выбираем стартовые значения $\{\alpha_{i,1}^{\nu}\}_{i=1}^{n}$, например, $\alpha_{i,1}^{\nu} = \frac{1}{\sqrt{n}}, i = 1,2,...,n.$
3. Вычисляем $e_1^{\nu} = \sum_{j=1}^{n} x_j \alpha_{j,1}^{\nu}$.
4. Далее находим $\beta_i = \frac{\langle x_i, e_1^{\nu} \rangle}{\langle e_1^{\nu}, e_1^{\nu} \rangle}$, и, нормируя, получаем $\alpha_{i,1}^{\nu+1} = \frac{\beta_i}{\sqrt{\sum_{j=1}^{n} \beta_j^2}}.$
5. Полагаем $\nu = \nu + 1$.
6. Проводим проверку критерия остановки, в качестве этого может быть либо стабилизация коэффициентов $\{\alpha_{i,1}^{\nu}\}_{i=1}^{n}$, либо стабилизация главной компоненты e_1^{ν}, либо проверка на заранее заданное фиксированное число итераций. Если условие окончания итерационного процесса выполнено, то переходим к пункту 3.

Проиллюстрируем итерационный алгоритм поиска главных компонент на том же примере, приведенном выше.

Для уже отцентрированных данных (см. таблицу 2.2) проведем несколько итераций. Итак, пусть вначале $\nu = 1$ и $\alpha_{i,1}^{1} = \frac{1}{\sqrt{2}}, i = 1,2.$ Вычисляя $e_{1,j}^{1} = \alpha_{1,1}^{1} \hat{x}_j + \alpha_{2,1}^{1} \hat{y}_j$, получаем:

Таблица 2.6 – Первое приближение главной компоненты

e_1^1	-4.154	-2,740	-2,740	-0,619	0,088	2,917	2,917	4,33

Далее вычислим $\beta_i = \frac{\langle \hat{x}_i, e_1^1 \rangle}{\langle e_1^1, e_1^1 \rangle} = (0.770, 0.644)$ и после нормировки получаем

$$\alpha_{i,1}^2 = \frac{\beta_i}{\sqrt{\beta_1^2 + \beta_2^2}} = (0.770, 0.643).$$

Таким образом, после первой итерации приближенные значения исходных данных будут равны $\tilde{x}_i = \alpha_{1,1}^2 e_{1,i}^1 + \mu_1$, $\tilde{y}_i = \alpha_{2,1}^2 e_{1,i}^1 + \mu_2$ (результаты сравните с таблицей 2.5):

Таблица 2.7 – Исходные данные, восстановленные по первому приближению главной компоненты

$\tilde{x}$	1,315	2,399	2,399	4,026	4,568	6,736	6,736	7,82
$\tilde{y}$	1,702	2,612	2,612	3,977	4,432	6,252	6,252	7,172

После десяти итераций получаем $\tilde{x}_i = \alpha_{1,1}^{11} e_{1,i}^{10} + \mu_1$, $\tilde{y}_i = \alpha_{2,1}^{11} e_{1,i}^{10} + \mu_2$ (результаты сравните с таблицей 2.5):

Таблица 2.8 – Исходные данные, восстановленные по десятой итерации приближения главной компоненты

$\tilde{x}$	1,262	2,345	2,445	4,02	4,612	6,678	6,778	7,861
$\tilde{y}$	1,685	2,585	2,668	3,976	4,468	6,184	6,267	7,167

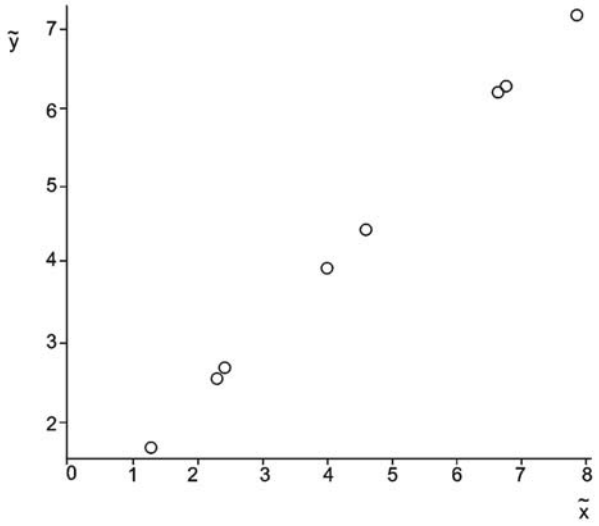


Рис. 2.7. Представление данных, восстановленных по десятой итерации приближения главной компоненты

2.2. Мягкие вычисления в обработке данных

2.2.1. Введение в мягкие вычисления

Термин «мягкие вычисления» (soft computing) ввел в 1994 году основоположник направления нечеткой логики – Лотфи Заде [79]. Под этим термином подразумевалась совокупность эмпирических, нечетких, приближенных методов решения задач, не имеющих точных алгоритмов ре-

шения за полиномиальное время (время работы которых полиномиально зависит от размера входных данных) [79,82]. В настоящее время, к направлениям, где требуется применение мягких вычислений, принято относить [28-30, 113, 123, 143, 145]:

- Нейронные сети;
- Эволюционные стратегии;
- Нечеткая логика;
- Многоагентные (*мультиагентные*) системы (интеллект стаи);
- Различные эвристические алгоритмы поиска решений;
- Динамический хаос и др.

Многие алгоритмы, используемые в данных направлениях, относят к разряду интеллектуальных.

Замечание. *Под искусственным (вычислительным) интеллектом (по одному из возможных определений) понимается концепция, позволяющая компьютерам или иным системам делать такие вещи, которые у людей выглядят разумно.*

Понятие «интеллект» подспудно предполагает свойство человека или некоторой системы решать интеллектуальные задачи. При этом под интеллектуальной задачей будем понимать такую задачу, для решения которой еще не придумано чёткого алгоритма и требуется применение именно интеллектуальных усилий естественного или искусственного разума. Другими словами, под интеллектом будем понимать наличие совокупности свойств у человека или некоторой системы (компьютера, киборга, робота), позволяющее создавать новое, творить. В частности, новые алгоритмы обработки данных.

Методы, использующие мягкие вычисления, обладают высокой универсальностью.

Другим важным общим свойством мягких вычислений является *свойство адаптивности* – подстройки под задачу в текущем времени ее решения. Это обязательно требует внесения в процедуру решения задачи этапа обучения или самообучения (который автоматически присущ интеллектуальной деятельности), что уменьшает для исследователя сложность задачи. Теперь он может работать с задачей как с черным или серым ящиком (система с возможностью ограниченной настройки пользователем), не вникая до конца в тонкости работы управляемой системы, в надежде, что эти тонкости будут скомпенсированы «интеллектом» метода.

Следствием универсальности мягких вычислений, являются также их *хорошие комбинаторные способности*.

В настоящее время применяются интеллектуальные системы с мягкими эволюционными вычислениями, где нейронные сети обучаются при помощи генетических алгоритмов [56], а нечеткие (fuzzy) или нейронечеткие (neuro-fuzzy) классификаторы, компрессоры, эмуляторы, предикторы и т.п., в свою очередь, являются основой многоагентных *(мультиагентных)* систем обработки данных [20,28,36,44,56,95,116,143].

2.2.2. Эволюционные вычисления

Краткая история

Естественная эволюция, которую с исторической точки зрения, можно рассматривать как продукт применения эволюционных вычислений, привела, по мнению большинства ученых, за длительный период, исчисляемый миллиардами лет, к появлению *Homo Sapience (человека мыслящего),* т.е. к индивиду, обладающему высокоразвитым *естественным интеллектом*.

Первым ученым, предложившим стройную *эволюционную* теорию биоценоза, явился Чарльз Дарвин. В своей работе «Происхождение видов» в 1859 году он показал, что основой эволюции являются следующие процессы:

- Изменчивость – новые особи популяции практически всегда немного (а иногда сильно) отличаются от своих родителей.
- Отбор – естественный или искусственный отбор отсеивают неудачные варианты изменений. Жить остаются только наиболее удачные (более приспособленные).
- Наследственность – изменения, произошедшие в одном из поколений, наследуются потомками.

Совокупность этих движущих сил и позволяет видам приспосабливаться к изменяющейся окружающей среде, совершенствоваться и выживать. Однако во времена Ч. Дарвина был понятен только механизм отбора. Механизм появления и закрепления новых информативных признаков и как они передаются потомкам, стал понятен гораздо позже. В 1944 году, О. Эйвери, К. Маклеод и М. Маккарти опубликовали результаты своих исследований. В них доказывалось, что за наследственные процессы в организмах отвечает ДНК (дезоксирибонуклеиновая кислота). Структура этой кислоты в виде двухцепочечной спирали стала известна еще позже – 27 апреля 1953 года в журнале «Nature» была опубликована знаменитая статья Уотсона и Крика. С тех пор началось и продолжается изучение механизма функционирования ДНК. Узнаются все новые и новые детали о её структуре, например о том, что разные участки ДНК мутируют с разной частотой, что многие гены могут присутствовать в ДНК в разном количестве экземпляров (copy number variation) и, что от этого может зависеть продукция того или иного белка, чувствительность организма к различным заболеваниям [25], а также то, что структура ДНК является фрактальной [147].

Почти одновременно с работами генетиков-биологов появились работы исследователей, которые хотели повторить аналогичный процесс с использованием вычислительной техники. К сегодняшнему дню накопилось огромное количество различных эволюционных алгоритмов. Вместе с тем, перечисление основных алгоритмов в этом направлении, следует начать с генетических алгоритмов (ГА) и классификационных систем Холланда. Впервые они были опубликованы в начале 60-х годов, но основное свое распространение получили после выхода книги, ставшей классикой – «Адаптация в естественных и искусственных системах» [148]. На террито-

рии бывшего СССР также работали в данном направлении, несмотря на то, что в то время по идеологическим соображениям был введен лозунг: «Генетика и кибернетика – «продажные девки» империализма».

Как было показано позднее, именно подход, основанный на эволюционных вычислениях, объединял эти два направления. Выдающимся отечественным ученым - Л.А. Растригиным, в 70-е годы в рамках теории случайного поиска был предложен ряд алгоритмов, моделировавших различные стороны поведения живых организмов. Эти идеи получили дальнейшее развитие в посвященных эволюционному моделированию работах И. Л. Букатовой. Ю. И. Неймарком было предложено осуществлять поиск глобального экстремума на основе множества независимых автоматов, при этом моделировались процессы рождения, развития и смерти особей. Большой вклад в развитие эволюционных вычислений внесли также Уолш и Фогель.

Каждая из этих школ взяла из известных на то время принципов эволюции что-то свое. После этого оно было упрощено до такой степени, что процесс можно было релизовать (промоделировать) на компьютере.

Как и многие эмпирические алгоритмы, эволюционные алгоритмы не гарантируют нахождения наилучшего результата. Также они бывают сложны и в настройке. При неправильных параметрах может проявляться склонность к вырождению, плохая поисковая способность. В этом случае не стоит сразу забрасывать эволюционные алгоритмы – стоит попытаться помочь им своим, естественным интеллектом («поиграться» с настройками, например). Здесь можно вспомнить, что если какого-то вида осталось очень мало – меньше десятка особей, то в природе он обречен на вымирание из-за вырождения. Однако лучше вспоминать чудные по конструкции тела птиц, обтекаемые крылья мант, эхолокаторы дельфинов и летучих мышей, а также черную плесень, использующую энергию излучения разрушенного реактора Чернобыльской АЭС [27].

2.2.3 Генетический алгоритм

Генетический алгоритм (ГА) является одним из самых известных эвристических алгоритмов. Он прост в принципах работы, но имеет большой потенциал для развития, что и будет показано в данном разделе.

По своей сути, ГА является алгоритмом случайного поиска глобального оптимума многоэкстремальной функции. Для этого он использует, с определенной степенью приближения, модель размножения живых организмов. Для того чтобы решить проблему, нам нужно представить её в виде так называемой фитнесс-функции от многих переменных (также называемой оценочной):

$$f(x_1, x_2, x_3, \dots, x_N).$$

Для решения задачи необходимо найти глобальный максимум или минимум (это не принципиально, поскольку поиск максимума легко заменяется поиском минимума этой же функции, взятой со знаком минус, и на-

оборот). При этом на значения входных переменных обычно налагаются определенные ограничения, хотя бы по диапазону их изменения.

Перед тем, как мы рассмотрим работу ГА, нам необходимо представить все входные переменные в виде *хромосом*. Под *хромосомами* в ГА подразумеваются цепочки символов, с которыми и производятся дальнейшие операции. Для кодирования параметров чаще всего применяют следующие два метода:

- двоичный формат;
- формат с плавающей запятой.

При использовании двоичного формата, под параметр выделяется N бит (для каждого параметра это N может быть различным). Поскольку для каждого из этих параметров имеются ограничения MIN и MAX, то взаимный переход между значениями параметров в формате с плавающей запятой и их бинарным представлением можно записать в следующем виде:

$$g = (r - MIN) / (MAX - MIN) \cdot (2^N - 1),$$
$$r = g \cdot (MAX - MIN) / (2^N - 1) + MIN,$$

где g – бинарное представление параметра, помещенное в N бит. r – значение параметра в формате с плавающей запятой. Часто используют последующее преобразование полученного бинарного представления в код Грея – это позволяет уменьшить разрушительную силу мутаций (тут мы немного забегаем вперед).

Полученные бинарные представления каждого параметра выкладывают в цепочку (строку) бит, которая дальше называется *хромосомой*.

При работе с параметрами в формате с плавающей запятой, их значения также выкладываются в цепочку бит, но без указанного выше преобразования, прямо в том представлении, с которым работает процессор компьютера.

После того, как закодированы все необходимые параметры в виде хромосом, можно приступить к основному циклу генетического алгоритма:

1. Генерация первоначальной случайной популяции.
2. Генерация следующего поколения.
3. Удаление худших решений во вновь сгенерированном поколении.
4. Если не достигнут критерий окончания, переходим на шаг 2.
5. Окончание работы. Экземпляр, у которого самое лучшее значение фитнесс-функции (*fitness fuction – FF*), является искомым решением.

Ключевым здесь, конечно же, является этап 2. Его можно детализировать следующим образом:

1. Сортируем родительское поколение в соответствии со значением *FF* для каждого экземпляра.
2. Пока не сгенерировано достаточное количество экземпляров новых поколений:
   2.1. Отбираем двух родителей.
   2.2. Объединяем их хромосомы (*кроссовер*).
   2.3. Применяем другие генетические операторы.

Опишем немного подробнее каждый из шагов генерации.

2.1. Для отбора используются различные стратегии. Например, просто случайным образом выбираем из N самые лучшие экземпляры. Другим распространенным подходом является турнирный. Он заключается в том, что для каждого из родителей выбирается случайная пара (или больше) претендентов. Из них используется тот, у которого значение *FF* лучше. Таким образом, более приспособленные особи чаще будут родителями, но менее приспособленные также имеют шанс пройти.

Для ускорения сходимости также часто используется стратегия элитизма – в следующее поколение решений проходят без изменений самые лучшие из имеющихся решений предыдущего поколения (элита). Но с этим подходом нужно быть крайне осторожным – при недостаточном размере популяции, она очень быстро становится похожей на элиту и поиск новых решений практически прекращается.

2.2. и 2.3 – применение генетических операторов. Основой ГА являются два оператора: *кроссовер*, который объединяет решения родителей, и *мутация*, которая обеспечивает поисковые способности. Кроме этих основных операторов могут также применяться дополнительные операторы, например, инверсия.

Оператор кроссовера работает с битовыми строками двух родительских хромосом. Наиболее простым вариантом является одноточечный кроссовер. В этом случае каждая из родительских хромосом перерезается в одной, случайно выбранной точке. Хромосома потомка формируется из «головы» хромосомы одного предка и «хвоста» второго:

Предок 1: 1001101110101|100110
Предок 2: 0010110010110|010101 → 1001101110101010101

Оператор кроссовера может быть и более сложным – двухточечным, многоточечным (*кроссинговер*), или даже использовать совершенно иные принципы (см. далее). Главное, чтобы он объединял решения предков, и потомку не было необходимости находить удачные решения заново.

Оператор *мутации* – это просто случайное изменение хромосомы в одном или большем количестве бит:

1001101110101100110 → 1001101100101100110

Мутация – разрушительный оператор. В большинстве случаев, он нарушает решение или даже приводит особь к неработоспособному состоянию. Поэтому вероятность его применения не должна быть чрезмерно высокой. Но отсутствие мутаций сводит на нет способность ГА к поиску глобального оптимума во всем пространстве решений.

Приведенный в качестве примера дополнительных операторов, оператор инверсии заключается в циклической перестановке бит в хромосоме случайное количество раз:

1001101110101100110 → 0110100110111010110

Теперь у нас есть все составные части генетического алгоритма, и мы можем закрепить их на практике. В электронном приложении к учеб-

ному пособию, в папке SimpleGA можно найти проект, где реализован простейший ГА. Язык реализации данного примера – C# (https://cmidpbook.codeplex.com/, загрузка программ архивом - https://cmidpbook.codeplex.com/SourceControl/latest#).

Программа является консольной, что позволяет не концентрироваться на интерфейсной части, а сразу же переходить к алгоритмизации. Из-за этого, и в связи с учебной направленностью этого и других примеров, основные настройки задаются в тексте программ в виде констант. На сленге программистов, параметры данной программы являются «захардкожеными». Однако минусом это будет в том случае, если пользователю программы не нужно иметь доступа к коду программы. Мы же обращаемся, скорее к разработчикам, поэтому правку параметров в тексте программы вполне можно считать своеобразным интерфейсом для программиста. Таким образом, в качестве интерфейса пользователя к изменению параметров, можно использовать IDE.

Попробуем найти глобальный экстремум одной из тестовых функций:

$$f(x,y) = \frac{100}{100 \cdot \left(x^2 - y\right)^2 + (x-1)^2 + 1} \quad .$$

Данная функция является овражной с достаточно малым наклоном в районе максимума. Максимум функции равен 100 при значении x=y=1. Конечно же, сделаем вид, что нам это неизвестно, только известно, что максимум находится при значениях параметров (и “x” и “y”) где-то между -1.28 и +1.28.

Применим бинарный способ кодирования генома, без использования кода Грея. Цепочку бит будем хранить в типе int, причем, старшие 16 бит будут отвечать за параметр x, а младшие – за параметр y.

В SimpleGA сосредоточена демонстрация отбора с элитизмом, турнирный выбор предков, кроссовер (одноточечный) и мутация. При тех настройках, которые указаны в примере по умолчанию, будет использован вариант, называемый ГА с элитизмом. В нашем случае, он заключается в том, что один наиболее приспособленный индивидуум гарантированно попадает в новое поколение. Такой подход гарантирует неухудшение показателей самого лучшего решения из поколения в поколение, часто обеспечивает более высокую стартовую скорость поиска решения. В то же время, может мешать более полному исследованию пространства решений и ускорению вырождения популяции. Читатель может самостоятельно поэкспериментировать с разными настройками и понаблюдать, как они влияют на качество и скорость поиска решений.

Запустив программу с приведенными настройками, довольно часто можно получить вот такую картину:

```
…
Поколение 197
Лучшая особь = 99,9999998156313
```

```
Поколение 198
Лучшая особь = 99,9999998156313
Поколение 199
x = 0,999995727473869
y = 0,999995727473869
Геном = E3FFE3FF
Нажмите "Ввод" для выхода . . .
```

Указанные параметры x и y – это наиболее близкий к числу 1.0 узел из множества узлов, которые получаются при разбиении промежутка [-1.28…+1.28] на 2^{16} отрезка. Этот результат можно наблюдать не всегда – мы ведь работаем с вероятностным (точнее, псевдослучайным) процессом, поскольку используем генератор случайных чисел (класс Random). А пространство решений, которые нужно исследовать достаточно велико.

О влиянии генератора случайных чисел на работу ГА

Одним из факторов, которые могут влиять на работу ГА, является способ получения случайных чисел. Приведем иллюстрацию работы ГА для поиска глобального экстремума двух тестовых функций Гривонка и Розенброка, с одновременным проведением исследования влияния на работу ГА различных алгоритмов генерации «псевдослучайных» чисел.

1. Функция Гривонка (Griewangk's function). Пусть n − натуральное число, тогда

$$F\left(x_1, x_2, ..., x_n\right) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos \frac{x_i}{\sqrt{i}}.$$

$$-600 \le x_i \le 600,\ i = 1, 2, ..., n.$$

Функция Гривонка близка по виду на функцию Растригина. Она имеет триллион локальных экстремумов. Однако эти локальные экстремумы регулярно распределены по всей поверхности функции. Глобальный минимум, равный 0, достигается в точке $x_1 = x_2 = ... = x_n = 0$. При $n = 10$ существует еще 4 локальных минимума, которые равны 0,0074, и достигаются приблизительно в точке

$$\left(\pm\pi, \pm\sqrt{2}\pi, 0, ..., 0\right).$$

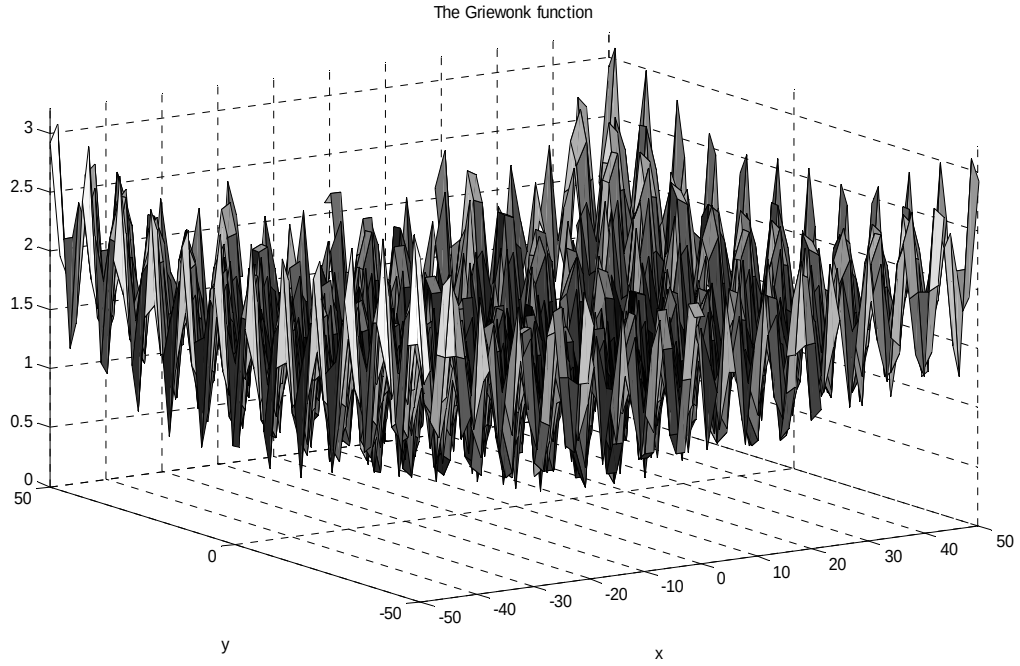


Рисунок 2.8 – График функции Гривонка двух переменных

2. Функция Розенброка (Rosenbrock's saddle или second function of De Jong). Функция Розенброка входит в набор тестовых функций De Jong и носит второе название De Jong 2. Пусть n − натуральное число, тогда

$$f(x)=\sum_{i=1}^{n-1}\left(100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right).$$

Интервал, которому принадлежит переменная: $-5{,}12 < x_i < 5{,}12$, $i=1,2,...,n$.

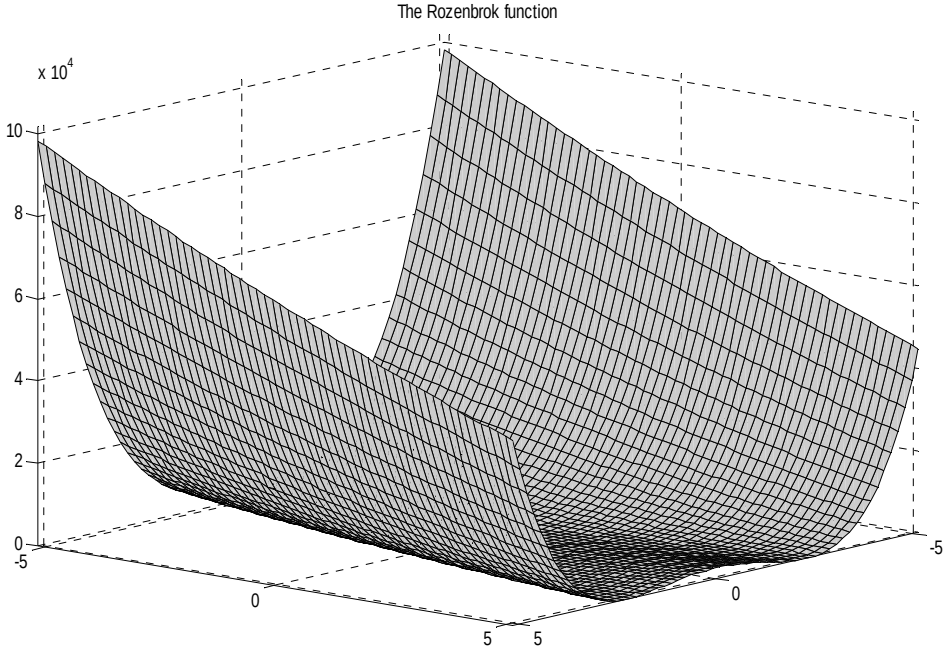


Рисунок 2.9– График функции Розенброка

Данная функция была предложена Ховардом Розенброком в 1960 году. Эта функция имеет крайне пологий изогнутый овраг, который очень усложняет поиск минимума (значения аргументов в точке минимума оче-

видны: $\bar{x}_1 = \ldots = \bar{x}_n = 1$ (при этом $f\left(\bar{x}\right) = 0$)). В случае двух переменных функция создает небольшое плато с двумя «крыльями» по обеим сторонам. Точка минимума $(1,1)$ находится на одной стороне плато, в то время как поисковые алгоритмы часто «застревают» на другой стороне. Благодаря сложным для методов поиска особенностям экстремума функция Розенброка часто используется для тестирования сходимости разных оптимизационных алгоритмов и для их сравнения.

Для исследования влияния на работу ГА алгоритмов генерации «псевдослучайных» чисел в среде Matlab программно реализовано 3 вида RANDUM генераторов [27]

- генератор Мерсена (стандартный);
- линейный конгруэнтный генератор (ЛКГ);
- генератор Фибоначчи с запаздыванием.

Указанные генераторы, а также отдельно исследованный линейный конгруэнтный генератор (Urand) были использованы для работы генетического алгоритма (ГА) [27, 133].

В то же время исследовалось влияние на работу ГА хаотических чисел, для реализации которых использовались генераторы Лоренца с параметрами $\sigma = 10$, $\rho = 28$, $\beta = 8/3$ и Чуа с параметрами $\alpha = 4{,}91667$, $\beta = 3{,}642$ и

$$f\left(x\right) = \begin{cases} -0{,}07x - 1{,}57, & x < -1, \\ -0{,}07x + 1{,}43, & x > 1, \\ 1{,}5x, & x \in [-1,1]. \end{cases}$$

Ниже в таблицах приведены результаты работы ГА на тестовых многоэкстремальных функциях Гривонка и Розенброка. Для каждого из генераторов псевдослучайных чисел, а также для хаотических генераторов было проведено по 100 экспериментов [27]. В 50-ти случаях из них вероятность мутации выбиралась на уровне 0,1, а в остальных 50 – на уровне 0,3. В каждом эксперименте брало участие по 100 индивидуумов, эксперимент длился в течении 100 поколений. Для удобства напомним значения глобального минимума рассмотренных функций:

- функция Гривонка – 0;
- функция Розенброка – 0;

Таблица 2.9 –Минимизация функций ГА со стандартным оператором мутации (коэффициент мутации равен 0,1)

Функции	Генераторы псевдослучайных чисел				Генераторы хаоса	
	Мерсен	ЛКГ	Фибоначчи	Urand	Лоренц	Чуа
Гривонка	0,0123216	0,0098595	**0**	0,252528	0,00739627	0,0297598
Розенброка	0,00221527	**1,54568e-05**	0,000954742	1,08383	0,00109675	**7,19038e-09**

Таблица 2.10 –Минимизация функций ГА со стандартным оператором мутации (коэффициент мутации равен 0,3)

Функции	Генераторы псевдослучайных чисел				Генераторы хаоса	
	Мерсен	ЛКГ	Фибоначчи	Urand	Лоренц	Чуа
Гривонка	**0**	0,00739618	**0**	0,697938	0,00986468	0,0239592
Розенброка	0,0112275	0,00799863	**0,000689653**	1,21357	0,0764429	0,0875793

Пространственный кроссовер

Задачи, подобные тем, что решались в простейшем примере ГА, любят приводить в учебных пособиях по ГА. И это понятно почему – проблема обозрима, ответ известен, легко проверить. В то же время, такие примеры оставляют двойственное впечатление. Ведь их часто нетрудно решить и другими, тоже простыми методами – от случайного перебора (метод Монте-Карло) до градиентных методов типа покоординатной оптимизации. Либо, наоборот, трудно любыми методами.

Возможности ГА раскрываются в примерах, где моделируемая система разделяется на подсистемы, решения в которых могут быть найдены параллельно. Часто для повышения эффективности работы в конкретной задаче, приходится адаптировать классические генетические операторы, изобретать новые [27, 133].

Другим важным моментом является то, что в реальных задачах, зачастую, наиболее трудоемкой частью процесса просчета является не генерация нового поколения, а оценка приспособленности каждой особи. Поэтому некоторое усложнение схемы генерации новых поколений не сильно увеличивает интегральную трудоемкость алгоритма, а вот увеличение «выхода» хороших потомков – очень даже уменьшает её.

Попробуем показать это все на примере решения задачи, приближенной к реальной. Итак, задача:

Военная интерпретация. У нас есть войска противника, расположенные неравномерно на определенной территории, и оружие – 10 боеголовок ядерного оружия. Известно, что при взрыве боеголовки в радиусе поражения KillingRadius, не остается ничего живого. Упрощенно будем считать, что за пределами данного радиуса, боевые единицы противника остаются живыми. Нашей задачей будет найти такие координаты для каждой боеголовки, чтобы у противника после удара осталось как можно меньше войск.

Коммерческая интерпретация. Конечно, если у вас более пацифистские настроения, эту же задачу можно представить и иначе – есть средства на постройку 10 магазинов в определенном районе города. Люди готовы ходить в магазин, расположенный не далее, чем за M (аналог KillingRadius) метров от дома. Люди проживают неравномерно по выбранному району. Необходимо расположить магазины таким образом, чтобы покрыть сервисом наибольшее количество жителей.

В тексте программы, названия идентификаторов выбраны исходя из первой формулировки задачи, но это не будет означать специализации только на первом варианте постановки проблемы.

В качестве входных данных для тестового приложения будут поступать изображения в формате PNG. Данный формат используется, поскольку он сжимает без потери информации, а для определения яркости точек это важно. Боевая единица противника кодируется черной точкой (яркости цветовых компонент в RGB – 0, 0, 0). Территории, накрытые взрывами, будем показывать «томатным» цветом. Такой способ подачи входных данных дает возможность наглядно увидеть полученный результат, а также задействовать при решении графическую карту компьютера, поскольку для оценки полученного результата можно его просто нарисовать: поверх исходного изображения нарисовать залитые окружности с радиусом поражения.

В этот раз программа (папка NukeGA в электронном приложении) представляет собой WinForms-приложение с единственным окном, поскольку довольно любопытно наблюдать за динамикой работы генетического алгоритма. А оценить её только по числам, бегущим в консоли, в данном случае не очень удобно. Различные же параметры приложения, как и в прошлом примере, можно изменить прямо в тексте программы, используя в качестве редактора IDE.

Прежде всего, решим, какой вариант кодирования пространственной информации в хромосомы мы будем использовать. Самый простой вариант – последовательно закодировать координаты в битовую строку так, как это было показано в предыдущем разделе. После этого, применить обычные операторы кроссовера и мутации. Но, практика показывает, что в таких случаях очень часто возникают различные проблемы – проблема конкурирующих решений, слабое качество синтезируемых решений. Более подробно данные проблемы описаны в работе [29]. Там же предложен и один из возможных вариантов решения этих проблем – кроссовер, основанный на пространственном положении точек решения. Далее будем сокращенно его называть пространственным кроссовером.

Итак, будем хранить данные, относящиеся к одной точке в виде неразделяемого набора данных. В биологии такие признаки называются сцепленными из-за того, что передаются потомкам, как правило, вместе. Т.е., в нашем случае это будут координаты на плоскости по осям X и Y.

Что же представляет собой пространственный кроссовер? Как уже было сказано выше, если к списку пространственных координат двух хороших, но разных родителей, применить обычный одно- или двухточечный кроссовер, мы, скорее всего, получим нежизнеспособного потомка. Основных причин тут две. Прежде всего, у разных особей одна и та же точка может храниться в разных местах генома, и, при обмене генетической информацией, мы можем поместить почти рядом геометрически две точки, находящиеся в разных местах генома родителей. При этом оголится какой-то другой участок. Поначалу, генетический алгоритм будет тратить всю

свою мощь на то, чтобы отобрать удачный способ кодирования координат, и, только потом, если к этому моменту еще не произойдет вырождения популяции, начнет собственно поиск удачного решения. Это и есть проблема конкурирующих решений – когда одному и тому же решению может соответствовать множество способов кодирования этого решения, что очень увеличивает разрушающее свойство кроссовера.

Второй проблемой является то, что если обращать внимание только на место, где расположена точка в геноме, то кроссовер будет случайным образом брать точки из одного родительского решения и из другого. Эти точки будут разбросаны по всему пространству поиска. Однако понятно, что часто точки, расположенные вблизи друг от друга, образуют своеобразные подкомплексы решений, которые поддерживают друг друга. И случайное изъятие или добавление точек в такой ансамбль точек существенно нарушает качество решений.

Обе эти проблемы решает пространственный кроссовер. При своей работе он ориентируется на положение точек в пространстве решаемой задачи. На следующем эскизе показана схема объединения двух родительских геномов.

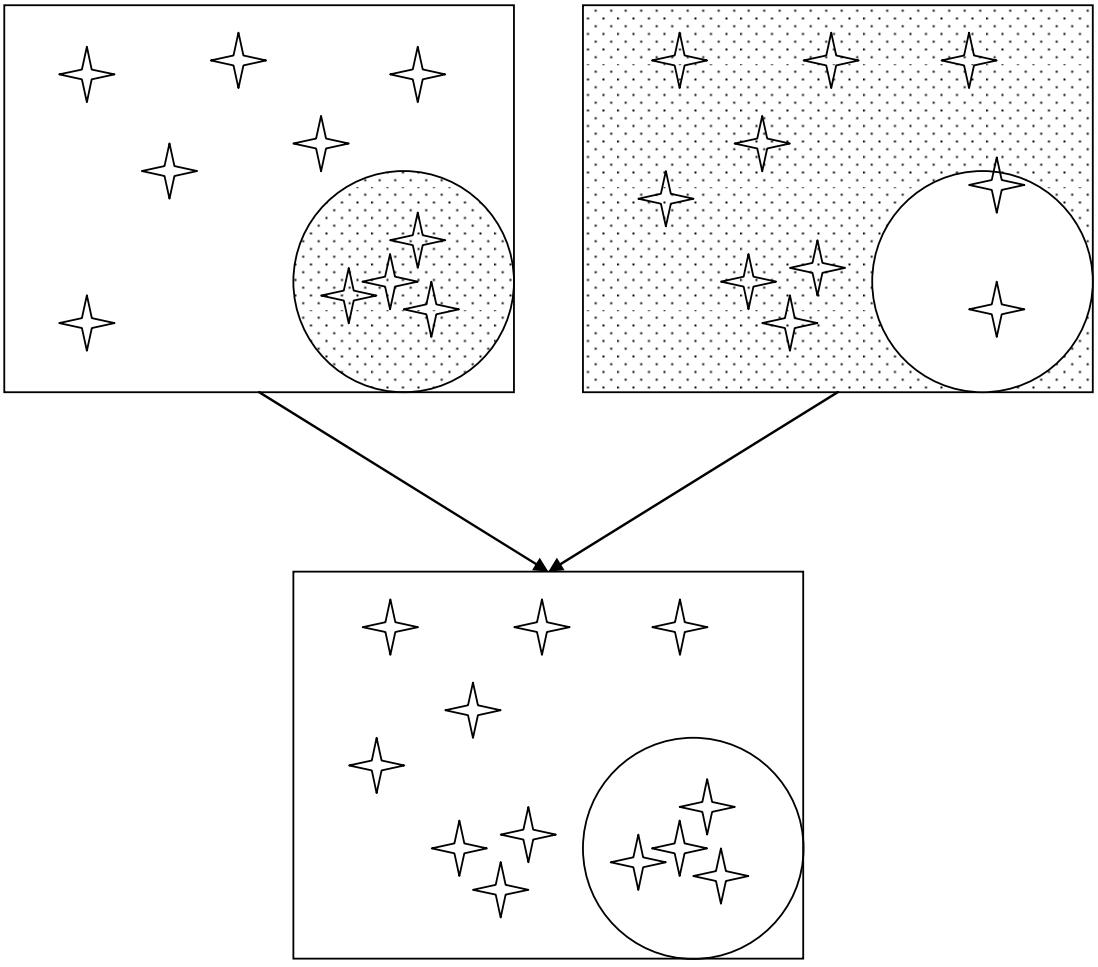


Рис.2.10- Схема пространственного кроссовера

Пространственный кроссовер состоит из следующих шагов:

1. Выбираем в пространстве решаемой задачи произвольную (случайную) точку.

2. У обоих родителей вырезаем одинаковую (по координатам центра и радиусу) окружность (сферу или гиперсферу для пространств с размерностью более 2-х).
3. У одного родителя берем точки, которые лежат внутри окружности, у другого – снаружи.
4. Объединяем взятые точки в одно решение.
5. Корректируем количество точек в потомке – удаляем (случайным образом), если есть лишние, либо добавляем еще не использованные точки родителей.

Такой подход позволяет сохранить уже сложившиеся локальные ансамбли точек с большей вероятностью (по сравнению с обычным кроссовером и способом кодирования).

В примере из электронного приложения стоит отметить также такую особенность, как использование метрики L_1 (метрика улиц) при вычислении расстояния между точками. Принципиально это ничего не меняет, просто из пространства родителей будут вырезаться квадраты (повернутые на 45 градусов), а не окружности.

При первоначальном тестировании примера оказалось, что после первоначального прогресса и нахождения наиболее перспективных конфигураций решений, далее идет крайне медленный процесс адаптации полученных решений, в основном за счет мутации. Однако мутация, в том виде, в котором она была приведена выше, является достаточно грубым средством, – ведь вместо одного взрыва получаем случайным образом другой. И вероятность того, что он лишь слегка подкорректирует предыдущий, очень мала. Поэтому в пример был добавлен еще один оператор, названный степпингом (Stepping). Фактически это тоже мутация, но более мягкая, которая лишь немного смещает взрыв, к которому она применяется.

Исходное изображение (можно найти в списке исходных файлов примера под именем 128x128_2.png) имеет размер 128x128 пикселов. Изначально на изображении 5910 черных пикселов, которые мы и должны максимально «накрыть взрывами».

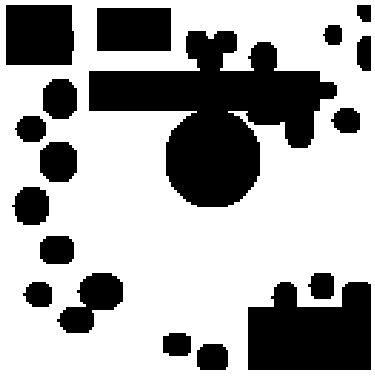


Рис.2.11 - Исходная область для поражения

Радиус области поражения – 12 пикселов, количество взрывов – 10. Размер популяции – 1000 особей. Использовался элитизм, турнирный отбор из 500 лучших особей. Вероятность применения пространственного кроссовера – 0.4, мутации – 0.1, степпинга – 0.3. И, конечно же, следует

иметь в виду, что при каждом запуске результаты будут несколько отличаться, – поскольку рассматриваются вероятностные процессы.

Работа с программой происходит следующим образом: 1) запускается приложение; 2) загружается нужная картинка (кнопка “Loadsample”); 3) запускается эволюция (нажимается кнопка “Start”). Текущий результат эволюции отражается в главном окне программы, а промежуточные результаты записываются в том же каталоге, из которого была загружена картинка (файл с таким же именем и расширением TXT, а также png-картинки).

лучший представитель нулевого (случайного) поколения. Качество 3435.

лучший представитель 5-го поколения. Качество 3170.

лучший представитель 20-го поколения. Качество 2688.

лучший представитель 100-го поколения. Качество 2214.

лучший представитель 500-го поколения. Качество 2153.

Рис.2.12 - Результаты применения эволюционного алгоритма

В приведенном примере мы показали лишь одну из возможных адаптиций ГА к практике. Природа является только первоначальным образцом для подражания, но мы не обязаны копировать все детали. Достаточно соблюдать только базовые принципы и руководствоваться здравым смыслом. Перечислим некоторые из них, позволяющие добиться более быстрой сходимости процесса эволюции, либо улучшить качество исследования предметной области:

Ускорение сходимости решения	**Улучшение качества работы (поисковые способности) ГА**
✓ Увеличение прессинга естественного отбора ✓ Уменьшение количества особей, допускаемых к размножению ✓ Использование элитизма	✓ Уменьшение прессинга естественного отбора ✓ Увеличение объема генетического материала ✓ Диплоидия (также увеличивает количество генетического

✓ Уменьшение общего количества генерируемого потомства ✓ Выполнение алгоритма параллельно на нескольких компьютерах (процессорах)	материала) ✓ Разбиение популяции на части. Это также является и способом легко распараллелить алгоритм

2.2.4 Генетическое программирование

Рассмотренный выше генетический алгоритм, работал с закодированной в генах информацией. Какой именно, в общем-то, не важно. Это вполне могут быть и команды какой-то вычислительной машины – программа, а качество получаемой особи будет зависеть от того, насколько хорошо эта программа выполняет поставленную задачу. Основным минусом будет то, что длина программы будет фиксирована. Да и классические генетические операторы не способствуют высокой вероятности появления работоспособных потомков.

Поэтому для эволюционного составления программ были разработаны иные (по сравнению с ГА) методы хранения генома и других реализации генетических операторов. Хронологически, первой формой, была древообразная форма хранения генома (Рис. 2.13), предложенная Н. Крамером [28, 32] и Дж. Коза[29].

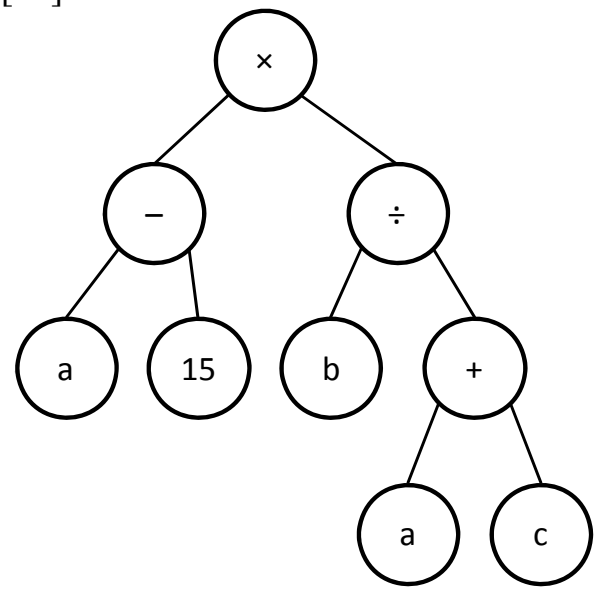


Рис. 2.13 - Алгоритм вычисления значения функции (a-15)(b/(a+c)), представленный в виде дерева

В качестве других форм, которые также часто используются, можно назвать линейную (рис. 2.14) и сетевую (или графовую) формы (2.15).

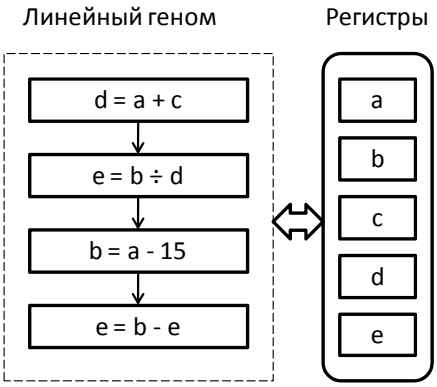


Рис. 2.14 - Алгоритм значения функции (a-15)(b/(a+c)), представленный в линейном виде (код какого-то виртуального процессора)

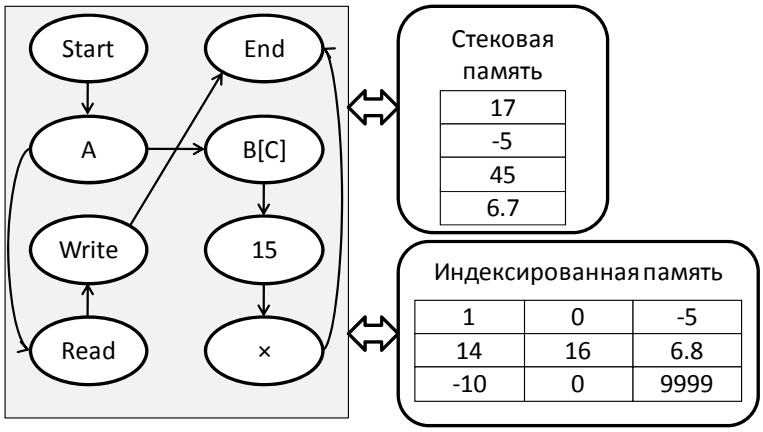


Рис. 2.15 - Геном небольшой программы для виртуального процессора, представленный в сетевом (графовом) виде

Операция мутации, в случае генетического программирования (ГП), похожа на аналогичную операцию в ГА, но при этом добавляется разнообразие в геном. Для линейного генома, например, к случайному изменению произвольной команды, добавляются операции вставки случайной команды или удаления в случайном месте.

В свою очередь, несколько сложнее выглядит и операция обмена генетическим материалом у предков – кроссовер, хотя основной принцип остается тем же – берется часть генома одного родителя и часть второго. Приведем примеры кроссовера для линейного (рис.2.16) и древовидного представления генома (рис.2.17).

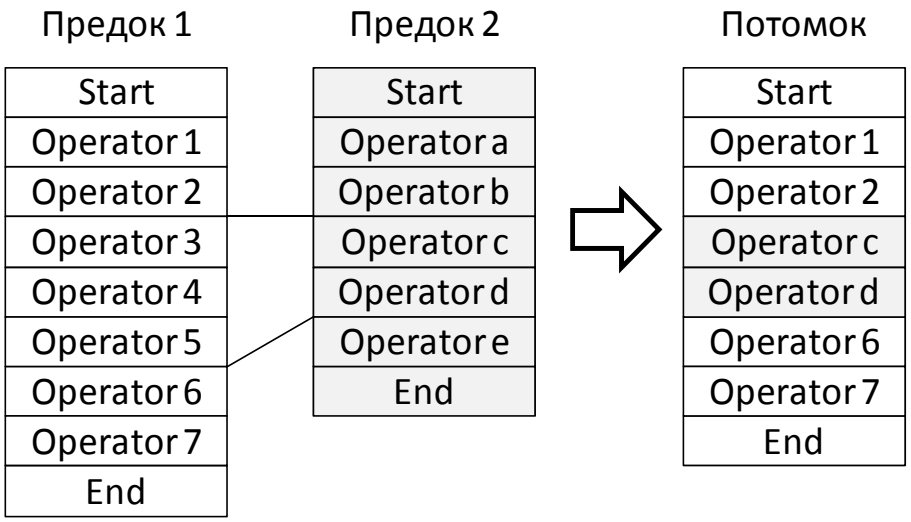


Рис. 2.16 - Кроссовер для линейного представления генома

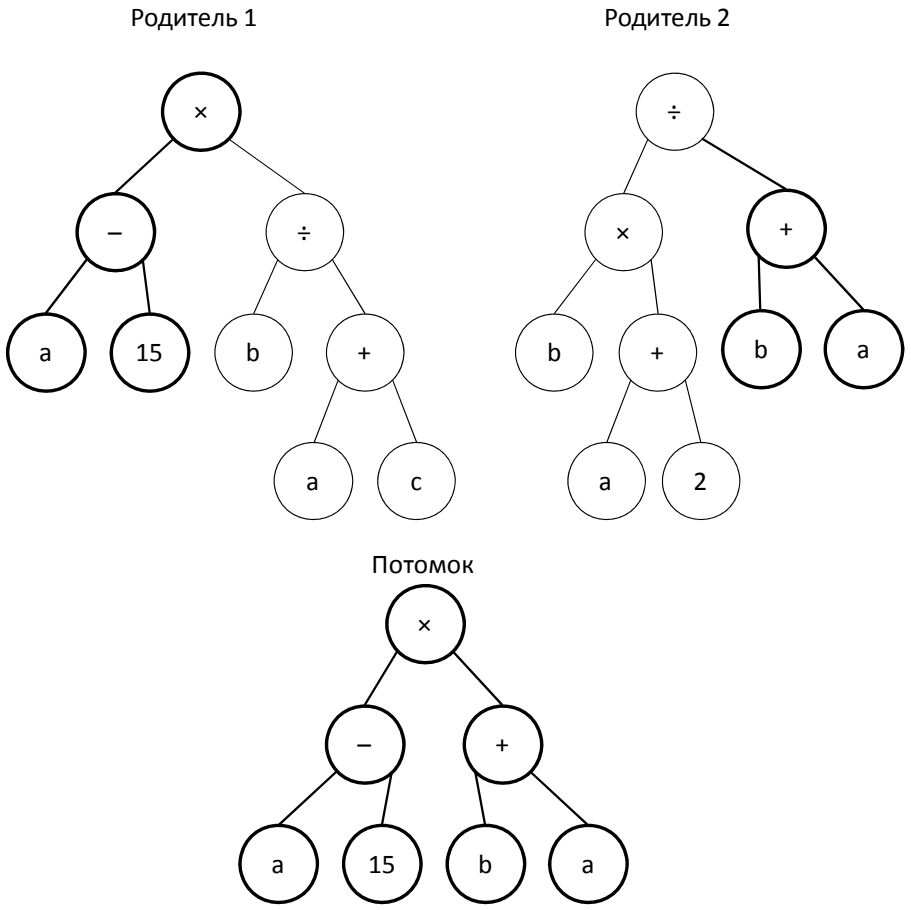


Рис. 2.17 - Кроссовер при древовидном представлении генома

Операция кроссовера чрезвычайно разрушительна для обычных программ и подавляющее количество потомков после такой операции становятся нежизнеспособными. Поэтому, в систему команд часто вводят необычные операции, например, перехода по комплементарным меткам [65], автоопределяемые функции (ADF) [32], и много других усовершенствований.

Сами программы также «борются» с тем, что их разрушает. Например, путем увеличения количества интронов – кусков кода, которые содержат ничего не делающие операции [32]:

- ✓ (NOT (NOT X))
- ✓ (AND … (OR X X))
- ✓ (+ … (- X X))
- ✓ (+ X 0)
- ✓ (* X 1)
- ✓ (* … (DIV X X))
- ✓ (MOVE-LEFT MOVE-RIGHT)
- ✓ (IF (2=1) … X)
- ✓ (A := A)

Такой способ защиты (как и дублирование самых важных участков генома) не является изобретением ГП. Похожие механизмы имеются и в геномах обычных клеток. Рассмотрение всех тонкостей генетического программирования может послужить темой отдельной книги (и не одной). Поэтому интересующимся этими вопросами предложим далее изучать данное направление по специализированной литературе. Хорошим введением является [32]. Рассмотрим относительно простой пример поиска решения для генома переменной длины.

Ставится следующая задача. Необходимо отгадать некоторую фразу. Для того чтобы узнать, отгадали мы фразу или нет, показываем любую строку, а в ответ говорят число (значение *FF*), которая представляет собой разность количества букв, которые стоят на своих местах, с теми, что занимают не свою позицию:

$$f(a,b) = \sum_{i=0}^{\max(length(a),length(b))-1} c(a,b,i)$$

$$c(a,b,i) = \begin{cases} 1, & \text{при } i < length(a) \cap i < length(b) \cap a[i] = b[i] \\ -1, & \text{иначе} \end{cases}. \quad (2.11)$$

Здесь a – строка-образец. Она известна только вычислителю фитнесс - функции. b – строка, которую мы предъявили для оценки. length() – функция, вычисляющая длину переданной строки. Индексация символов начинается с нулевой позиции.

Имея только такие скудные данные и приблизительный размер текста, попробуем отгадать фразу из бессмертного произведения В. Шекспира. А именно, монолог Гамлета: "To be or not to be". Текст монолога взят со страницы http://ru.wikipedia.org/wiki/To_be_or_not, со всеми знаками препинания.

Для решения данной задачи, предназначено приложение SimpleGP из электронного приложения к данному учебному пособию. Так же, как и простой пример реализации ГА, оно является консольным, все данные и настройки внесены в код, поэтому для того, чтобы их поменять, нужно использовать IDE и перекомпилировать программу.

Основные особенности данного примера:

- Метод Main включает в себя генерацию нулевого поколения и основной цикл – генерацию новых поколений. Цикл прерывается тогда, когда находится точная фраза – она хранится в константе SecretPattern класса GPWorld. Это место хранения выбрано только для упрощения примера. На самом деле, исходный текст может храниться где угодно, хоть на другом компьютере. Главное, чтобы была возможность выполнить с ним две операции: сравнение (как условие окончания цикла) и оценку похожести (вычисление fitness-функции).
- Отбор производится турнирным методом из представителей наиболее приспособленной половины популяции. Таким образом, несколько усиливается давление отбора.
- Операция кроссовера похожа на одноточечный кроссовер у обычного ГА, но приспособлена для обработки геномов переменной и неравной длины. Результирующий геном имеет длину, равную первому геному. Если второй геном оказывается короче, то к строке добавляются случайные символы из множества допустимых. Если длиннее – то используется только часть символов в пределах длины первого генома. В данной реализации кроссовера отсутствуют операции, которые приводят к сдвигу фрагментов строк к началу или к концу. Это связано с тем, что оценочная функция оценивает только символы, точно попавшие в свою позицию. Поэтому любые сдвиги приведут практически в каждом случае к тому, что геном будет «испорчен». Подобные операции можно реализовать самостоятельно вместе с модификацией fitness-функции. В этом случае fitness-функция должна добавлять очки геному, если в нем имеются общие с образцом подстроки, пусть даже они стоят не на своих местах.

Приведем фрагменты работы алгоритма (вывод программы сокращен – оставлены только начало и конец фразы):

Поколение 0:

```
Genome=EA-B;-gOcLaRGGhjaQonoAFaHWz
S?QlBBQs;UaoUIpGBUJYLoUeU
        pd--.eEh?g.XqfYWpoPaLm
…
;u,sW.rAFbkYs
  Length=1409
  Fitness=-1350
```

Поколение 500:

```
  Genome=TE be, or notGto b?, tpan .' Ghe eBeJtiln;
WhetRer Vpis irblgrBgT tfekqi dxto iuWdIrw
The SliJ?sLaIn Ao
…
ADd losG theonCme of a-tikn.
  Length=1442
  Fitness=182
```

Поколение 1000:

```
 Genome=To be, or not to be, that is the qBestion;
Whether 'tis nobler in the mind to sufder
The Slings ann Arrmws of outrageous
…
ADd lose the name of action.
 Length=1442
 Fitness=1066
```

Поколение 1426:

```
Genome=To be, or not to be, that is the question;
Whether 'tis nobler in the mind to suffer
The Slings and Arrows of outrageous Fortune
…
And lose the name of action.
 Length=1442
Fitness=1442
```

2.2.5 Интеллект стаи

Большинство алгоритмов, относящихся к мягким вычислениям, эксплуатируют идею того, что множество относительно простых объектов, работающих по вполне понятным правилам, объединяясь, демонстрируют поведение, намного превышающее по интеллектуальности поведение отдельного индивидуума, т.е. наблюдается тенденция к соблюдению принципа *синергии* [44]. Сюда можно отнести и нейронные сети, и эволюционные алгоритмы. Нейронные сети в данном пособии подробно освещаться не будут, поскольку это отдельный курс. К счастью, теория нейронных сетей проработана и описана достаточно хорошо (см., например, [36, 44,56,58,59,60,104,108,116,127,136,143]), поэтому при желании несложно найти именно то учебное пособие, которое позволит вникнуть в данную тематику более глубоко. С эволюционными алгоритмами мы познакомились в предыдущем разделе. Однако, наиболее ярко и явно идея взаимодействия (синергирования) индивидуальных интеллектов, воплощается в таком направлении исследований, как «Интеллект стаи» (swarm intelligence).

Познакомимся с направлением, используя фрагмент фантастического произведения Станислава Лема – «Непобедимый» [33].

Небольшая преамбула. Большой космический корабль землян «Непобедимый» садится на планету, которая кажется безжизненной. Обнаружены только очень простые «растения» и «насекомые» на основе полупроводников и металлов. Но постепенно начинают происходить странные вещи – люди теряют память, погибают, технике кто-то наносит сильнейшие удары. Начинается более подробное изучение «простейших»:

«*"Пленники" занимали во времясовещаниях почетное место в закрытом стеклянном сосуде, стоявшем посреди стола. Их осталось всего*

десятка полтора, остальные были уничтожены в процессе изучения. Все эти создания обладали тройственной симметрией и напоминали формой букву Y, с тремя остроконечными плечами, соединяющимися в центральном утолщении…

Каждый кристаллик соединялся с тремя; кроме того, он мог соединяться концом плеча с центральной частью любого другого, что давало возможность образования многослойных комплексов. Соединение не обязательно требовало соприкосновения, кристалликам достаточно было сблизиться, чтобы возникшее магнитное поле удерживало все образование в равновесии…

Кроме цепи, заведующей такими движениями, каждый черный кристаллик содержал в себе еще одну схему соединений, вернее ее фрагмент, так как она, казалось, составляла часть какой-то большой структуры. Это высшее целое, вероятно возникающее только при объединении огромного количества элементов, и было истинным мотором, приводящим тучу в действие. Здесь, однако, сведения ученых обрывались. Они не ориентировались в возможностях роста этих сверхсистем, и уж совсем темным оставался вопрос об их "интеллекте". Кронотос допускал, что объединяется тем больше элементарных единиц, чем более трудную проблему им нужно решить. Это звучало довольно убедительно, но ни кибернетики, ни специалисты по теории информации не знали ничего соответствующего такой конструкции, то есть "произвольно разрастающемуся мозгу", который свои размеры примеряет к величине намерений.

Часть принесенных Роханом "насекомых" была испорчена. Остальные демонстрировали типовые реакции. Единичный кристаллик мог подпрыгивать, подниматься и висеть в воздухе почти неподвижно, опускаться, приближаться к источнику импульсов либо удаляться от него. При этом он не представлял абсолютно никакой опасности, не выделял, даже при угрозе уничтожения. …Зато объединившись даже в сравнительно небольшую систему, "насекомые" начинали, при воздействии на них магнитным полем, создавать собственное поле, которое уничтожало внешнее, при нагревании стремились избавиться от излишка тепла инфракрасным излучением - а ведь ученые располагали лишь маленькой горсточкой кристалликов».

Чем закончилось противостояние людей и этой интеллектуальной стаи, описывать не будем – если вас заинтересовало само произведение, прочтите его. Оно того стоит. Приведено оно лишь как описание практически идеальной системы «интеллекта стаи».

Почему это направление так привлекает исследователей? Наверное, потому, что в окружающем мире все мы видим примеры таких «интеллектов»:

- Животные (человек как высшее проявление) – уже понятно, что за все очень сложное поведение отвечает система относительно простых (но все еще не до конца изученных) элементов, нейронов.

• Группы ученых чаще всего могут решать более сложные в интеллектуальном отношении задачи, чем одиночки. Хотя, очень часто здесь происходит не параллельное объединение, а последовательное – научные школы, учителя-ученики.

• Производство компьютеров (и любой сложной современной техники) – ни один из ныне живущих людей не знает полностью цикл их производства во всех деталях. Начиная с добычи нефти для производства пластика и металла для производства проводов, заканчивая вопросами дизайна рекламных буклетов для магазинов. Тем не менее, множество людей, объединяясь, успешно производят и продают всю эту технику.

• Военные радиоуправляемые роботы – т.н. «дроны», объединенные в стаи, способные автономно, без участия человека, решать боевые задачи, именно благодаря наличию у них развитого «интеллекта стаи».

• Муравьи и пчелы – рой пчел или муравьиная стая для стороннего наблюдателя производят впечатление чего-то гораздо более интеллектуального, чем отдельные особи. Например, когда наблюдается процесс поиска пищи и оптимизация пути, по которому муравьи приносят найденное в муравейник.

Именно последний пример моделируется в «Муравьиных алгоритмах», на примере которых удобно раскрыть направление Интеллекта Стаи.

Оригинальный алгоритм (Ant Colony Optimization) возник в процессе наблюдения за тем, как живые муравьи вида Argentine Ant обследуют территорию вокруг муравейника, находят пищу и несут её в муравейник, постоянно оптимизируя (сокращая) путь, который проходит каждый из муравьев. Эти исследования проводились в 1989-м году Госсом и в 1990-м Денеборгом. Первая математическая формализация алгоритма предложена в 1992-м году Марко Дориго [144].

Живые муравьи во время поисков пищи ходят вокруг муравейника случайным образом по тропам, которые не являются физическими дорожками, «протоптанными» поколениями насекомых. Основная ориентация у муравьев происходит за счет феромонов, к которым они очень чувствительны и которыми они помечают все вокруг. Более того, каждый муравейник имеет свой индивидуальный запах и муравей того же вида, но из другого муравейника, будет воспринят как враг. Существуют слепые муравьи, которые в пространстве ориентируются только за счет запаха и осязания.

Найдя пищу, муравей-разведчик возвращается домой, используя одну из троп. При этом весь его путь помечается особым феромоном. По возвращении в муравейник, этот муравей «зовет» (назовем это действие таким образом, чтобы не усложнять описание) за собой других, рабочих муравьев, и начинается процесс переноса пищи в муравейник.

Рабочие муравьи идут, ориентируясь на запах, оставленный муравьем-разведчиком. При этом они усиливают запах, оставляя свои следы на дорожке в том случае, если находят пищу. Таким образом, тропинки становятся все более и более заметными. Но они не всегда идут одним и тем

же путем – иногда муравьи сбиваются с пути и тогда случайным образом ищут место, помеченное феромоном. Иногда это приводит к нахождению более короткого пути. Также, в это время происходит процесс испарения феромонов. Если бы его не было, то первоначальный путь всегда имел бы самый сильный запах и процесс поиска более короткого пути не происходил бы.

Представим этот процесс графически.

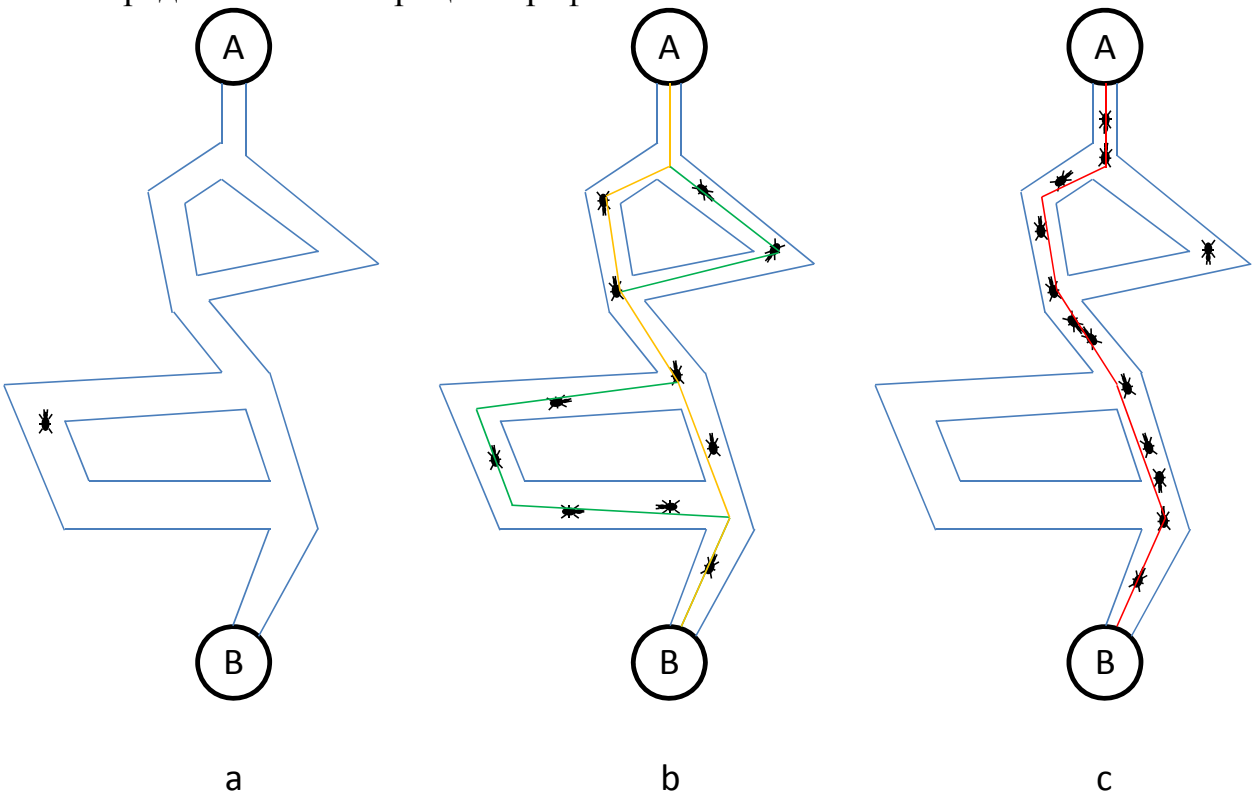


Рис. 2.18. Процесс оптимизации пути при переносе найденной пищи (A) в муравейник (B)

На иллюстрации a), муравей разведчик находит еду, после чего произвольным путем возвращается домой.

b) – рабочие муравьи переносят пищу в муравейник, прокладывая феромонные тропы. По более короткому пути муравьи успевают пройти в большем количестве, поэтому постепенно этот путь становится все более «пахнущим».

c) – большая часть муравьев движется по самому короткому пути и только отдельные особи используют другие пути.

Опишем простейший муравьиный алгоритм более формально. Лабиринт задан графом (вершины и ребра). Считаем, что муравьи ищут оптимальный путь (самый короткий) между двумя вершинами (муравейник и еда).

```
Предварительный этап:
  1. Создаем муравьев.
  2. Муравьи ищут решения.
  3. Происходит обновление уровня феромона.
```

Рассмотрим каждый из шагов цикла более подробно.

1. Начальные точки, куда помещаются муравьи, зависят от ограничений задачи. В простейших случаях, мы можем их всех поместить в одну точку, либо случайно распределить по площади лабиринта. На этом же этапе каждое ребро лабиринта помечается небольшим положительным числом, характеризующим запах феромона. Это нужно для того, чтобы на следующем шаге у нас не было нулевых вероятностей.

2. Определяем вероятность перехода из вершины i в вершину j по следующей формуле:

$$P_{ij}(t) = \frac{\tau_{ij}(t)^{\alpha}\left(\frac{1}{d_{ij}}\right)^{\beta}}{\sum_{j \in connected\ nodes} \tau_{ij}(t)^{\alpha}\left(\frac{1}{d_{ij}}\right)^{\beta}}.$$

Здесь $\tau_{ij}(t)$ – уровень феромона, d_{ij} – эвристическое расстояние, α,β – константы.

Если α=0, то наиболее вероятен выбор ближайшего соседа, и алгоритм становится «жадным».

В случае $\beta=0$, наиболее вероятен выбор только на основе уровня феромона, что приводит к застреванию на уже «протоптанных» путях.

Как правило, используется некоторое компромиссное значение этих величин, подбираемое экспериментально для каждой задачи.

3. Обновление уровня феромона производится следующим образом:

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{k \in usededge(ij)} \frac{Q}{L_k(t)}$$

Здесь ρ – параметр, задающий интенсивность испарения, $L_k(t)$ – цена текущего решения для k-го муравья, а Q характеризует порядок цены оптимального решения. Таким образом, выражение $\frac{Q}{L_k(t)}$ определяет количество феромона, которым муравей k пометил ребро (ij).

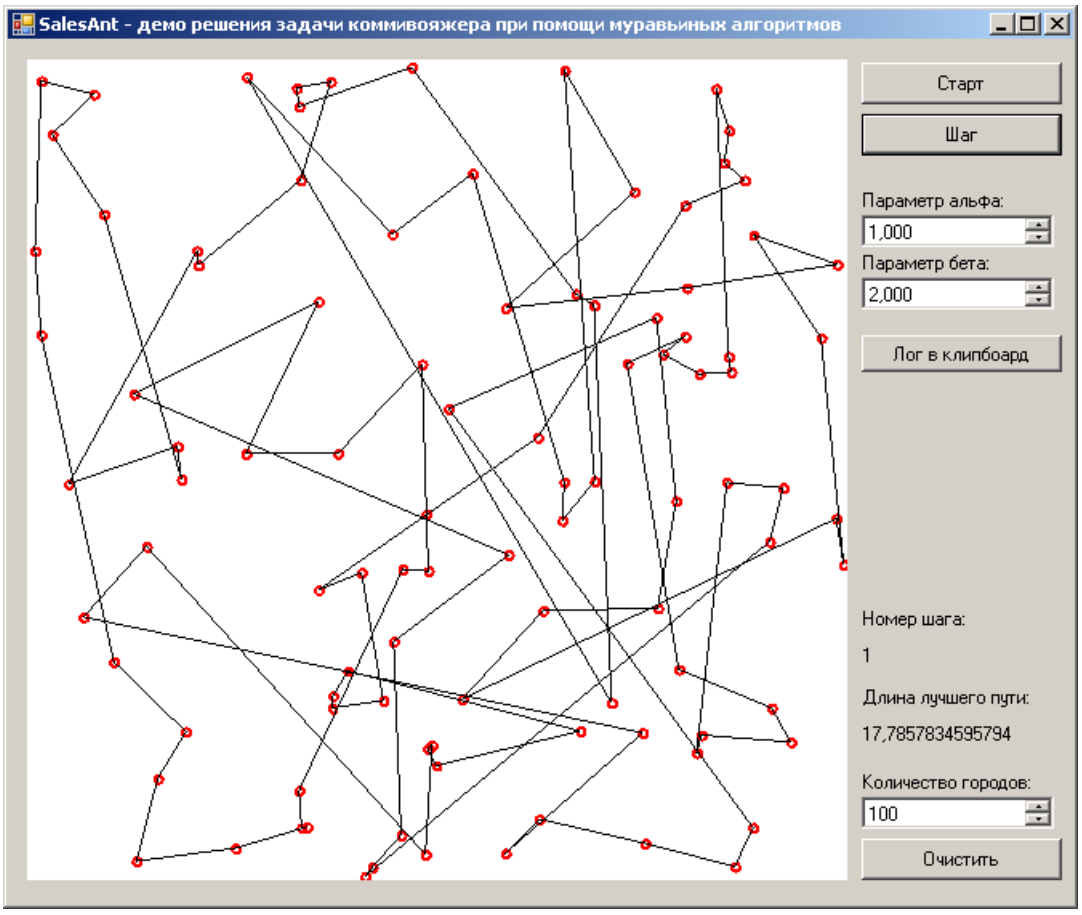
Задача коммивояжера и муравьиный алгоритм

Для иллюстрации работы муравьиного алгоритма решим известную задачу коммивояжера, суть которой в том, что есть набор городов, которые должен посетить коммивояжер, побывав в каждом из них по одному разу и вернувшись в тот город, из которого путешествие было начато. Задачей здесь является оптимизация пути таким образом, чтобы пройденное расстояние было наименьшим. Это одна из самых простых постановок задачи и именно попробуем решить.

Задачи, связанные с графами, неудобно показывать в консольном приложении, поэтому пример SalesAnt (в электронном приложении к учебному пособию: https://cmidpbook.codeplex.com/, загрузка программ архивом - https://cmidpbook.codeplex.com/SourceControl/latest#), написан с использованием WinForms.

Посмотрим, как это приложение выглядит графически. Запустим программу, установим количество городов равным 100, а параметр β равным 2. Далее следуют скриншоты начала работы алгоритма, середина и установившийся процесс. Поскольку при каждом запуске города имеют случайные координаты, то у разных пользователей будут свои картинки, отличные от приведенных.

Приведем скриншоты решения задачи коммивояжера.



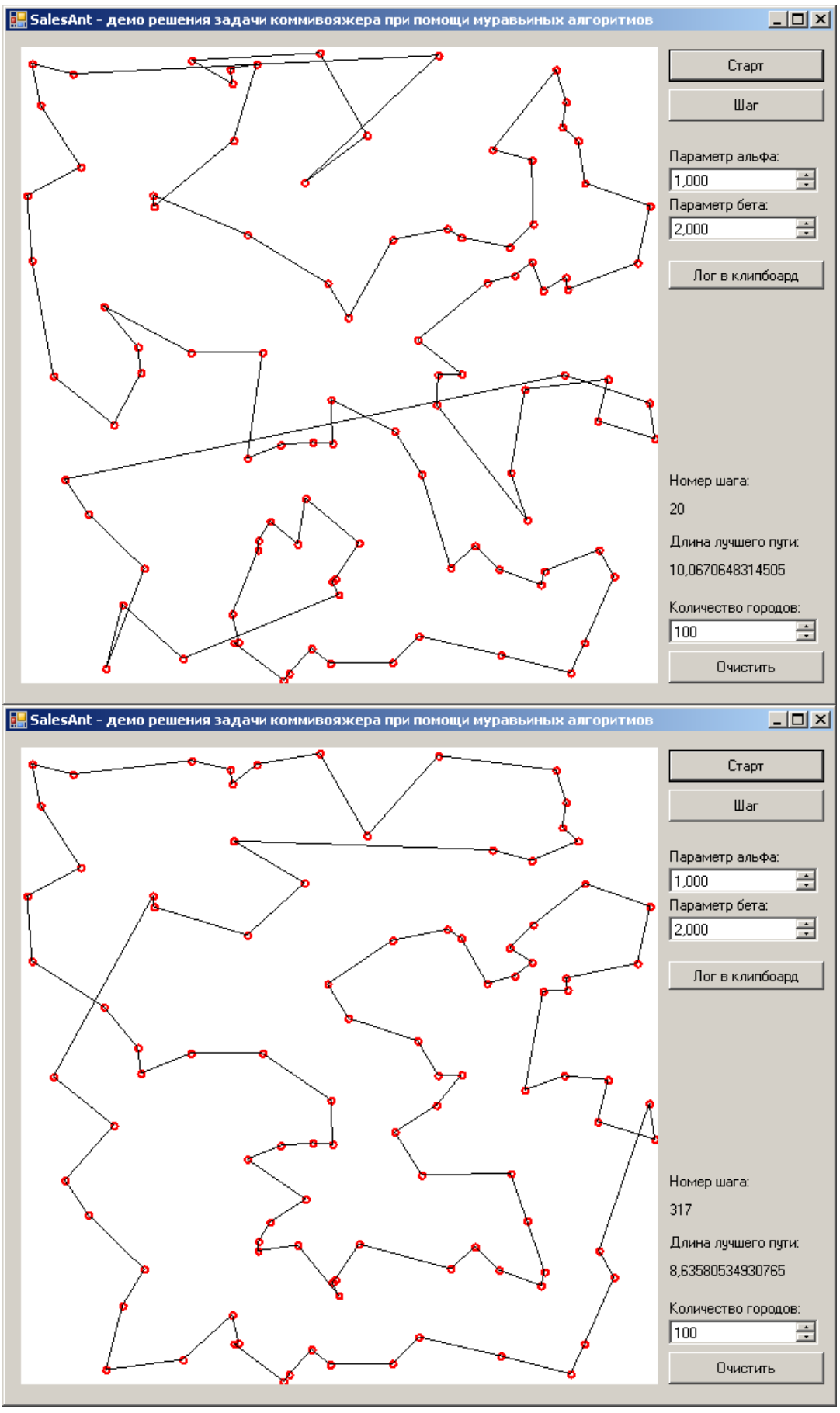


Рис.2.18 - Скриншоты решения задачи коммивояжера

Видно, что в установившемся решении есть еще петли в пути, что говорит о некоторой неоптимальности, но она уже не столь велика, как вначале.

Попробуйте поиграться с параметрами. Так, например, если при данном количестве городов мы выберем параметр β равным 10, то решение найдется быстрее. Но поиск замрет быстрее (ухудшаются поисковые способности алгоритма). Можно выполнить большее количество запусков – мы ведь имеем дело со стохастическим процессом. Можно также придумать свои варианты и оптимизации алгоритма, например, убирающие мелкие петли.

Контрольные вопросы к разделу II:

1. Показать суть процесса выявления ассоциаций.
2. Привести алгоритм МНК.
3. Привести алгоритм метода главных компонент.
4. Формальная постановка задачи поиска ассоциативных правил.
5. Генетический алгоритм и его применение.
6. Определение хромосомы. Привести алгоритмы их формирования.
7. Алгорим построения пространственного кроссовера.
8. Для чего необходим этап мутации в ГА?
9. В чем с точки зрения поиска экстремума целевой функции проявляется социальность колонии муравьев?
10. В чем состоит алгоритм решения задачи коммивояжера?

РАЗДЕЛ III
КЛАСТЕРНЫЙ АНАЛИЗ ДАННЫХ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

3.1. Введение в ИАД кластеризацию

Кластерный анализ занимает одно из центральных мест среди методов интеллектуального анализа данных и представляет собой совокупность подходов, методов и алгоритмов группировки многомерных объектов, основанных на представлении результатов отдельных наблюдений точками подходящего геометрического пространства с последующим выделением групп как «сгустков» - кластеров этих точек.

Как научное направление кластерный анализ (также называемый терминами «распознавание образов без учителя», «численная таксономия», «стратификация», «автоматическая классификация») заявил о себе в середине 60–х годов прошлого века и с тех пор развивается, являясь одной из наиболее мощных ветвей сначала статистической науки, а на сегодняшний день интеллектуального анализа данных, о чем свидетельствует поток публикаций в зарубежных и отечественных журналах [37-45, 50-55]. Причиной этого является то, что моделирование операции группирования является одной из самых важных не только в статистике и методах анализа данных, но и в познании, в принятии решений [66]. При этом, общий вопрос, задаваемый исследователями во многих областях, состоит в том, как организовать наблюдаемые данные в наглядные структуры [50-51].

Автоматическая классификация применяется в целях получения гипотез о логической структуре изучаемой статистической совокупности объектов. Слово «автоматическая» подчеркивает тот факт, что разделение проводится без предварительного обучения с помощью учителя или обучающей выборки, на которой все объекты разнесены по классам. Содержательный смысл деления на классы состоит в выделении качественно различных состояний объектов, характеризуемых своими особенными закономерностями. Само понятие «кластер» (англ. cluster – «скопление», «гроздь») не имеет однозначного определения. В общем, кластер можно охарактеризовать как группу объектов, имеющих общие свойства. Характеристиками кластера можно назвать два признака: 1 – внутренняя однородность, 2 – внешняя изолированность.

В работах [44,45] приведен обзор многих опубликованных исследований, содержащих результаты, полученные методами кластерного анализа. Например, в области медицины кластеризация симптомов заболеваний приводит к широко используемым таксономиям. Известны приложения кластерного анализа в маркетинговых исследованиях, в задачах анализа визуальной информации, геоинформационных системах [57,11,19].

Таким образом, кластерный анализ – это совокупность многомерных статистических процедур, которая позволяет упорядочить объекты по од-

нородным группам на основе схожести признаков для объектов одной группы и отличий между группами.

На рис. 3.1 представлена обобщенная схема методов интеллектуального анализа данных в классе задач кластерного анализа.

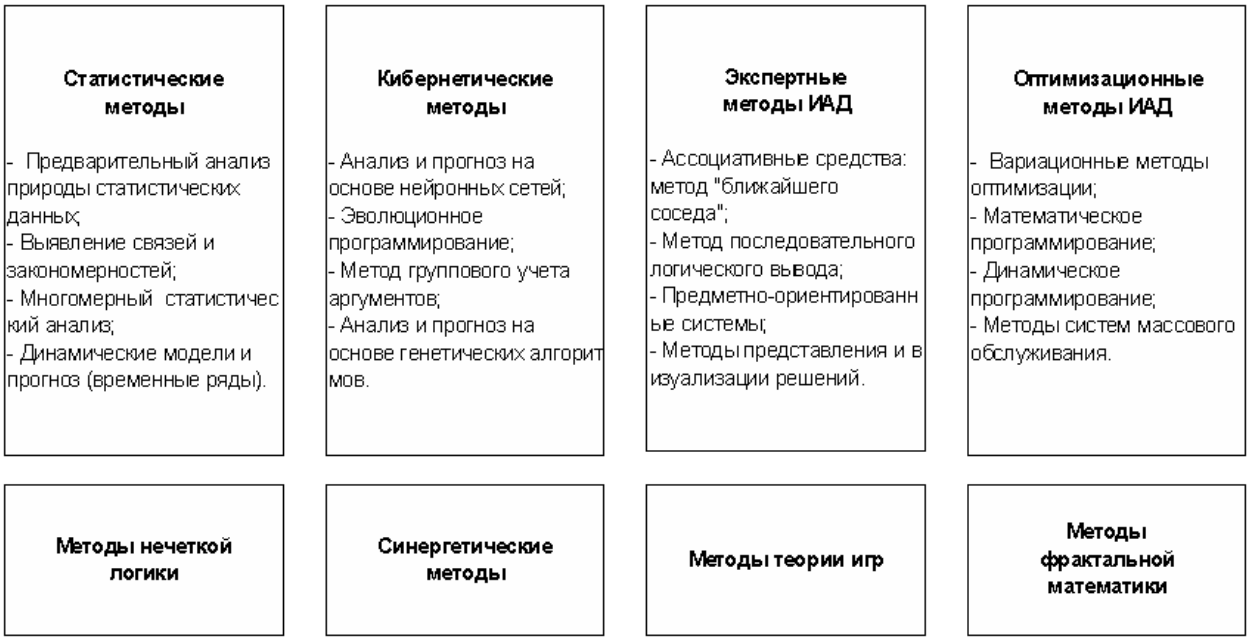


Рис. 3.1. - Методы интеллектуального анализа данных в классе задач кластерного анализа

Кластеризацию проводят для объектов с количественными, качественными или смешанными признаками. Данные обычно представляют собой наблюдения некоторых физических процессов. Каждое наблюдение состоит из n измерений, сгруппированных в n– мерный вектор-столбец $x_k = (x_{1k},...,x_{nk})$, $x_k \in \Re^n$. Множество, состоящее из N наблюдений обозначим $X = \{x_k \mid k = 1, 2,...,N\}$. Таким образом, исходной информацией для кластерного анализа является матрица наблюдений, представленная в виде:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nN} \end{pmatrix}. \qquad (3.1)$$

В терминологии распознавания образов, столбцы матрицы X называются паттернами, образами или объектами, а строки – свойствами или атрибутами, а сама матрица – паттерном или матрицей данных.

3.2. Исследование методов ИАД кластеризации

Задача кластеризации состоит в разбиении объектов на несколько подмножеств (кластеров), в которых объекты более схожи между собой, чем с объектами из других кластеров. В метрических пространствах «схожесть» обычно определяют через расстояние. Расстояние может рассчитываться как между исходными объектами (строчками матрицы), так и от этих объектов к прототипу кластеров. Обычно координаты прототипов заранее неизвестны – они находятся одновременно с разбиением данных на кластеры. Прототипы могут быть векторами, соразмерными с объектами данных, однако могут представлять собой геометрические объекты более высоких порядков, линейные и нелинейные подпространства или функции [64,106,153, 155]

Проведем анализ метрик сходства и различия объектов, используемых в задачах кластерного анализа. Пусть дано множество X. Метрикой на множестве X называется функция d(x, y), определенная на произведении X × X и удовлетворяющая следующим аксиомам:

1. $d(x, y) \geq 0$, для всех $x, y \in X$;
2. $d(x, y)=0$ влечет $x = y$;
3. $d(x, y)= d(y, x)$;
4. $d(x, z) \leq d(x, y)+ d(y, z)$ для всех $x, y, z \in X$ (неравенство треугольника).

Метрическим пространством называется пара (X, d) [47,48].

Меры сходства для кластерного анализа могут быть следующих видов: мера сходства типа расстояния (функции расстояния), называемая также мерой различия; мера сходства типа корреляции, называемая связью, – является мерой, определяющей похожесть объектов.

Наиболее общей мерой является метрика Минковского:

$$d_{ij} = \sqrt[r]{\sum_{k=1}^{n} \left| x_{ij} - x_{jk} \right|^r}. \qquad (3.2)$$

Если в метрике Минковского положить r =2, получим стандартное Евклидово расстояние:

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ij} - x_{jk})^2}. \qquad (3.3)$$

При r =1 метрика Минковского дает Манхэттенское расстояние

$$d_{ij} = \sum_{k=1}^{n} | x_{ij} - x_{jk} |. \qquad (3.4)$$

При $r \to \infty$ метрика Минковского дает метрику доминирования:

$$d_{ij} = \max |x_{ij} - x_{jk}|, \; k =1, 2, ..., n, \qquad (3.5)$$

которая совпадает с супремум-нормой (∞–нормой):

$$d_{ij} = \sup \{|x_{ij} - x_{jk}|\} \; ,k =1, 2, ..., n. \qquad (3.6)$$

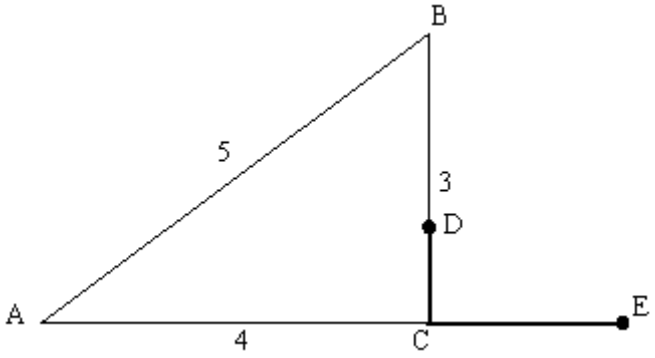


Рис. 3.2 - Сравнение метрик

На рис. 3.2 изображен прямоугольный треугольник ABC с катетами BC длиной 3 единицы и AC – 4 единицы (так называемый египетский треугольник). Пусть точка A – начало декартовой системы координат, то есть ее координаты (0; 0), тогда точка B имеет координаты (4; 3), точка C (4; 0). Еще две точки: D (4; 1), E (6; 0). Сравним расстояния AB, AE, AD, вычисленные с помощью метрик:

Таблица 3.1.

Сравнение метрик

Метрика	Расстояние между точками, ед.		
	AB	AE	AD
Евклидово расстояние	5	6	$\sqrt{17} \approx 4.12$
Манхэттенское расстояние	7	6	5
Метрика доминирования	4	6	4

Анализируя результаты, приведенные в табл. 3.1, можно отметить, что по Манхэттенскому расстоянию, в отличие от Евклидова расстояния, точка B более удалена от точки A, чем точка E. По метрике доминирования точки B и D равноудалены от точки A.

Мера Махаланобиса (расстояние Махаланобиса, обобщенное Евклидово расстояние) вычисляется как:

$$d_{ij} = (X_i - X_j)`\Sigma^{-1}(X_i - X_j), \qquad (3.7)$$

где Σ – общая внутригрупповая дисперсионно-ковариационная матрица.

Метрика (3.3) используется традиционно, метрика (3.4) является наиболее известным представителем класса метрик Минковского. Расстояние Махаланобиса, по определению метрикой не являющееся, связано с помощью дисперсионно-ковариационной матрицы с корреляциями переменных, и широко применяется как в кластерном, так и в других методах анализа данных [46].

В тех случаях, когда распознаваемые объекты представлены бинарными кодовыми последовательностями или строками символов, рекомендуется использование расстояния Левенштейна. Расстояние Левенштейна между строками X и Y определяется как наименьшее число преобразований, требуемых для получения строки Y из строки X:

$d_L(X, Y) = \min (a(i)+ b(i)+ c(i))$, (3.8)

где $a(i)$ – количество замен символов при выполнении i–го варианта преобразований символов, $b(i)$ – количество вставок символов, $c(i)$ – количество удалений.

При наличии качественных признаков используются меры различия типа расстояния Хэмминга [46]. Так, на n–мерном булевом кубе E_n, расстояние Хэмминга определяется как

$$d_{H_m}(X,Y) = \sum_{i=1}^{n} x_i \oplus y_i \,, \tag{3.9}$$

что можно проинтерпретировать как количество отличающихся битов в двух бинарных векторах.

Метрика Хаусдорфа [47, 48] позволяет вычислить расстояние между множествами. Пусть в некотором пространстве определено расстояние между точками $d(x, y)$. Расстояние $d(x, Y)$ от точки x до множества Y определяется как нижняя грань расстояний $d(x, y)$ для $y \in Y$. Расстояние от множества X до множества Y определяется как верхняя грань расстояний $d(x,Y)$ для всех $x \in X$. Расстояние Хаусдорфа между двумя множествами X и $Y H(X, Y)$ определяется как

$$d_H(X,Y) = \max\{\max_{y\in Y} \min_{x\in X} d(x,y), \max_{x\in X} \min_{y\in Y} d(x,y)\}. \tag{3.10}$$

В работе [49] рассмотрен вопрос построения локальной «контекстнозависимой» метрики, позволяющей учитывать контекст взаимоотношений объекта с окружающими его объектами.

Наряду с вышеперечисленными широко известными метриками, существуют и другие метрики, такие как метрика Брея–Кертиса, Канберровская метрика, меры близости Журавлева, Воронина, Миркина и многие другие [44, 48-53].

В задачах кластерного анализа выбор метрики имеет большое значение. Например, в том случае, если все независимые переменные имеют одну и ту же размерность, Евклидова метрика имеет естественный смысл, понятна и адекватна. Но если независимые переменные измерены в разных шкалах (например, одна из независимых переменных – вес пациента, а вторая – рост, возникает проблема сравнения разницы по одной оси в 1 кг с разницей в 1 см по другой оси), то есть, в этом случае пространство независимых переменных – это аффинное пространство, а не метрическое. Один из возможных способов преодоления этой трудности – нормирование всех независимых переменных на некоторое естественное значение этой переменной или характерный масштаб [54].

Если естественные характерные значения переменных неизвестны, каждую независимую переменную можно разделить на величину ее дисперсии. При этом дисперсии всех независимых переменных становятся равными единице, и это дает основания полагать, что их изменения на одну и ту же величину сопоставимы между собой.

Кроме того, для изучения полученного разбиения объектов на однородные группы традиционно применяют следующие математические характеристики кластеров.

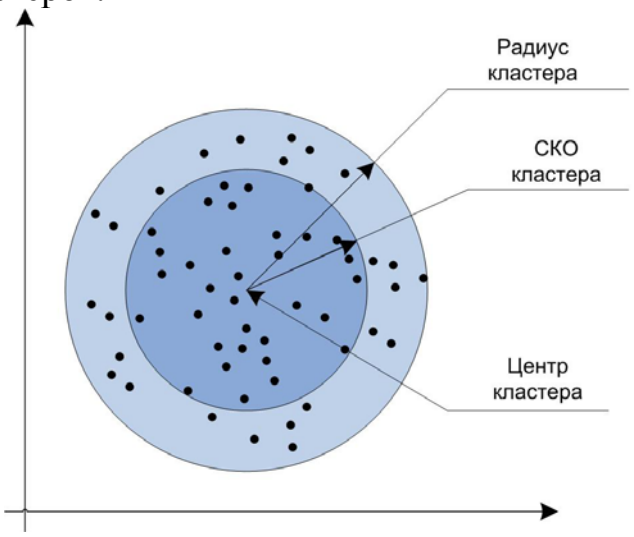


Рис. 3.3 - Математические характеристики кластера

Пусть исследуемая совокупность объектов X = ($x_1,x_2,...,x_n$) разделена на классы $A^1 = (A^1,...,A^c)$, l =1,...,*c*, n^{A^l} – количество объектов, принадлежащих A^1 классу. Определим следующие характеристики полученного разбиения.

Центр кластера – это среднее геометрическое место точек в пространстве переменных:

$$\bar{V}_{A^l} = \frac{\sum_{j=1}^{n^{A^l}} x_j}{n^{A^l}}. \qquad (3.11)$$

Дисперсия кластера – это мера рассеяния точек в пространстве относительно центра кластера:

$$D_{A^l} = \frac{\sum_{j=1}^{n^{A^l}} (x_j - V_{A^l})^2}{n^{A^l} - 1}. \qquad (3.12)$$

Среднеквадратичное отклонение (СКО) объектов относительно центра кластера:

$$S_{A^l} = \sqrt{\frac{\sum_{j=1}^{n^{A^l}} (x_j - V_{A^l})^2}{n^{A^l} - 1}}. \qquad (3.13)$$

Радиус кластера – максимальное расстояние точек от центра кластера:

$$R_{A^l} = \max \sqrt{\sum_{j=1}^{n} (x_j - \bar{V}_{A^l})^2}\ . \qquad (3.14)$$

В зависимости от используемой парадигмы, выделяют следующие направления кластерного анализа [53, 40, 54]: иерархические методы направлены на выявление структуры исходного множества; оптимизационные методы сводятся к поиску оптимального разбиения, доставляющего экстремум выбранному функционалу; эвристические методы ставят задачу формального определения понятия «кластер» и построения соответствующей заданному определению кластер-процедуры; аппроксимационные процедуры ставят задачу нахождения аппроксимирующего отношения, соответствующего представлению о наилучшей классификации исходной совокупности.

Группа методов иерархического направления включает в себя агломеративные и дивинзимные процедуры, характеризующиеся последовательным объединением либо разделением элементов исходной совокупности. Их результаты обычно оформляются в виде так называемой дендрограммы (рис. 3.4), где по горизонтали показаны номера объектов, а по вертикали значения межклассовых расстояний, при которых произошло объединение двух данных кластеров.

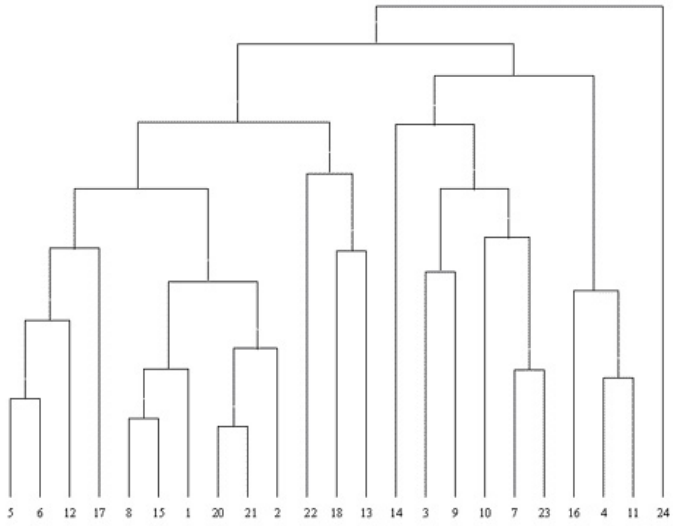


Рис. 3.4 - Древовидная диаграмма

В агломеративных иерархических методах первоначально все объекты рассматриваются как отдельные кластеры, состоящие из одного элемента. После вычисления матрицы расстояний начинается процесс агломерации (от лат. agglomero – присоединяю), который характеризуется последовательным объединением исходных элементов и соответствующим уменьшением числа кластеров. В противоположность агломеративным, в дивизимных методах на начальном этапе вся выборка рассматривается как единый кластер, на последующих шагах происходит процесс его деления на составляющие части. Важным вопросом для иерархических методов является выбор метрики близости между объектами, а также способ измерения расстояния между кластерами. Одиночная связь, используемая в методе ближайших соседей, под расстоянием между кластерами понимает расстояние между их ближайшими объектами. Противоположная ей «полная»

связь используется в методе наиболее удаленных соседей и определяет расстояние между наиболее удаленными объектами кластеров. Центроидный и медианный методы рассматривают расстояние между кластерами как расстояние между их центрами тяжести, причем в методе медиан размеры кластеров используются как веса. Расстояние между кластерами может быть измерено как среднее расстояние между всеми парами их объектов, причем в методах, использующих взвешенное попарное среднее, размер соответствующих кластеров используется в качестве весового коэффициента при вычислениях расстояний. Иерархические методы очень наглядны, однако неэффективны для выборок больших объемов.

Наиболее распространенным среди оптимизационных методов является алгоритм *k*–средних, рассмотренный в [45]. В отличие от иерархических методов, которые не требуют предварительных предположений относительно числа кластеров, для возможности использования этого метода необходимо иметь гипотезу о наиболее вероятном количестве кластеров. Метод основан на итерационном процессе вычисления центров и перераспределения объектов (рис. 3.5) [55], пока не произойдет стабилизация центров кластеров, либо число итераций не достигнет максимума. Недостатком данного метода является чувствительность к выбросам, а также ограничения, проистекающие из предположений о количестве кластеров.

Эвристические алгоритмы в основном базируются на процедурах разрезания графа, обладают возможностью визуализации и выделения кластеров сложной, в том числе невыпуклой формы. В качестве недостатков, можно отметить трудоемкость, и, как следствие, малую пригодность для обработки больших массивов.

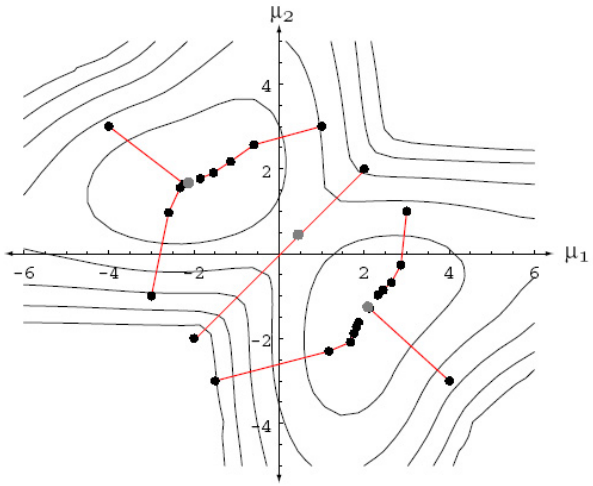


Рис. 3.5 - Процесс кластеризации методом k–средних

В современной литературе встречаются многочисленные примеры решения задач кластерного анализа с помощью искусственных нейронных сетей (ИНС) – как обучаемых, так и ИНС без учителя [56-60], которые относятся к аппроксимационному направлению кластерного анализа. Серьезный недостаток нейронно–сетевого подхода состоит в том, что обученная нейронная сеть представляет собой «черный ящик». Выявленные зако-

номерности (знания), зафиксированные как веса многих сотен межнейронных связей, практически не поддаются анализу и интерпретации человеком. Для кластерного анализа объектов, обладающих качественными признаками, могут быть использованы алгоритмы семейства ART [61], предложенные Гроссбергом и Карпентером, основанные на теории адаптивного резонанса и биологической мотивации.

3.3. Проблема неопределенности в кластерном анализе

Зачастую в ходе решения прикладных практических задач, оказывается, что задаче свойственна нечеткость, значительно затрудняющая или делающая невозможным получение решения, так что на первый план выходит проблема устранения нечеткости, присущей задаче классификации. Понятие нечеткости является общенаучным [62] и может быть определено как внешнее выражение качества внутренней основы явлений, специфика которого заключается в непрерывности перехода от отсутствия проявления к полному выявлению качества предметов, свойств и отношений реального мира [63], что находит свое отражение в познавательной и мыслительной деятельности человека.

Рассмотрим возможные виды и формы проявления нечеткости в задачах кластерного анализа.

Данные могут образовывать кластеры различных геометрических форм, размеров и плотности. На рис. 3.6 показаны различные варианты форм и пространственного взаимного расположения кластеров: а – иллюстрирует случай, когда кластеры соединены цепочкой из внутренне связных объектов выборки; б – различный объем и плотность кластеров; в – кластеры являются невыпуклыми множествами; г – имеет место пересечение кластеров.

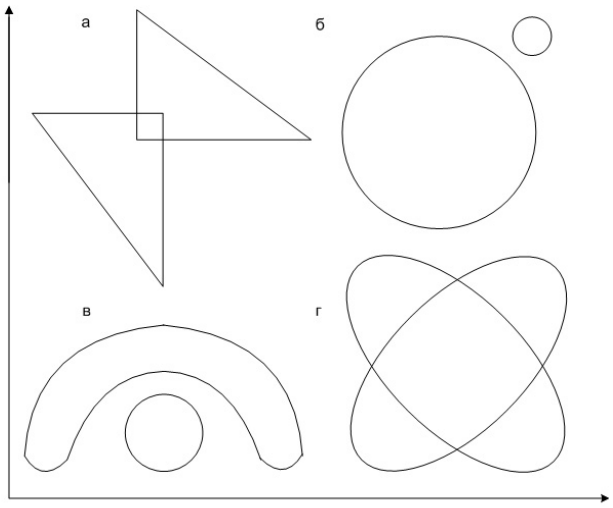


Рис. 3.6 - Вариации форм и взаимного пространственного расположения кластеров

Практические исследования показывают, что традиционные статистические методы кластерного анализа в таких случаях зачастую не дают устойчивых результатов.

Проведем обобщение проблем неопределенности, возникающих в задачах кластерного анализа.

В работе [64] было дано следующее обобщающее определение, базирующееся на известной классификации типов проблем управления и принятия решений Г. Саймона, основывающейся на способах описания характеристик исследуемого процесса: 1 – хорошо структурированные (количественно сформулированные); 2 – неструктурированные (качественно выраженные); 3 – слабо структурированные (смешанные).

Определение 3.1. *Слабоформализованным* процессом называется динамический процесс, относящийся к классу неструктурированных и слабо структурированных проблем принятия решений, обладающий следующими характеристиками:

- уникальность процесса;
- неоднородность (разнотипность) шкал измерений параметров;
- нелинейный (импликативный) характер взаимосвязи характеристик;
- многоуровневая иерархическая организация взаимосвязи подпроцессов;
- многообразие возможных форм взаимодействия подпроцессов между собой, порождающее неоднородность информации, циркулирующей в системе.

Исходя из цепочки (рис. 3.7), введем следующее определение.

Определение 3.2. *Слабоструктурированные* данные представляют собой буквенно-цифровые значения, характеризующие *слабо формализованные* процессы и обладающие следующими характеристиками:

- уникальность;
- гетерогенность (отсутствие части данных, разнотипность шкал измерений различных атрибутов, дублирование элементов данных, структура части информации может зависеть от точки зрения пользователя, часть данных могут иметь плавающую структуру либо не иметь структуры;
- многоуровневая иерархическая организация взаимосвязей атрибутов;
- отсутствие возможности однозначной классификации по определенным атрибутам;
- отсутствие априорной информации (вид и параметры распределения вероятности по кластерам, центры плотности, количество кластеров).

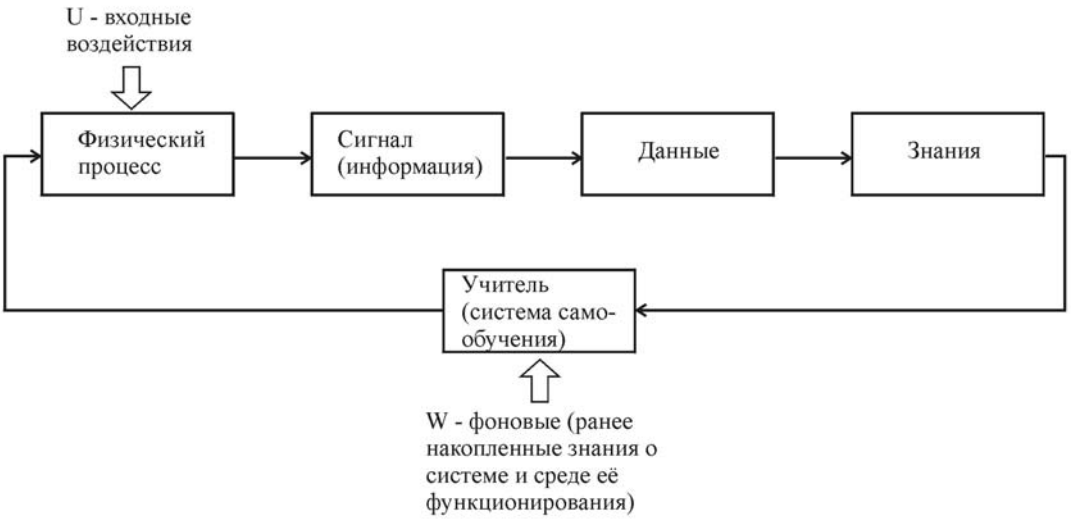


Рис. 3.7 - Схема процесса приобретения знаний

К слабоструктурированным можно отнести данные, полученные при анализе различных социальных систем [57, 65, 66, 67], систем технической диагностики машин и механизмов, систем идентификации и проектирования сложных объектов [68, 69, 70, 71, 72, 73]. К данному классу также можно отнести графическую информацию – фото-и видеоизображения, которые являются одним из наиболее сложных типов данных с точки зрения интеллектуального анализа, но в то же время очень важным практически и сосредотачивающим усилия большого числа исследователей [41, 60, 74]. Примеры слабо структурированных данных содержит репозиторий, сформированный учеными университета г. Ирвин (Калифорния, США) [75]. Репозиторий UCI (UCI Machine Learning Repository) – крупнейший репозиторий реальных и модельных задач машинного обучения, содержащий реальные данные по прикладным задачам в области биологии, медицины, физики, техники, социологии, и других сфер науки и жизни и используемый научным сообществом для эмпирического анализа алгоритмов машинного обучения.

Рассмотрим понятие слабой структурированности данных как один из аспектов неопределенности, присущей задаче кластерного анализа, а также систематизируем типы неопределенности и этапы кластерного анализа, на которых данный тип неопределенности может возникнуть (рис. 3.8).

Конечной целью кластерного анализа является либо выделение четко выраженных классов в анализируемом многомерном пространстве, либо получение наглядного представления о структуре исследуемой совокупности объектов, либо оценка параметров искомой классификации, минимально отличающейся от структуры исходных данных [76]. Таким образом, *неопределенность установки целей* исследования может повлечь за собой некорректность математической постановки задачи. Касательно *характера результатов исследования*, неопределенность может проявляться как качественная характеристика, отвечающая на вопрос: "А должно ли полученное разбиение быть размытым, имеется ли перекрытие или нало-

жение классов?" Если же говорить о количестве, форме и взаимном расположении кластеров, то неопределенность выступает как количественная характеристика. Неопределенность *исходных данных* может проявляться как в недостаточности данных (отсутствии части значений в матрице исходных данных), так и в неточности (в случае, если исходные данные были определены путем экспертной оценки, либо измерены с некоторой погрешностью).

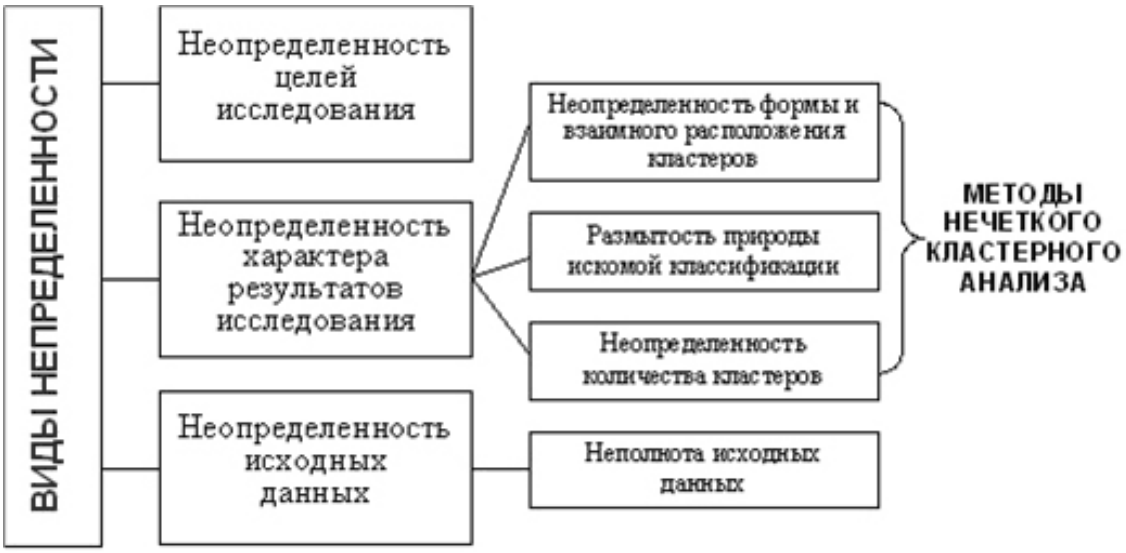


Рис. 3.8 - Основные концепции неопределенности в задачах автоматической классификации

Из изложенного выше возникает вопрос о том, какие методологии наиболее пригодны для обработки неопределенности в задачах автоматической классификации.

Анализ современных публикаций по методам интеллектуальной обработки данных [37, 49, 65, 77, 78] показывает, что по сравнению с традиционными жесткими вычислениями, мягкие вычисления более приспособлены для работы с неточными, неопределенными или частично истинными данными и знаниями. По словам Л. Заде [79], **руководящим принципом мягких вычислений является:** *«терпимость к неточности, неопределенности и частичной истинности для достижения удобства манипулирования, робастности, низкой стоимости решения и лучшего согласия с реальностью»*.

Нечеткая логика лежит в основе методов работы с неточностью, информацией с зернистой структурой (гранулированной), приближенных рассуждений и, что наиболее важно, вычислений со словами (вербальной информацией).

Важной характеристикой нечеткой логики является то, что любая теория Т может быть фаззифицирована и, следовательно, обобщена путем замены понятия четкого множества в Т понятием нечеткого множества. Таким способом можно прийти к нечеткой арифметике, нечеткой топологии, нечеткой теории вероятностей, нечеткому управлению, нечеткому анализу

решений и т.д. Выигрышем от фаззификации является большая общность и лучшее соответствие модели действительности. Однако с нечеткими числами труднее оперировать, чем с четкими. Кроме того, значения большинства нечетких понятий зависят от контекста решаемой задачи. «Это та цена, которую необходимо заплатить за лучшее согласие с реальностью» [79].

Одним из важнейших особенностей нечетких систем, по мнению Заде, является их способность к гранулированию информации.

Грануляция рассматривается как одна из базисных концепций когнитивной обработки информации. Предполагается, что любой составной информационный объект (переменная, отображение, образ) может быть декомпозирован на гранулы. Каждая гранула является набором элементарных объектов, которые связаны вместе неопределенностью, близостью, подобностью и функциональностью. Формально объект O_c может быть представлен гранулировано, то есть:

$$\begin{aligned} O_c &= ins_g\left(G_1,...,G_i,...,G_N\right), \\ G_i &= has_g\left(A_1,...,A_j,...,A_M\right), \\ A_j &= has_v\left(V_1,...,V_q,...,V_Q\right), \end{aligned} \tag{3.15}$$

где A_j – j–й атрибут гранулы G_i; V_q – q–е значение атрибута A_j; ins_g – отношение включения для гранул; has_a и has_v отношение «имеет» для атрибута и значения соответственно.

Гранулы могут быть точными (интервалы переменных, выделенные области определения функций и отношений, сегменты образов) и неточными (терм-множества переменных, элементы нечетких или вероятностных графов, нечеткие или вероятностные правила и др.)

В нечеткой логике гранулирование информации лежит в основе понятий лингвистической переменной и нечетких продукционных правил [80, 81] и формально было введено в [68].

Таким образом, на основе вышеизложенного, можно сделать выводы о том, что нечеткий подход к решению задач кластерного анализа открывает новые возможности интерпретации результатов классификации.

В самом деле, гибридизация методов интеллектуальной обработки информации на основе «мягких вычислений» (soft computing, раздел II) – современное перспективное направление исследований в области искусственного интеллекта [28,82].

В результате расширения существующих методов средствами нечеткой логики образовались такие новые направления как: нечеткие нейронные сети, позволяющие осуществлять выводы на основе аппарата нечеткой логики и являющиеся универсальными аппроксиматорами [83, 84]; адаптивные нечеткие системы, позволяющие осуществлять подбор параметров нечеткой системы в процессе обучения на экспериментальных данных [85]; нечеткие запросы к базам данных, разработанные Д. Дюбуа и

Г. Праде [86]; нечеткие ассоциативные правила, позволяющие осуществлять извлечение из баз данных закономерностей в виде лингвистических высказываний [87, 88]; нечеткие когнитивные карты, предложенные Б. Коско [84], использующиеся для моделирования причинно–следственных взаимосвязей, выявленных между концептами некоторой области; нечеткие деревья решений; нечеткие и нейро-начеткие сети Петри, другие гибридные методы [146].

Контрольные вопросы к разделу III:

1. Привести математические характеристики кластера.
2. Назвать направления развития кластерного анализа.
3. В чём состоит эвристический метод выбора направления.
4. Оптимизационный метод выбора направления.
5. В чём суть агломеративніх и дивизионных процедур в КА?
6. Дать определение слабоструктурированных данных.
7. К чему может привести неопределённые установки целей исследований?
8. В чём суть грануляции данных? Какие гранулы в нечёткой логике Вы знаете?
9. Привести руководящий принцип «мягких вычислений».

РАЗДЕЛ IV

ОСНОВЫ ПРОГНОЗИРОВАНИЯ ДАННЫХ

Раздел посвящен изложению основ прогнозирования данных на примере решения задачи краткосрочного прогнозирования значений временного ряда методом экспоненциального сглаживания (the Method of Moving Average по Р. Брауну, [92,93,98]).

В заключение раздела рассмотрены вопросы структурного моделирования систем прогнозирования на основе алгоритмов агрегатирования моделей (создание ансамблей моделей) такие как: беггинг (bagging), бустинг (boosting) и стэкинг (stacking).

4.1 Временные ряды и стохастические процессы

Рассматрим только дискретные временные ряды, в которых наблюдения делаются через фиксированный интервал времени, принимаемый за единицу счета. Переход от момента одного наблюдения к моменту следующего наблюдения будем называть шагом.

Известно [99], что если значения членов временного ряда точно определены какой-либо математической функцией, то временной ряд называется детерминированным. Если эти значения могут быть описаны только с помощью распределения вероятностей, временной ряд называется случайным.

Явление, развивающееся во времени случайным образом, можно рассматривать и называть стохастическим процессом, в случае, если механизм случайности можно описать в понятиях теории вероятности. В противном случае случайный процесс необходимо рассматривать как хаотический.

В данном разделе рассмариваются только стохастические процессы. При этом анализируемый отрезок временного ряда рассматривается как одна из частных реализаций (выборка) изучаемого стохастического процесса со скрытым вероятностным механизмом.

В свою очередь, среди стохастических процессов выделяют класс процессов, называемых стационарными. Для их математической формализации обозначим член временного ряда, наблюденный в момент t через x_t.

Стохастический процесс называется стационарным, если его свойства не изменяются во времени. В частности, он имеет постоянное математическое ожидание $\bar{x} = M(x_t)$ (т. е. среднее значение, относительно которого он варьирует), постоянную дисперсию $D(x) = M[(x_t - \bar{x})^2] = \sigma_x^2$, оп-

ределяющую размах его колебаний относительно среднего значения, а также постоянную автоковариацию и коэффициенты автокорреляции*.

Ковариация между значениями x_t и x_{t+k}, отделенными интервалом в k единиц времени, называется автоковариацией с лагом (задержкой) k и определяется как

$$R_{xx}(k) = \mathrm{cov}(x_t, x_{t+k}) = M[(x_t - \bar{x})(x_{t+k} - \bar{x})].$$

Для стационарных процессов автоковариация зависит только от лага k и $R_{xx}(0) = \sigma_x^2$. Автокорреляция с лагом k является лишь нормированной автоковариацией и равна:

$$p_k = \frac{M[(x_t - \bar{x})(x_{t+k} - \bar{x})]}{\sqrt{M[(x_t - \bar{x})^2]M[(x_{t+k} - \bar{x})^2]}} = \frac{M[(x_t - \bar{x})(x_{t+k} - \bar{x})]}{\sigma_x^2},$$

так как для стационарного процесса $\sigma_x^2 = const$. Таким образом, k-й коэффициент автокорреляции $p_k = \dfrac{R_{xx}(k)}{R_{xx}(0)}$.

Коэффициент автокорреляции обладает тем свойством, что $-1 \le p_k \le 1$.

Представим, что временной ряд x_t, генерируемый некоторой системой, можно представить в виде двух компонент:

$$x_t = \xi_t + \varepsilon_t,$$

где величина ε_t является случайным неавтокоррелированным процессом с нулевым математическим ожиданием и конечной (не обязательно постоянной) дисперсией, а величина ξ_t может быть либо детерминированной функцией, либо случайным процессом, либо какой-нибудь их комбинацией. Величины ε_t и ξ_t различаются характером воздействия на значения последующих членов ряда. Переменная ε_t влияет только на значение синхронного ей члена ряда, в то время как величина ξ_t в известной степени определяет значение нескольких или всех последующих членов ряда. Через величину ξ_t осуществляется взаимодействие членов ряда; таким образом, в ней содержится информация о системе, необходимая для получения прогнозов. Назовем величину ξ_t уровнем ряда в момент *t,* а закон эволюции уровня во времени - *трендом.* Таким образом, тренд может быть выражен как детерминированной, так и случайной функциями, либо их комбинацией. Стохастические тренды имеют, например, ряды со случайным уровнем или случайным скачкообразным характером изменения (случайные процессы волновой структуры).

* Такие процессы называют стационарными процессами второго порядка, но, так как другие классы стационарных процессов рассматриваться в этом разделе не будут, мы будем называть их просто стационарными.

Приведем пример детерминированного тренда 2-го порядка:

$$\xi_t = a_1 + a_2 t + a_3 t^2,$$

где a_1, a_2, a_3 — постоянные коэффициенты; t – время.

Пример случайного тренда:

$$\xi_t = \xi_{t-1} + u_t = \xi_0 + \sum_{i=1}^{t} u_i$$

где ξ_0 – начальное значение; u_t – случайная переменная.

Пример тренда смешанного типа:

$$\xi_t = a_1 + a_2 t + u_t + q u_{t-1} + b \sin \omega t,$$

где a_1, a_2, q, b, ω- постоянные коэффициенты; u_t — случайная переменная.

Известно множество определений уровня и тренда ряда (см. [91, с. 16]). Существующие понятия тренда противоречивы и имеют условный характер. Каждое из этих определений скорее указывает на частный способ оценки тренда, а не на его сущность. Очень часто под трендом понимают детерминированную составляющую процесса, что значительно обедняет содержание термина и препятствует его применению для анализа временных рядов в общем случае.

Компоненты временного ряда ξ_t и ε_t ненаблюдаемы. Они являются теоретическими величинами. Их выделение и составляет предмет анализа временного ряда в задаче прогнозирования. Оценку будущих членов ряда обычно делают по прогнозной модели. Прогнозная модель — это модель, аппроксимирующая тренд. Прогнозы — это оценки будущих уровней ряда, а последовательность прогнозов для различных периодов упреждения $\tau = 1, 2, ..., k$ составляет оценку тренда.

При построении прогнозной модели выдвигается гипотеза о динамике величины ξ_t, т.е. о характере тренда. Однако в связи с тем, что уверенность в гипотезе всегда относительна, рассматриваемые модели наделяются адаптивными свойствами, способностью к корректировке исходной гипотезы или даже к замене ее другой, более адекватно (с точки зрения точности прогнозов) отражающей поведение реального ряда.

Простейшая адаптивная модель основывается на вычислении так называемой экспоненциальной средней (по Р. Брауну – движущейся средней (moving average)).

4.2 Экспоненциальное сглаживание

Предположим, что исследуется временной ряд x_t. Выявление и анализ тенденции динамического ряда часто производится с помощью его выравнивания или сглаживания. Экспоненциальное сглаживание - один из простейших и распространенных приемов выравнивания ряда. В его основе лежит расчет экспоненциальных средних.

Экспоненциальное сглаживание ряда осуществляется по рекуррентной формуле, по существу описывающей марковский итерационный процесс аппроксимации исходного ряда x_t:

$$S_t = \alpha x_t + \beta S_{t-1}, \tag{4.1}$$

где S_t — значение экспоненциальной средней в момент *t;*

α — параметр сглаживания, $\alpha = const$, $0 < \alpha < 1$; $\beta = 1 - \alpha$.

Выражение (4.1) можно переписать следующим образом:

$$S_t = \alpha x_t + (1-\alpha) S_{t-1} = S_{t-1} + \alpha (x_t - S_{t-1}). \tag{4.2}$$

Экспоненциальная средняя на момент *t* здесь выражена как экспоненциальная средняя предшествующего момента плюс доля α-разницы текущего наблюдения и экспоненциальной средней прошлого момента (moving average).

Если последовательно использовать реккурентное соотношение (4.1), то экспоненциальную среднюю S_t можно выразить через значения временного ряда *x:*

$$\begin{aligned} S_t &= \alpha x_t + \beta S_{t-1} = \alpha x_t + \alpha\beta x_{t-1} + \beta^2 S_{t-2} = \ldots = \\ &= \alpha x_t + \alpha\beta x_{t-1} + \alpha\beta^2 x_{t-2} + \ldots + \alpha\beta^i x_{t-i} + \ldots + \beta^N S_0 = \\ &= \alpha \sum_{i=0}^{N-1} \beta^i x_{t-i} + \beta^N S_0, \end{aligned} \tag{4.3}$$

где N — количество членов ряда; S_o — некоторая величина, характеризующая начальные условия для первого применения формулы (4.1) при $t = 1$.

Так как $\beta < 1$, то при $N \to \infty$, $\beta^N \to 0$, а сумма коэффициентов $\alpha \sum_{i=0}^{N-1} \beta^i \to 1$.

Тогда

$$S_i = \alpha \sum_{i=0}^{\infty} \beta^i x_{t-i}.$$

Таким образом, величина S_t оказывается взвешенной суммой всех членов ряда. Причем веса падают экспоненциально в зависимости от давности («возраста») наблюдения. Это и объясняет, почему величина S_t названа экспоненциальной средней. Если, например α = 0.3, то текущее наблюдение будет иметь вес 0.3, а веса предшествующих данных составят соответственно 0,21; 0,147; 0,103 и т.д.

Рассмотрим, ряд, сгенерированный моделью

$$x_t = a_1 + \varepsilon_t,$$

где $a_1 = const$; $\varepsilon_t -$ случайные неавтокоррелированные отклонения, или шум со средним значением 0 и дисперсией σ^2. Применим к нему процедуру экспоненциального сглаживания (4.1). Тогда

$$S_t = \alpha \sum_{i=0}^{\infty} \beta^i x_{t-i} = \alpha \sum_{i=0}^{\infty} \beta^i (a_1 + \varepsilon_{t-i}) =$$

$$= a_1 + \alpha \sum_{i=0}^{\infty} \beta^i \varepsilon_{t-i} \quad .$$

Найдем математическое ожидание

$$M(S_t) = M(x_t) = a_1$$

и дисперсию

$$D(S_t) = M[(S_t - a_1)^2] = M\left[\left(\alpha \sum_{i=0}^{\infty} \beta^i \varepsilon_{t-i}\right)^2\right] = \quad . \qquad (4.4)$$

$$= \alpha^2 \sum_{i=0}^{\infty} \beta^{2i} \sigma^2 = \frac{\alpha}{2-\alpha} \sigma^2$$

Так как $0 < \alpha < 1,$ то $D(S_t) < D(x_t) = \sigma^2$.

Таким образом, экспоненциальная средняя S_t имеет то же математическое ожидание, что и ряд x, но меньшую дисперсию. Как видно из (4.4), при высоком значении α дисперсия экспоненциальной средней незначительно отличается от дисперсии ряда x_t. Чем меньше α, тем в большей степени сокращается дисперсия экспоненциальной средней. Следовательно, экспоненциальное сглаживание можно рассматривать как фильтр, на вход которого в виде потока последовательно поступают члены исходного ряда, а на выходе формируются текущие значения экспоненциальной средней. И чем меньше α', тем в большей степени фильтруются, подавляются колебания исходного ряда.

После появления работ Р. Брауна [92,93] экспоненциальная средняя часто используется для краткосрочного прогнозирования. В этом случае предполагается, что ряд генерируется моделью

$$x_t = a_{1,t} + \varepsilon_t \,,$$

где $a_{1,t}$ - варьируемый во времени средний уровень ряда; ε_t – случайные неавтокоррелированные отклонения с нулевым математическим ожиданием и дисперсией σ^2.

Прогнозная модель (предиктор) имеет вид

$$\hat{x}_\tau(t) = \hat{a}_{1,t} \,,$$

где $\hat{x}_\tau(t)$ — прогноз, сделанный в момент t на τ единиц времени (шагов) вперед; $\hat{a}_{1,t}$ — оценка $a_{1,t}$ (знак $\wedge$ над величиной здесь и далее будут означать оценку).

Оценкою единственного параметра модели служит экспоненциальная средняя $\hat{a}_{1,t} = S_t$. Таким образом, все свойства экспоненциальной средней распространяются на прогнозную модель. В частности, если S_{t-1} рассматривать как прогноз на 1 шаг вперед, то в выражении (4.2) величина $(x_t - S_{t-1})$ есть погрешность этого прогноза, а новый прогноз S_t получается в результате корректировки предыдущего прогноза с учетом его ошибки. В этом и состоит существо адаптации как процесса уточнения последую-

щей точки прогноза на основании текущей информации о его предыдущем значении.

При краткосрочном прогнозировании желательно как можно быстрее отразить изменения $a_{1,t}$ и в то же время как можно лучше «очистить» ряд от случайных колебаний.

Таким образом, с одной стороны, следует увеличивать вес более свежих наблюдений, что может быть достигнуто повышением α (см. (4.3)), с другой стороны, для сглаживания случайных отклонений величину α нужно уменьшить. Как видим, эти два требования находятся в противоречии. Поиск компромиссного значения α составляет задачу оптимизации модели.

Для уяснения процедуры расчета экспоненциальной средней и ее свойств рассмотрим числовой пример сглаживания ряда курса акций фирмы IBM (см. табл. 4.1).

Определим S_0 как $\frac{1}{5}\sum_{i=1}^{5} x_i = \frac{1}{5}(510+497+504+510+509) = 506$. Дальнейшее вычисления при $\alpha = 0,1$ выглядит следующим образом:

$$S_1 = \alpha x_1 + (1-\alpha)S_0 = 0,1 \cdot 510 + 0,9 \cdot 506 = 506,4;$$
$$S_2 = \alpha x_2 + (1-\alpha)S_1 = 0,1 \cdot 497 + 0,9 \cdot 506,4 = 505,46$$
$$S_3 = \alpha x_3 + (1-\alpha)S_2 = 0,1 \cdot 504 + 0,9 \cdot 505,46 = 505,31$$

и т.д.

Результаты вычислений экспоненциальных средних при $\alpha = 0,1$, $\alpha = 0,5$ и $\alpha = 0,9$ приведены в табл. 4.1.

На рис. 4.1 изображен график динамики временного ряда и экспоненциальных средних при $\alpha = 0,1$ и $\alpha = 0,5$. На графике наглядно проявляется влияние величины α на подвижность экспоненциальной средней.

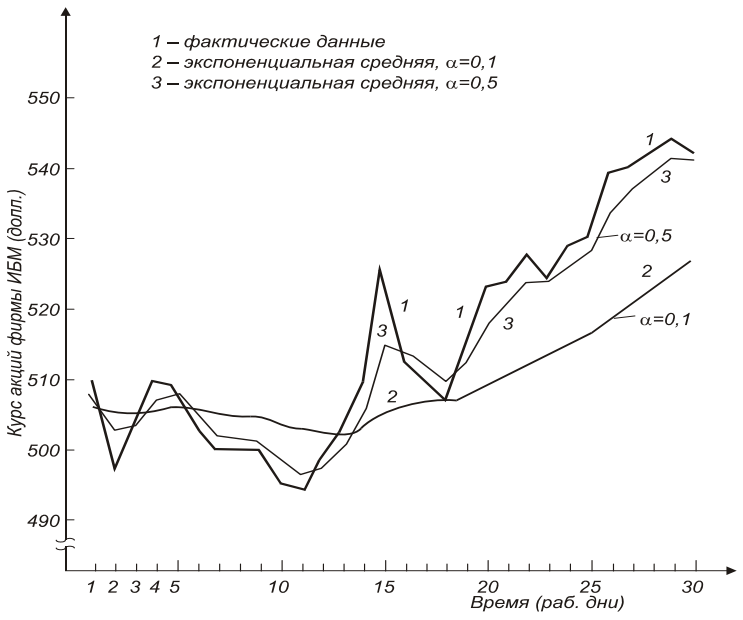


Рис 4.1 - Экспоненциальное сглаживание временного ряда

Таблица 4.1

*Экспоненциальные средние**

№ точки (время)	Члены ряда**	*a=0.1*	*a=0.5*	*a=0.9*	№ точки (время)	Члены ряда**	*a=0.1*	*a=0.5*	*a=0.9*
1	510	506,4	508,0	509,6	16	512	505,7	513,3	513,1
2	497	505,5	502,5	498,3	17	510	506,1	511,7	510,3
3	504	505,3	503,2	503,4	18	506	506,1	508,8	506,4
4	510	505,8	506,6	509,3	19	515	507,0	511,9	514,1
5	509	506,1	507,8	509,0	20	522	508,5	517,0	521,2
6	503	505,8	505,4	503,6	21	523	509,9	520,0	522,8
7	500	505,2	502,7	500,4	22	527	511,6	523,5	526,6
8	500	504,7	501,4	500,0	23	523	512,8	523,2	523,4
9	500	504,2	500,7	500,0	24	528	514,3	525,6	527,5
10	495	503,3	497,8	495,5	25	529	515,8	527,3	528,9
11	494	502,4	495,9	494,2	26	538	518,0	532,7	537,1
12	499	502,0	497,5	498,5	27	539	520,1	525,8	538,8
13	502	502,0	499,7	501,2	28	541	522,2	538,4	540,8
14	509	502,7	504,4	508,3	29	543	524,3	540,7	542,8
15	525	505,0	514,7	523,3	30	541	525,9	540,9	541,2

* Во всех случаях начальное значение экспоненциальной средней было принято равным $\frac{1}{5}\sum_{i=1}^{5} x_i = 506$

** Свойства сглаживания особенно наглядно проявляются при значительной зашумленности исходных данных. В связи с этим для иллюстрации взят один из показателей, отражающий конъюнктурные колебания американской экономики, — курс акций фирмы IBM, производящей компьютеры. Этот временной ряд уже использовался для испытания некоторых адаптивных моделей Р.Г. Брауном, из работы которого [92] он и взят.

Экспоненциальное сглаживание является простейшим вариантом самообучающейся модели. Вычисления просты и выполняются итеративно. Они требуют даже меньше арифметических операций, чем скользящая средняя, а массив прошлой информации уменьшен до одного значения S_{t-1}. Такую модель будем называть адаптивной экспоненциального типа, а величину α — параметром адаптации. Исследуем ее свойства.

4.3 Начальные условия экспоненциального сглаживания

Экспоненциальное выравнивание всегда требует предыдущего значения экспоненциальной средней. Когда процесс только начинается, должна быть некоторая величина S_0, которая может быть использована в качестве значения, предшествующего S_1. Если есть прошлые данные к моменту начала выравнивания, то в качестве начального значения S_0 можно использовать арифметическую среднюю всех имеющихся точек или какой-то их части. Когда для такого оценивания S_0 нет данных, требуется предсказание начального уровня ряда.

Предсказание может быть сделано исходя из априорных знаний о процессе или на основе его аналогии с другими процессами. После k шагов вес, придаваемый начальному значению, равен $(1-\alpha)^k$. Если есть уверенность в справедливости начального значения S_0, то можно коэффициент α взять малым. Если такой уверенности нет, то параметру α следует дать большое значение, с таким расчетом, чтобы влияние начального значения быстро уменьшилось. Однако большое значение α, как это следует из (4.4), может явиться причиной большой дисперсии колебаний S_t. Если требуется подавление этих колебаний, то после достаточного удаления от начального момента времени величину α можно уменьшить.

Рассмотрим роль параметра α в начальный период сглаживания в случае, когда нет уверенности в справедливости выбора начальной величины S_0.

Таблица 4.2

Изменение весов в начальный период времени при экспоненциальном сглаживании с α=0,1

Итерация	Вес начальной величины	Вес первого члена ряда	Вес второго члена ряда	Вес третьего члена ряда	Вес четвертого члена ряда
1	0,900	0,100			
2	0,810	0,090	0,100		
3	0,729	0,081	0,090	0,100	
4	0,656	0,073	0,081	0,090	0,100

Как видно из таблицы 4.2, составленной для значения *α=0,1* начальная величина S_0 в течение длительного времени имеет чрезмерный вес. Даже после 20 итераций вес S_0 равен 0,122, что означает, что ему дается все еще больший вес, чем любому другому члену ряда. Таким образом, в этом случае получение прогнозов по экспоненциальной средней, построенной на малом отрезке ряда (выборке), чревато большими ошибками. Для того чтобы элиминировать избыточный вес, приданный начальной величине S_0, Р. Вейд [94] предложил модифицировать процедуру сглаживания следующим образом.

Для исходного момента времени задаётся

$$S'_0 = \alpha S_0,$$

$$S'_1 = \alpha x_1 + (1-\alpha)S'_0 = \alpha x_1 + \alpha(1-\alpha)S_0,$$

где S_0— как и раньше, начальная оценка уровня ряда.

Так как коэффициенты α и $\alpha(1-\alpha)$ в сумме теперь не дают 1, то следует использовать множитель, равный единице, деленной на сумму коэффициентов. Таким образом, модифицированной экспоненциальной средней для t=1 будет

$$\tilde{S}_1 = S'_1 \frac{1}{\alpha + \alpha(1-\alpha)} = [\alpha x_1 + (1-\alpha)S'_0] \frac{1}{\alpha + \alpha(1-\alpha)}$$

и вообще

$$\widetilde{S}_t = S'_t \frac{1}{\sum_{i=0}^{t} \alpha(1-\alpha)^i} = [\alpha x_t + (1-\alpha)S'_{t-1}] \frac{1}{\sum_{i=0}^{t} \alpha(1-\alpha)^i}.$$

Из табл. 4.3 можно видеть, что сущность этого метода состоит в том, чтобы убрать избыточный вес от веса, даваемого начальному значению S_0, и распределить его пропорционально по всем членам ряда. Прогнозы, получаемые по соответствующей модифицированной модели, основываются

Таблица 4.3

Изменение весов в начальный период времени при $\alpha = 0{,}1$ *в модифицированной модели*

Итерация	Вес начальной величины	Вес первого члена ряда	Вес второго члена ряда	Вес третьего члена ряда	Вес четвертого члена ряда
1	0,474	0,526			
2	0,299	0,332	0,369		
3	0,212	0,236	0,262	0,291	
4	0,160	0,178	0,198	0,220	0,224

в большей степени на фактических данных, чем на предварительной оценке S_0 даже при малых выборках. Для того чтобы сократить время вычислений, целесообразно вернуться к обычному экспоненциальному сглаживанию, когда сумма коэффициентов $\sum_{i=0}^{t}\alpha(1-\alpha)^i$ приближается к 1. На основе эмпирического анализа рекомендуется осуществлять такой переход при сумме коэффициентов 0,995. При заданном значении α можно заранее определить, на каком шаге следует вернуться к обычной модели.

4.4 Выбор постоянной сглаживания

Выбору величины постоянной сглаживания следует уделять особое внимание. Поиски должны быть направлены на отыскание оснований для выбора наилучшего значения. Нужно учитывать условия, при которых эта величина должна принимать значения, близкие то одному крайнему значению, то другому. Нетрудно заметить, что при $\alpha = 0$, $S_t = S_0$ - наблюдается случай абсолютной фильтрации и полного отсутствия адаптации, а при $\alpha = 1$ приходим к так называемой наивной модели $\hat{x}_\tau(t) = S_t = x_t$, в соответствии с которой прогноз на любой срок равен текущему, фактическому значению ряда. На практике эта модель из-за простоты пользуется особой популярностью.

В подразделе 4.2 уже отмечалось, что постоянная сглаживания характеризует скорость реакции модели $\hat{x}_\tau(t) = S_t$ на изменения уровня процесса, но одновременно определяет и способность системы сглаживать случайные отклонения. Поэтому величине α следует давать то или иное промежуточное значение между 0 и 1 в зависимости от конкретных свойств динамического ряда.

В качестве удовлетворительного компромисса рекомендуется брать ее в пределах от 0,1 до 0,3. Эта рекомендация некритически повторена в ряде работ. Между тем в [95] показано, что наилучшие результаты получаются при α = 0,9. Однако, как правило, если в результате испытаний обнаружено, что наилучшее значение константы α близко к 1, следует проверить законность выбора модели данного типа. Часто к большим значениям α приводит наличие в исследуемом ряде ярко выраженных тенденций или сезонных колебаний (высокая персистентность процесса, когда показатель Хёрста близок к 1 [11,22]). В этом случае для получения эффективных прогнозов требуется другая модель.

Ясно, что наилучшее значение α в общем случае должно зависеть от срока прогнозирования τ. Для конъюнктурных прогнозов в большей мере должна учитываться свежая информация. При увеличении периода упреждения τ более поздняя информация, отражающая последнюю конъюнктуру, должна, по-видимому, иметь несколько меньший вес, чем в случае малых τ. Для того чтобы сгладить конъюнктурные колебания, следует в

большей мере учитывать информацию за прошлые периоды времени. Для проведения подобного анализа вводят *понятие среднего возраста данных*. Возраст текущего наблюдения равен 0, возраст предыдущего наблюдения равен 1 и т. д. Средний возраст — это сумма взвешенных возрастов данных, использованных для подсчета сглаженной величины. Причем возраст имеют те же веса, что и соответствующая информация. При экспоненциальном выравнивании вес, даваемый точке с возрастом k равен $\alpha\beta^k$, где $\beta = 1-\alpha$ и средний возраст информации равен:

$$k = 0\cdot\alpha + 1\cdot\alpha\beta + 2\cdot\alpha\beta^2 + \ldots = \alpha\sum_{k=0}^{\infty} k\beta^k = \frac{\beta}{\alpha}.$$

Таким образом, чем меньше α, тем больше средний возраст (степень забывания) информации. Для конъюнктурных прогнозов значение α , как правило, надо брать большим, а для более долгосрочных — малым. Это положение иллюстрирует рис. 4.2, на котором отображена зависимость стандартной ошибки прогнозирования, обычно принимаемой за показатель точности от α. Однако характер зависимостей, аналогичных тем, что отражены на рисунке, следует изучать специально в каждом конкретном случае.

Теоретический анализ проблемы выбора постоянной сглаживания при применении простейшей экспоненциальной модели для прогнозирования стационарного процесса с функцией вида $p_k = p_1^k$, где p_1 - коэффициент автокорреляции при лаге $k = 1$, проведен Д. Р. Коксом [97] и Дж. Д. Кохеном [96].

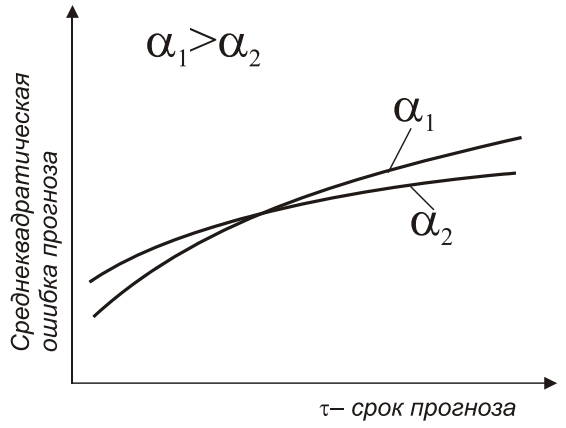


Рис. 4.2. - Примерная зависимость среднеквадратической ошибки прогноза от величины постоянной сглаживания α и периода упреждения τ

Показано, что минимум среднего квадрата ошибки при прогнозировании такого ряда на 1 шаг вперед $(\tau = 1)$ будет при

$$a_{опт} = \begin{cases} \dfrac{3p_1 - 1}{2p_1} & 1/3 < p_1 \le 1; \\ 0 & -1 \le p_1 \le 1/3. \end{cases} \qquad (4.5)$$

Соответствующая дисперсия ошибки прогноза при этом:

$$D_e = \begin{cases} \dfrac{8p_1(1-p_1)}{(1+p_1)^2}\sigma_x^2 & 1/3 < p_1 \le 1; \\ \sigma_x^2 & -1 \le p_1 \le 1/3. \end{cases}$$

Табл. 4.4 показывает соотношения между p_1 , $\alpha_{опт}$ и точностью прогнозирования на 1 шаг вперед.

Табл. 4.5 показывает, что для данной p_1 величина D_e при $\tau = 1$ слабо зависит от α, так что точность прогноза в некоторой окрестности $\alpha_{опт}$ нечувствительна к выбору постоянной сглаживания.

Результат (4.5) означает, что если $p_1 > 1/3$, то при соответствующем выборе величины α экспоненциальная средняя в определенной степени отражает колебания, связанные с сильной автокорреляцией.

Таблица 4.4

Соотношения между $p_1, \alpha_{опт}, D_e/\sigma_x^2$ *при прогнозировании стационарного процесса с* $p_k = p_1^k$ *по модели экспоненциального сглаживания* $(\tau = 1)$

p_1	$\alpha_{опт}$	D_e/σ_x^2	p_1	$\alpha_{опт}$	D_e/σ_x^2
$\le 1/3$	0	1	0,7	0,786	0,581
0,4	0,250	0,980	0,8	0,875	0,395
0,5	0,500	0,889	0,9	0,944	0,199
0,6	0,667	0,750	0,95	0,974	0,100

Таблица 4.5

	D_e/σ_x^2 *как функция от* α *при* $p_1 > 1/3$						
α	$p_1 = 0,4$	$p_1 = 0,7$	$p_1 = 0,9$	α	$p_1 = 0,4$	$p_1 = 0,7$	$p_1 = 0,9$
1	1,200	0,600	0,200	0,4	0,987	0,647	0,272
0,9	1,136	0,587	0,200	0,3	0,980	0,692	0,318
0,8	1,087	0,581	0,203	0,2	0,980	0,758	0,397
0,7	1,049	0,584	0,211	0,1	0,987	0,853	0,554
0,6	1,020	0,595	0,223	0	1,000	1,000	1,000
0,5	1,000	0,615	0,242	$a_{опт}$	0,980	0,581	0,200

С другой стороны, если $p_1 \le 1/3$, то наибольшее, что может дать простейшая модель, это оценка среднего уровня, вокруг которого варьирует процесс. В то же время на практике при $p_1 \le 1/3$ не следует брать a слишком малым, иначе предиктор окажется нечувствительным к изменениям среднего уровня (тренда).

Определенным руководством при этом может служить табл. 4.6, которая характеризует дисперсии ошибок, получаемых при прогнозировании стационарных процессов с $p_k = p_1^k$, где $p_1 \le 1/3$. Из таблицы видно, что при

$p_1 < 0$, можно добиться немногого, полагая a меньше 0,1…0,2. Вообще говоря, очевидно, что если $p_1 < 0$, то простейшая модель экспоненциального типа не является хорошим предиктором.

Если $\tau > 1$, то существенно повышается критическая величина $p_{1крит}$, ниже которой оптимальное значение α равно 0. Этот факт иллюстрирует табл. 4,7.

Таблица 4.6

Дисперсия ошибки прогноза для стационарного процесса с $p_k = p_1^k$, где $p_1 \le 1/3, \tau = 1$

α	D_e / σ_x^2				
	$p_1 = 1/3$	$p_1 = 1/10$	$p_1 = 0$	$p_1 = -1/4$	$p_1 = -1/2$
0	1,000	1,000	1,000	1,000	1,000
0,05	1,001	1,020	1,026	1,036	1,043
0,10	1,002	1,041	1,053	1,074	1,089
0,20	1,011	1,087	1,111	1,157	1,190
0,30	1,022	1,139	1,176	1,252	1,307
0,40	1,042	1,197	1,250	1,359	1,442
0,50	1,067	1,263	1,333	1,481	1,600

Таблица4.7

Зависимость $p_{1крит}$ от τ

τ	1	2	3
$p_{1крит}$	0,333	0,516	0,821

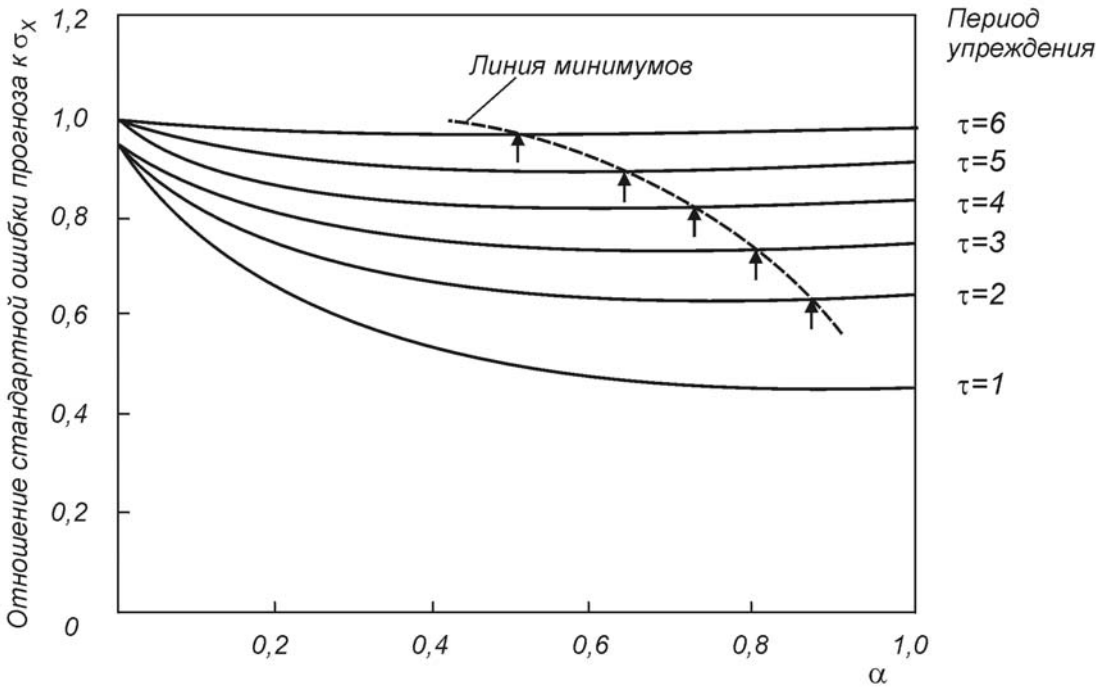


Рис. 4.3 - Влияние α на точность прогнозирования при однократном экспоненциальном сглаживании данных с $p_k = 0,9^{|k|}$

Рис. 4.3 показывает стандартную ошибку прогнозирования для всех значений постоянной сглаживания в случае стационарного процесса с сильной автокорреляцией $p_k = 0,9^{|k|}$, т.е. автоковариацией $R_{xx}(k) = \sigma_x^2(0,9)^{|k|}$. При этом пунктирная линия выделяет геометрическое место точек решений, которые минимизируют ошибку прогнозирования. Отсюда можно сделать вывод, что если данные сильно коррелированны и период упреждения τ мал, то сглаживать не стоит, а целесообразно в качестве прогноза использовать наиболее позднее наблюдение.

С другой стороны, можно показать, что экспоненциальное сглаживание является эффективным средством выделения тренда (рис. 4.4). Используя *m-file*, приведенный ниже, можно поэкспериментировать с выбором коэффициента сглаживания *α, и визуально оценить его влияние на точность прогнозирования.*

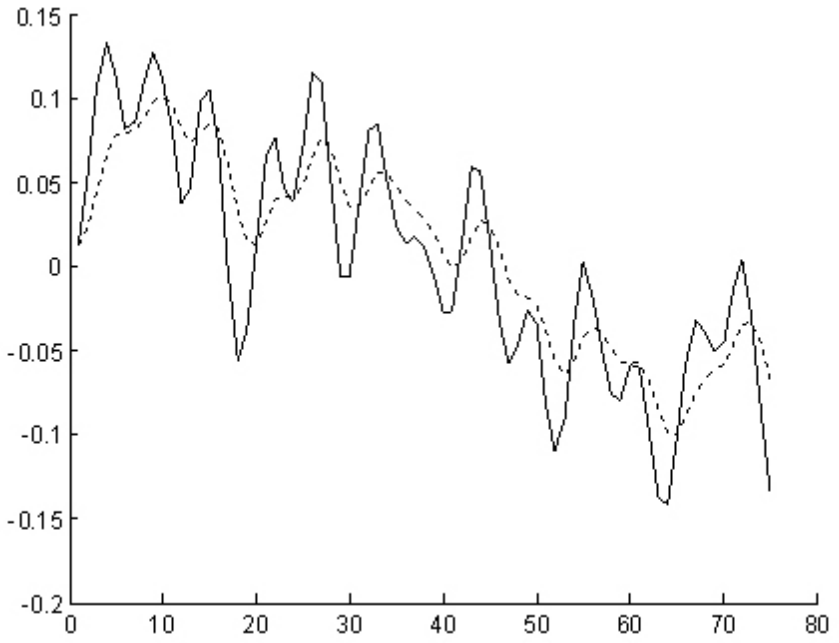


Рис. 4.4

```
function trend(X)     % Обращение к программе в среде Matlab:
   sizeX=length(X); % >> trend(sig(1:75));
   t=1;                  % временной ряд X задан файлом sig.mat
     Alpha=0.3;        % Alpha = α
   S(t)= Alpha *X(t+1);
      while ( t < sizeX )
   S(t+1)=(S(t)+(Alpha *(X(t+1)-S(t))));
   t=t+1;
end;
     hold on;
     plot(X);
     plot(S,[':','r']);
hold off
```

4.5 Вопросы формирования ансамблей моделей-предикторов

В настоящее время существует достаточно широкий спектр инструментов интеллектуального анализа данных для решения задач прогнозирования: от традиционных методов, подобных рассмотренному выше методу экспоненциального среднего (движущегося среднего по Р. Брауну), классических методов статистического анализа, рассмотренных в разделе II, до современных интеллектуальных методов обработки данных, использующих деревья решений, нейро- и нейро-фаззи сети, логистическую регрессию и т. д.

Вместе с тем разнообразие алгоритмов извлечения знаний (Data Mining) говорит о том, что не существует одного универсального метода для решения всех задач. Кроме того, применение различных инструментов анализа и моделирования к одному и тому же набору данных может преследовать разные цели: либо построить упрощенную, прозрачную, легко интерпретируемую модель в ущерб точности, либо построить более точную, но и более сложную, а стало быть, менее интерпретируемую модель.

Таким образом, одной из актуальных задач современного подхода к обработке данных и в том числе к прогнозированию является нахождение компромисса между такими показателями, как точность, сложность и интерпретируемость.

Большинство исследователей предпочитают получение более точных результатов, так как для конечных пользователей понятие прозрачности субъективно. Точность результатов зависит от качества исходных данных, предметной области и используемого метода анализа данных.

Получение более точных результатов тем более актуально, поскольку в последние годы значительно возрос интерес к точности моделей Data Mining, основанных на интеллектуальных методах обучения, за счет объединения усилий нескольких методов и создание ансамблей моделей-предикторов, что позволяет повысить качество решения аналитических задач. Под обучением ансамбля моделей понимается процедура обучения конечного набора базовых классификаторов, результаты прогнозирования которых затем объединяются и формируется прогноз агрегированного классификатора.

При формировании ансамбля моделей необходимо решить три основные задачи:

- выбрать базовую модель;
- определить подход к использованию обучающего множества;
- выбрать метод комбинирования результатов.

В силу того, что ансамбль - это агрегированная модель, состоящая из отдельных базовых моделей, то при его формировании возможны две альтернативы:

- ансамбль составляется из базовых моделей одного типа, например, только из деревьев решений, только из нейронных сетей и т. д.;
- ансамбль составляется из моделей различного типа - деревьев решений, нейронных сетей, регрессионных моделей и т. д.

С другой стороны при построении ансамбля используется обучающее множество, для использования которого существуют два подхода:

- перевыборка, т. е. из исходного обучающего множества извлекается несколько подвыборок, каждая из которых используется для обучения одной из моделей ансамбля;
- использование одного обучающего множества для обучения всех моделей ансамбля.

В свою очередь, для комбинирования результатов, выданных отдельными моделями, используют три основных способа:

- голосование - выбирается тот класс, который был выдан простым большинством моделей ансамбля;
- взвешенное голосование - для моделей ансамбля устанавливаются весы, с учетом которых выносится результат;
- усреднение (взвешенное или невзвешенное) - выход всего ансамбля определяется как простое среднее значение выходов всех моделей, при взвешенном усреднении выходы всех моделей умножаются на соответствующие веса.

Очевидно, что приведенное выше иллюстрирует частный случай применения методов многокритериального анализа, в частности, метода анализа иерархий Т. Саати. При этом в более полном объёме с вопросами агрегирования моделей и многокритериального анализа в условиях ограниченности альтернатив можно ознакомиться на сайте: http://nootron.net.ua/, разработанном на кафедре информационных технологий и систем Национальной металлургической академии Украины.

Вместе с тем, исследования в области синтеза ансамблей моделей в ИАД (Data Mining) стали проводиться относительно недавно и имеют свои особенности и свою терминологию. К настоящему времени в ИАД разработано уже множество различных методов и алгоритмов формирования ансамблей моделей, среди которых наибольшее распространение получили такие как: беггинг (bagging), бустинг (boosting) и стэкинг (stacking): http://arbir.ru/articles/a_4053.htm .

Алгоритм беггинга

Главная идея бэггинга в реализации параллельного обучения на нескольких различных выборках одинакового размера, полученных путем случайного отбора примеров из исходного набора данных. Алгоритм беггинга подразумевает следующие шаги. Сначала формируется несколько выборок путем случайного отбора из исходного множества данных. Затем на основе каждой выборки строится классификатор, и выходы всех классификаторов агрегируются с использованием голосования или простого усреднения. Очевидно, что точность предсказания построенных с помощью беггинга комбинированных предикторов значительно выше, чем точность отдельных моделей.

Алгоритм бустинга

Основная идея бустинга заключается в построении цепочки моделей, при этом каждая следующая обучается на примерах, на которых предыдущая модель допустила ошибку. По сравнению с беггингом бустинг является более сложной процедурой, но во многих случаях работает эффективнее. Бустинг начинает создание ансамбля на основе единственного исходного множества, но в отличии от беггинга каждая новая модель строится на основе результатов предыдущей, т. е. модели строятся последовательно. Бустинг создает новые модели таким образом, чтобы они дополняли ранее построенные, выполняли ту работу, которую другие модели сделать не смогли на предыдущих шагах. И наконец, последнее отличие бустинга от бэггинга заключается в том, что всем построенным моделям в зависимости от их точности присваиваются веса. Бустинг-алгоритм по сути относится к итерационным алгоритмам индуктивного моделирования [80] (по существу МГУА-подобным, см. раздел V). Он учится распознавать примеры на границах классов. Каждой записи данных на каждой итерации алгоритма присваивается вес. Первый классификатор обучается на всех примерах с равными весами. На каждой последующей итерации веса расставляются соответственно классифицированным примерам, т. е. веса правильно классифицированных примеров уменьшаются, а неправильно классифицированных - увеличиваются. Следовательно, приоритетными для следующего классификатора станут неправильно распознанные примеры, обучаясь на которых новый классификатор будет исправлять ошибки классификатора на прошлой итерации.

Алгоритм стэкинга

Стэкинг - один из способов создания составных моделей. Данный метод был разработан недавно, поэтому менее известен, чем беггинг и бустинг. Отчасти это связано со сложностью теоретического анализа, а отчасти с тем, что общая концепция использования данного метода пока отсутствует - основная идея может применяться в самых разнообразных вариантах. В отличие от бэггинга и бустинга стэкинг обычно применяется к моделям,

построенным с помощью различных алгоритмов, обучаемых на одинаковых данных. Стэкинг вводит концепцию метаобучения, т. е. пытается обучить каждый классификатор, используя алгоритм метаобучения, который позволяет обнаружить лучшую комбинацию выходов базовых моделей.

Конрольные вопросы к разделу V:

1. Привести алгоритм экспоненциального сглаживания.
2. Указать свойства алгоритма экспоненциального сглаживания.
3. В чем состоит выбор экспоненциальной средней?
4. Влияние срока прознозирования на ошибку прогноза.
5. От чего зависит выбор величины постоянной сглаживания?
6. В чём состоит проблема выбора коэффициента адаптации α?
7. Можно ли рассматривать предиктор экспотенциального сглаживания как фильтр?
8. Как влияет на точность прогноза персистентность или сезонность процесса?
9. Что такое беггинг?
10. Что такое бустинг?
11. Что такое стэкинг?
12. Используя листинг программы, приведенной на стр. 172, провести прогнозирование данных на 1, 2 и 3 шага вперед. Оценить точность прогноза.

РАЗДЕЛ V

ДИНАМИЧЕСКИЙ ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Динамический интеллектуальный анализ данных (Dynamic Data Mining - DDM) объединяет в себе методы стандартного интеллектуального анализа данных с методами вычислительного интеллекта для решения задач прогнозирования, сегментации, компрессии, эмуляции и идентификации, а также обнаружения разладок нестационарных нелинейных сигналов, наблюдения которых поступают в последовательном (on-line) режиме. Рассмотрим по-подробнее задачи и проблемы, которые существуют в динамическом интеллектуальном анализе данных [130, 143].

5.1. Компрессия больших массивов данных и временных рядов

Выделим в качестве основных следующие типовые прикладные задачи снижения размерности анализируемого признакового пространства, которые рассматриваются в рамках анализа данных большого объема:

1. Отбор наиболее информативных признаков (включая выявление латентных факторов). Речь идет об отборе из исходного (априорного) множества признаков $X = (x_1, x_2, \ldots, x_p)$, которые обладали бы свойством наибольшей информативности в смысле, определенном, как правило, некоторым специально подобранным для каждого конкретного типа задач критерием обучения. Так, например, если критерий обучения направлен на достижение максимальной точности статистического прогноза некоторого результирующего количественного показателя по известным значениям предыстории, то речь идет о наилучшем подборе наиболее существенных предикторов моделей регрессии. Если же критерий обучения устроен таким образом, что его оптимизация обеспечивает наивысшую точность решения задачи отнесения объекта к одному из классов по значениям X его описательных признаков, то это является задачей построения системы типообразующих признаков в задаче классификации или задачей выявления и интерпретации некоторой латентной характеристики изучаемого свойства. Наконец, критерий обучения может быть нацелен на максимальную автоинформативность новой системы показателей, т.е. на максимально точное воспроизведение всех исходных признаков по сравнительно небольшому числу вспомогательных переменных.

2. Сжатие (компрессия) массивов обрабатываемой и хранимой информации. Такой вид задач связан с рассмотренными выше и, в частности, требует в качество одного из основных приемов решения построения экономной системы вспомогательных признаков, обладающих наивысшей автоинформативностью. В действительности при решении достаточно серьезных задач сжатия больших массивов информации используется сочета-

ние методов классификации и снижения размерности. Методы классификации позволяют перейти от массива, содержащего информацию по всем n статистически обследованным объектам, к соответствующей информации только по k эталонным образцам $(k << n)$, где в качестве эталонных образцов берутся специальным образом отобранные наиболее типичные представители классов, полученных в результате операции разбиения исходного множества объектов на однородные группы. Методы же снижения размерности позволяют заменить исходную систему показателей набором вспомогательных (наиболее автоинформативных) переменных.

3. Визуализация данных. Данная задача дает ответ на вопрос еще на предварительной стадии анализа данных распадается ли анализируемая выборка на четко выраженные классы или кластеры в заданном пространстве, каково примерное их число и т.д. Так как максимальный размер фактически осязаемого пространства равен трем, поэтому естественно возникает проблема проецирования анализируемых многомерных данных из исходного пространства на прямую, плоскость, в крайнем случае - в трехмерное пространство, но так, чтобы интересующие специфические особенности исследуемой совокупности, если они присутствовали в исходном пространстве, сохранились бы и после проецирования. Следовательно, и здесь идет речь о снижении размерности анализируемого признакового пространства, но снижении, во-первых, подчиненном некоторым специальным критериям и, во-вторых, оговоренном условием, что минимальная размерность редуцированного пространства не должна превышать трех.

4. Сжатие временных рядов. Задача такого рода позволяет проводить компрессию многомерных временных рядов, во-первых, с целью анализа, прогноза, эмуляции скомпрессированого процесса, который учитывает все локальные особенности, а, во-вторых, с целью упрощения хранения больших объемов многомерных временных рядов, таких как биомедицинские временные ряды электроэнцефалограмм, кардиограмм, телеметрических данных, различного рода данных промышленных испытаний в металлургии и машиностроении, которые снимаются с нескольких датчиков.

Для решения задач компрессии существует ряд разработанных методов таких, как метод главных компонент, линейный дискриминантный анализ, вэйвлет-анализ, однако эти методы не могут быть применены для решения задач компрессии в on-line режиме, с другой стороны, предложен ряд методов на основе нейронных сетей таких как нейронная сеть "Бутылочное горлышко", нейронная сеть Хэбба-Сэнгера, нейронная сеть Оя-Карунена, нейронная сеть Рубнера-Шультена-Тэвена, но все предложенные нейронные сети не могут применяться для компрессии именно временных рядов.

Попытки синтеза метода сжатия временных рядов с целью их дальнейшего сегментирования предпринимаются, но они основаны, как правило, на методе главных компонент и не применимы в последовательном режиме [101].

5.2. Сегментация нестационарных временных рядов

В последние годы наблюдается рост интереса к задаче анализа нестационарных временных последовательностей, которые изменяют свои свойства в априори неизвестные моменты времени. Проблема анализа временных последовательностей присуща задачам обработки речи, текстов и "web-mining", анализа сенсоров роботов и особенно в медицинской и биологической диагностике. Важно заметить, что эти задачи должны решаться в режиме реального времени при условии поступления новых данных.

В процессе решения указанных выше проблем временная последовательность делится (сегментируется) на внутренне однородные (гомоморфные) части, которые в дальнейшем представляются некоторым более компактным описанием для дальнейшей диагностики или обработки.

Такие задачи в некоторых случаях решаются с использованием подхода, который основывается на методах выявления изменений свойств сигналов и систем. Однако, известные методы, по обыкновению предназначенные для выявления резких изменений, недостаточно хорошо подходят для выявления медленных изменений в характеристиках последовательностей.

В реальных задачах, особенно в биомедицинских применениях, внутренние изменения в наблюдаемом объекте по обыкновению достаточно медленные и стабильные состояния организма перекрываются по многим характеристикам. Больше того, существуют временные переходные состояния, имеющие характеристики, которые относятся одновременно к нескольким стабильным состояниям.

В таких случаях более эффективными являются методы нечеткой кластеризации временных последовательностей, которые основываются на известных методах нечеткого кластерного анализа [102]. Эти методы досказали свою эффективность в решении многих задач в пакетном режиме, однако их использование в задачах реального времени усложняется рядом проблем, некоторые из которых возможно преодолеть, используя быстрые методы рекуррентного кластерного анализа. Однако, указанные выше методы эффективны только при условии, что кластеры, которые пересекаются, являются компактными, то есть они не содержат резких (аномальных) выбросов. В то же время реальные выборки данных но обыкновению содержат до 20% выбросов, а потому предположение о компактности кластеров может быть некорректным.

Для задачи сегментации временной последовательности

$$Y = \{y(1), y(2), \ldots, y(k), \ldots, y(N)\} \tag{5.1}$$

часто используется подход, который базируется на косвенной кластеризации последовательности [103]. Согласно этому подходу выделяются некоторые характеристики для дальнейшего отображения их в преобразованное пространство признаков. В дальнейшем в преобразованном пространстве данных известные методы кластеризации могут использоваться для фор-

мирования кластеров. В качестве характеристик начальной временной последовательности могут быть избраны корреляционные, регрессионные, спектральные и другие характеристики, которые в случае обработки в реальном времени должны быть вычислены с помощью адаптивных процедур.

Для этого могут быть использованы оценки среднего значения, дисперсии, коэффициентов автокорреляции. При этом для придания адаптивных свойств, эти оценки могут быть вычислены с помощью процедуры экспоненциального сглаживания [раздел IV], [98,102]. Среднее значение может быть оценено в форме

$$s(k) = \alpha y(k) + (1-\alpha)s(k-1),\ 0 < \alpha < 1, \tag{5.2}$$

где $\alpha = 2/(T+1)$ - коэффициент, определяющий сглаживание на окне ширины T.

Значение дисперсии временной последовательности также может быть оценено как

$$\sigma^2(k) = \alpha(y(k) - s(k))^2 + (1-\alpha)\sigma(k-1), \tag{5.3}$$

а коэффициенты автокорреляции

$$\rho(k,\tau) = \alpha(y(k) - s(k))(y(k-\tau) - s(k)) + (1-\alpha)\rho(k-1,\tau), \tag{5.4}$$

где $\tau = 1,2,\ldots,\tau_{\max}$ – коэффициенты запаздывания.

Итак, векторы признаков

$$x(k) = (s(k), \sigma^2(k), \rho(k,2), \ldots, \rho(k,\tau_{\max}))^T, \tag{5.5}$$

содержащие $(2+\tau_{\max})$ элементов, вычисляются на каждом шаге дискретного времени k и образуют множество, которое составляется из N n-мерных векторов признаков

$$x(k) = \{x(1), x(2), ..., x(N)\}, \tag{5.6}$$

где $x(k) \in \Re^n, k = 1,2,\ldots,N$.

Результатом применения процедуры нечеткой кластеризации должно быть распределение исходных данных на τ кластеров с некоторой степенью принадлежностей $w_j(k)$ k-го вектора признаков $x(k)$ к j-му кластеру.

Временные последовательности наблюдений, которые идут подряд во времени и принадлежат к одинаковым кластерам, будут образовывать сегменты временной последовательности на выходе.

5.3. Прогнозирование и эмуляция нестационарных нелинейных сигналов

Поскольку в общем случае природа наблюдаемой последовательности неизвестна, наиболее адекватным для прогнозирования в данной ситуации

является применение искусственных нейронных сетей, позволяющих по прошлым наблюдениям восстанавливать нелинейное отображение вида [104-108]

$$x(k) = F(x(k-1), x(k-2), ..., x(k-n_A)) + \\ + e(k) = \hat{x}(k) + e(k), \tag{5.7}$$

где $\hat{x}(k)-$ оценка (прогноз) значения $x(k)$, полученная на выходе нейросети, представляющей в данном случае нелинейную авторегрессионную (NAR) модель; $e(k)$ – ошибка прогнозирования.

Возможность и эффективность использования NAR-модели (Nonlinear AutoRegressive model, нелинейная авторегрессионная модель) (5.7) в задачах прогнозирования определяется теоремой Тэкенса о диффеоморфизме [109], устанавливающей существование порядка модели n_A, который обеспечивает сколь угодно малое значение ошибки $e(k)$, и универсальными аппроксимирующими свойствами ИНС.

В качестве основы для построения NAR-моделей чаще всего используются многослойные сети с прямой передачей информации, входной (нулевой) слой которых образован линиями элементов чистой задержки z^{-1} с отводами.

На рис. 5.1 приведена архитектура многослойной сети.

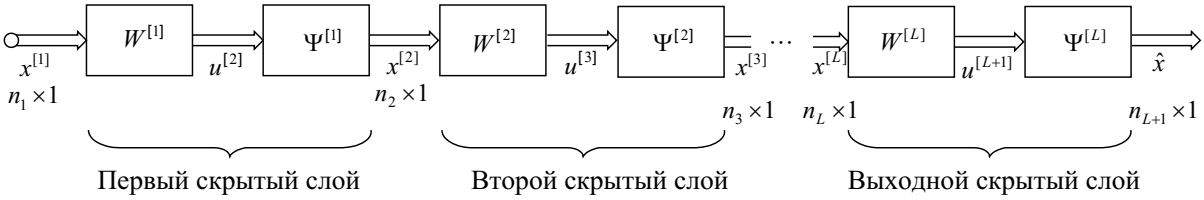


Рис. 5.1 - L-слойная нейронная сеть с прямой передачей информации

На первый скрытый слой сети поступает $n = n_1 = n_A = n_A^{[1]}$ - мерный вектор $x^{[1]}$, сформированный в нулевом слое с помощью элементов задержки z^{-1} и образованный прошлыми значениями прогнозируемого временного ряда $x(k-1), x(k-2), ..., x(k-n_A^{[1]})$. Выходным сигналом первого скрытого слоя является $(n_2 \times 1)$ вектор $x^{[2]}$, подающийся на вход второго скрытого слоя и т.д. На выходе L-го (выходного) слоя появляется прогнозный $m = n_{L+1}$ - мерный вектор $\hat{x}$. Таким образом, каждый слой имеет n_l входов и n_{l+1} выходов и характеризуется $(n_l \times n_{l+1})$ - матрицей синаптических весов $W^{[l]}$ и $(n_{l+1} \times n_{l+1})$ - диагональным оператором $\Psi^{[l]}$, образованным нелинейными активационными функциями $\varphi_j^{[l]}$, $j = 1, 2, \ldots, n_{l+1}$.

"Строительным блоком" такой сети является стандартный статический нейрон, реализующий нелинейное отображение

$$x_j^{[l+1]} = \varphi_j^{[l]}(u_l^{[l+1]}) = \varphi_j^{[l]}\left(\sum_{i=0}^{n_l} w_{ji}^{[l]} x_j^{[l]}\right), \tag{5.8}$$

где $n_l + 1$ синаптических весов $w_{ji}^{[l]}$ которого подлежат уточнению в процессе обучения нейронной сети. Общим недостатком прогнозирующих нейронных сетей на статических нейронах является чрезвычайно большое число настраиваемых весов и низкая скорость обучения, что, естественно, вызывает серьезные проблемы, особенно при работе в реальном времени.

В связи с этим Э. Ваном было предложено [109, 110] в прогнозирующих нейронных сетях вместо статических нейронов использовать их динамические аналоги, у которых синаптические веса образованы цифровыми адаптивными нерекурсивными фильтрами с конечной импульсной характеристикой (КИХ- фильтры, FIR- filters).

Нелинейное отображение, реализуемое динамическим КИХ-нейроном, можно записать в виде

$$x_j^{[l+1]} = \varphi_j^{[l]}(u_l^{[l+1]}) = \varphi_j^{[l]}\left(\sum_{i=0}^{n_l} W_{ji}^{[l]T} X_j^{[l]}(k)\right). \tag{5.9}$$

И хотя динамический нейрон содержит $n_l(n_A^{[l]} + 1) + 1$ параметров, что превышает количество синаптических весов обычного нейрона, сеть, построенная из таких узлов, содержит много меньше параметров, чем стандартная архитектура. на статических нейронах с линиями задержки на входе. В [111] было доказано, что в сети на статических нейронах с линиями задержки количество параметров растет в геометрической зависимости от n_A, в то время как в ИНС на динамических нейронах число настраиваемых синаптических весов есть линейная функция от n_A и L, кроме того такая сеть обладает универсальными динамическими аппроксимирующими свойствами [112]. Для обучения ИНС на динамических нейронах в [110] была введена градиентная процедура, получившая название обратного распространения ошибок во времени (Temporal error backpropagation - TBP). Используя стандартный одношаговый критерий обучения

$$E(k) = \frac{1}{2}\left\|e(k)\right\|^2 = \frac{1}{2}\left\|d(k) - \hat{x}(k)\right\|^2 \tag{5.10}$$

(здесь $d(k)$ – обучающий сигнал, в качестве которого в задачах прогнозирования принимается текущее значение $x(k)$), можно записать в алгоритм его обучения, который обладает сглаживающими свойствами, которые необходимы при обработке "зашумленных" сигналов,

$$\begin{cases} W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) + \dfrac{e_j^{[l+1]}(k) J_i^{[l]}(k)}{\beta_i^{[l]}(k)}, \quad 1 \le l \le L, \\ \beta_i^{[j]}(k+1) = \alpha\beta_i^{[j]}(k) + \left\|J_i^{[l]}(k)\right\|^2, \quad 0 \le \alpha \le 1. \end{cases} \tag{5.11}$$

Нейронная сеть на динамических нейронах может работать в двух режимах: обучения и собственно прогнозирования, при этом, благодаря внутренней памяти КИХ-нейронов, в режиме обучения на вход ИНС достаточно подавать лишь одно значение прогнозируемой последовательности (см. рис. 5.2).

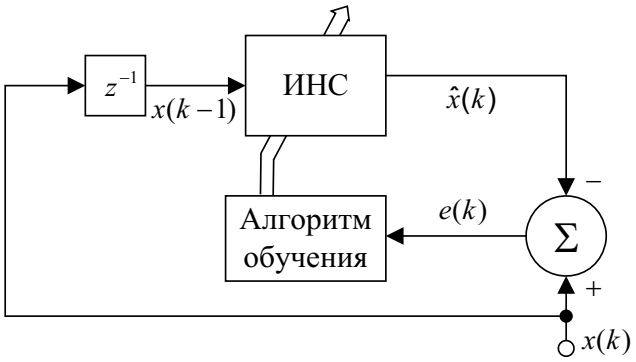


Рис. 5.2 - Динамическая нейронная сеть в режиме обучения

Более глубокая предыстория сигнала формируется в динамических нейронах скрытых слоев. Следовательно, при прогнозировании одномерных временных рядов ИНС имеет только один вход, в то время как использование статических нейронов приводит к тому, что сеть должна иметь, как минимум, n_A входов.

Режим прогнозирования, иллюстрируемый рис. 5.3, реализуется еще проще, при этом выходной сигнал-прогноз сети по обратной связи через элемент задержки z^{-1} подается на ее вход.

Таким образом, прогнозирующая нейронная сеть на динамических нейронах КИХ-фильтрах, имея стандартную архитектуру многослойной ИНС с прямой передачей информации, имеет меньшее количество настраиваемых синаптических весов, а следовательно, более высокую скорость обучения.

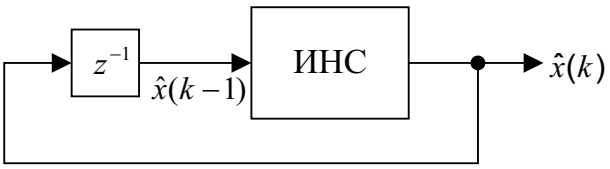


Рис. 5.3 - Динамическая нейронная сеть и режиме прогнозирования

Большинство предложенных нейронных сетей для прогнозирования нестационарных процессов основаны на структуре многослойных нейронных сетей с алгоритмом обучения на основе обратного распространения ошибки, что приводит к снижению скорости обучения. С другой стороны, большинство предложенных нейронных сетей не способны выявлять локальные особенности сигналов, что делает актуальным вопрос синтеза новых гибридных архитектур для решения задачи прогнозирования в последовательном режиме.

5.4. Обнаружение изменений свойств стохастических последовательностей с помощью нейросетевого подхода

Задача обнаружения изменений свойств стохастических последовательностей тесно связана с проблемой диагностики объектов и систем различного назначения и широко изучена с различных позиций [113-115]. Для ее решения предложено множество подходов, связанных большей частью с идеями математической статистики, теории случайных процессов, распознавания образов, кластер-анализа и т.п. При этом следует заметить, что достаточно жесткие предположения о статистических свойствах реальных временных рядов ограничивают возможности традиционных методов.

Рассмотрим искусственную нейронную сеть, предназначенную для обнаружения в реальном времени изменений свойств контролируемого сигнала $x(k)$ и сочетающую в себе достоинства многомодельного подхода и аппроксимирующие свойства прогнозирующих ИНС. Изменение свойств стохастической последовательности фиксируется с помощью диагностирующего вектора, элементы которого являются синаптическими весами выходного нейрона.

Архитектура данной ИНС приведена на рис. 5.4 и представляет собой сеть элементарных нейронов, отличающихся между собой видом активационных функций и алгоритмами обучения, являющимися в общем случае градиентными процедурами безусловной или условной оптимизации.

Контролируемая стохастическая последовательность $x(k)$ подается на входной (нулевой) слой сети, образованный элементами задержки z^{-1}, в результате чего на выходе этого слоя формируется набор задержанных значений временного ряда $x(k-1), x(k-2), \ldots, x(k-n_A)$, при этом чем больше значение n_A, тем более широкими диагностирующими возможностями обладает нейронная сеть.

Первый скрытый слой образован стандартными формальными нейронами с нелинейными активационными функциями φ_j, на входы которых подаются задержанные значения сигнала $x(k)$ и по цепи обратной связи - задержанные значения выходных сигналов (прогнозов) $\hat{x}(k)$, $j = 1,2,\ldots,n_A$.

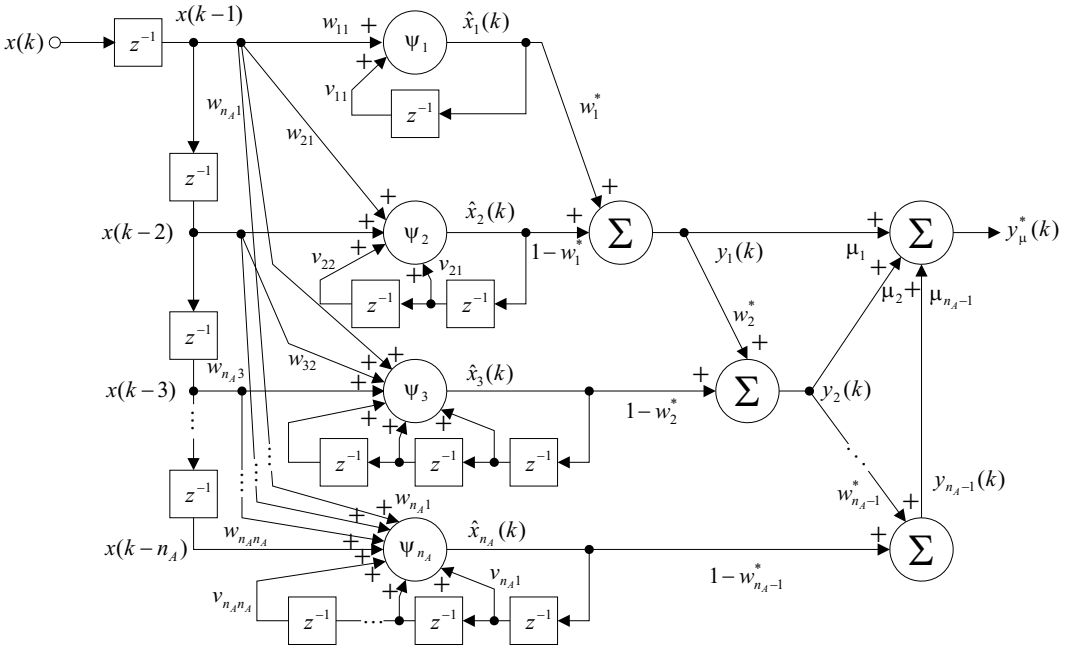


Рис. 5.4 - Диагностирующая нейронная сеть

В результате обработки сигнала нейронами первого слоя на их выходах появляются прогнозные оценки

$$\begin{cases} \hat{x}_1(k) = \varphi_1(x(k-1), \hat{x}_1(k-1)), \\ \hat{x}_2(k) = \varphi_2(x(k-1), x(k-2), \hat{x}_2(k-1), \hat{x}_1(k-1)), \\ .... \\ \hat{x}_{n_A}(k) = \varphi_{n_A}(x(k-1), ..., x(k-n_A), \hat{x}_{n_A}(k-1), ..., \hat{x}_1(k-1)) \end{cases} \tag{5.12}$$

соответствующие нелинейным процессам авторегрессии-скользящего среднего (NARMA - модели) порядка от 1 до n_A. Таким образом, нейроны первого скрытого слоя формируют элементарные "кирпичики", из которых во втором скрытом слое "собираются" прогнозы последовательности $x(k)$. Задачей, решаемой сетью, является определение текущего значения порядка NARMA-процесса и моментов возможного его изменения в реальном времени.

Во втором скрытом слое, образованном $n_A - 1$ адаптивными линейными ассоциаторами, производится попарное объединение прогнозов с целью получения оценок $y_j(k)$, $j = 1,2,\ldots,n_A - 1$

$$\begin{cases} y_1(k) = F(\hat{x}_1(k), \hat{x}_2(k), w_1^*), \quad \hat{x}_1(k) \equiv y_0(k), \\ y_2(k) = F(y_1(k), \hat{x}_3(k), \hat{x}_2(k), w_2^*, w_1^*), \\ ... \\ y_{n_A-1}(k) = F(y_{n_A-2}(k), \hat{x}_{n_A}(k), w_{n_A-1}^*, w_{n_A-2}^*, ..., w_1^*) \end{cases} \tag{5.13}$$

и весовых коэффициентов w_j^*, характеризующих точность объединяемых $y_{j-1}(k), \hat{x}_{j+1}(k)$ и объединенного $y_j(k)$ прогнозов. Следует отметить, что хотя во втором слое формально производится попарное объединение, на содержательном уровне это не совсем так. Если первый нейрон второго скрытого слоя строит объединенный прогноз на основе $\hat{x}_1$ и $\hat{x}_2$, то $y_2(k)$ уже содержит в себе $\hat{x}_1$, $\hat{x}_2$ и $\hat{x}_3$, $y_3 - \hat{x}_1(k)$, $\hat{x}_2$, $\hat{x}_3$ и $\hat{x}_4$ и т.д. Именно во втором скрытом слое формируются оптимальные одношаговые прогнозы, отличающиеся друг от друга объемом используемой предыстории. Вектор текущих весов $w^* = (w_1^*(k), w_2^*(k), ..., w_{n_4-1}^*(k))^T$ описывает качество прогнозирования, достигаемое во втором скрытом слое в каждый текущий момент времени, а изменение соотношений между его элементами уже само по себе свидетельствует об изменении структуры и параметров сигнала $x(k)$. Заметим также, что уже на уровне этого слоя по номеру соответствующего нейрона можно установить, сколько нейронов первого слоя потребуется для удовлетворительной аппроксимации контролируемой последовательности.

В результате обученная нейронная сеть обеспечивает требуемое качество прогнозирования входного сигнала на уровне второго скрытого слоя, при этом в единственном нейроне – адаптивном линейном ассоциаторе выходного слоя формируются оценки "вкладов" каждого из прогнозов $y_j(k)$ в общую модель контролируемого сигнала. Наибольшему "вкладу'" соответствует максимальный вес $\mu_j(k)$, являющийся аналогом вероятности гипотезы о том, что "истинное" состояние $x(k)$ наилучшим образом описывается оценкой $y_j(k)$. Максимальное значение $\mu_j(k)$ определяет порядок наилучшей NARMA-модели в момент времени *к,* а непрерывное уточнение вектора $\mu(k)$ с помощью соответствующего алгоритма обучения позволяет обнаруживать момент изменения свойств.

После возникшей разладки, когда свойства контролируемого сигнала претерпели изменения, "вклады" отдельных $y_j(k)$ соответственно изменяются и оптимальный прогноз обеспечивается уже иной комбинацией нейронов второго слоя. При этом, естественно, изменяются соответствующие $\mu_j(k)$, что и фиксируется на уровне выходного нейрона.

Таким образом, рассмотренная диагностирующая ИНС обеспечивает наряду с традиционным в теории и практике нейросетей прогнозированием и раннее обнаружение разладок в реальном времени. Также можно выделить и недостаток диагностирующей нейронной сети, который связан с тем, что если временной ряд зашумлен различного видами выбросов негаусовского распределения, такая сеть не обеспечивает требуемой точности анализа.

Таким образом, актуальным является разработка гибридных вэйвлет-нейро-фаззи-систем с подсистемой обнаружения разладок, которые бы по-

зволили обрабатывать нелинейные процессы и выявлять разладки при последовательном поступлении наблюдений.

5.5. Интеллектуальное управление на основе нейро- и фаззи- систем

Абсолютное большинство адаптивных параметрически оптимизируемых регуляторов основывается на гипотезе о линейности объекта управления, что резко сужает их возможности при управлении реальными технологическими процессами, имеющими, как правило, существенно нелинейные характеристики.

В связи с этим в последние годы пристальное внимание специалистов привлечено к возможностям использования аппарата теории искусственных нейронных сетей в задачах управления техническими объектами и технологическими процессами [118, 119]. По сравнению с адаптивными системами управления уровень неопределенности при использовании нейрорегуляторов может быть значительно выше, а главное – использование искусственных нейронных сетей не требует аналитического описания объекта управления, кроме предположения, что имеется некоторая достаточно произвольная функциональная связь между входом и выходом

$$y(k+1) = f(y(k), y(k-1), ..., y(k-n_A+1), u(k), ..., u(k-n_B+1)). \quad (5.14)$$

Функциональная схема адаптивного нейрорегулятора приведена на рис. 5.5, где ЛЗ обозначают линии элементов запаздывания, на которых формируются задержанные значения входного, выходного и внешнего задающего сигналов объекта управления.

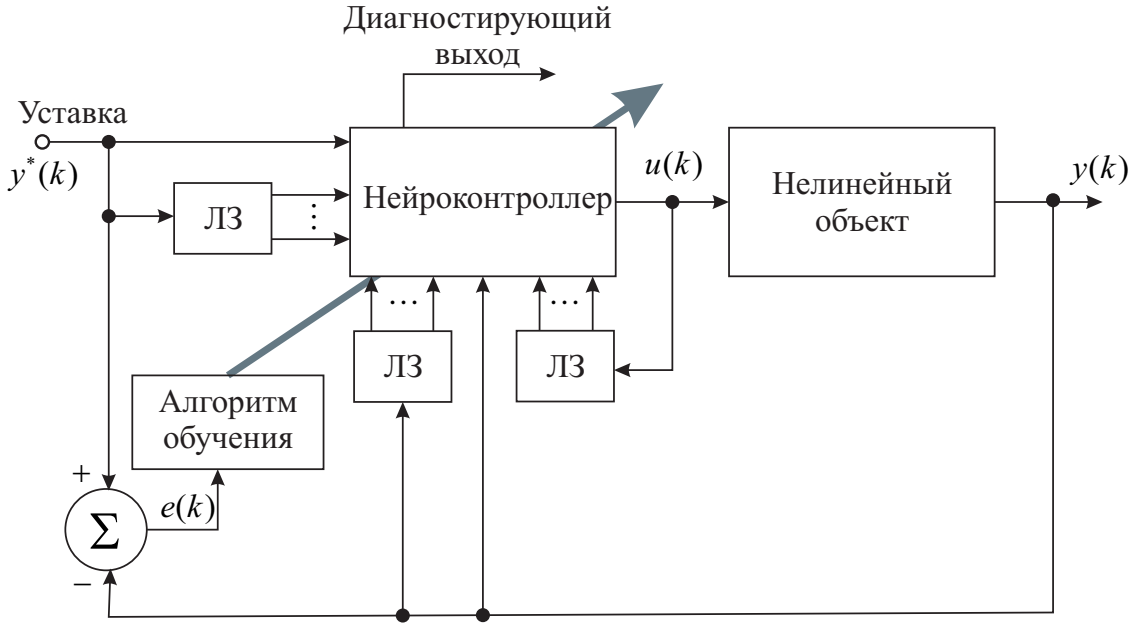


Рис. 5.5 - Адаптивный нейрорегулятор

Нейрорегулятор реализуется на основе трехслойного персептрона, представленного на рис. 5.6.

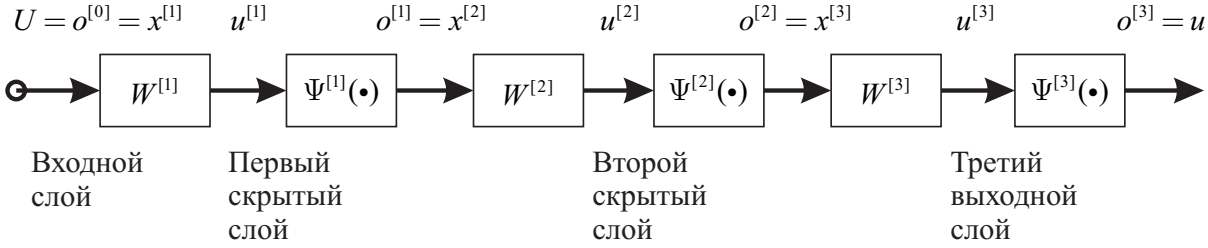


Рис. 5.6 - Персептрон в качестве регулятора

Здесь $x^{[s]} = (x_1^{[s]}, x_2^{[s]}, ..., x_{n_s-1}^{[s]})^T - (n_{s-1} \times 1)$ вектор входных сигналов s-го слоя;

$o^{[s]} = (o_1^{[s]}, o_2^{[s]}, ..., o_{n_s}^{[s]})^T - (n_s \times 1)$ вектор выходных сигналов s-го слоя;

$u^{[s]} = (u_1^{[s]}, u_2^{[s]}, ..., u_{n_s}^{[s]})^T - (n_s \times 1)$ вектор сигналов на выходе сумматоров s-го слоя;

n_s – количество нейронов в s-м слое ($s = 1,2,3$);

$W^{[s]} = \{w_{ji}^{[s]}\} - (n_s \times n_{s-1})$ – матрица настраиваемых синаптических весов s-го слоя;

$\Psi^{[s]}(\bullet) - (n_s \times n_s)$ – диагональный нелинейный оператор, образованный функциями активации нейронов s-го слоя. При этом очевидно, что выходные сигналы $o_i^{[s-1]}$ $(s-1)$-го слоя являются входными $x_i^{[s]}$ для s-го слоя так, что $x_i^{[s]} = o_i^{[s-1]}$, $x^{[1]}(k) = o^{[0]}(k) = U(k), o^{[3]}(k) = u(k)$; число нейронов первого слоя равно $n_0 = n_1 = n_G$, а выходной слой образован одним нейроном ($n_3 = 1$), на выходе которого появляется управляющий сигнал $u(k)$. Количество нейронов во втором скрытом слое принимается $1 \le n_2 \le n_G$.

Для j-го нейрона s-го слоя справедливо соотношение

$$o_j^{[s]} = \varphi_j^{[s]}(u_j^{[s]}) = \varphi_j^{[s]}\left(\sum_{i=1}^{n_{s-1}} w_{ji}^{[s]} o_i^{[s-1]}\right), \tag{5.15}$$

где $\varphi_j^{[s]}(\bullet)$ - функция активации j-го нейрона, для s-го слоя

$$o^{[s]} = \Psi^{[s]}(W^{[s]} x^{[s]}) = \Psi^{[s]}(W^{[s]} o^{[s-1]}) \tag{5.16}$$

и, наконец, для нейронной сети в целом –

$$u = \Psi(U) = \Psi^{[3]}(W^{[3]}\Psi^{[2]}(W^{[2]}\Psi^{[1]}(W^{[1]}(U)))). \tag{5.17}$$

В качестве критерия управления примем локальную функцию

$$E_k = \frac{1}{2}(y^*(k) - y(k))^2 = \frac{1}{2}e^2(k), \tag{5.18}$$

а для обучения нейронной сети будем использовать процедуру обратного распространения ошибок [118], при этом синаптические веса выходного нейрона уточняются согласно рекуррентной процедуре вида

$$W^{[3]}(k+1) = W^{[3]}(k) + \frac{\delta^{[3]}(o^{[2]}(k))^T}{\left\|(\varphi^{[3]})' o^{[2]}(k)\right\|^2} \tag{5.19}$$

$$W_j^{[2]}(k+1) = W_j^{[2]}(k) + \frac{\delta_j^{[2]}(o^{[1]}(k))^T}{\left\|(\varphi_j^{[2]})' o^{[1]}(k)\right\|^2}, \quad j = 1,2,...,n_2, \tag{5.20}$$

$$W_j^{[1]}(k+1) = W_j^{[1]}(k) + \frac{\delta_j^{[1]} U^T(k)}{\left\|(\varphi_j^{[1]})' U(k)\right\|^2}, \quad j = 1,2,...,n_G. \tag{5.21}$$

Таким образом, с помощью алгоритмов (5.19), (5.20), (5.21) обеспечивается оптимальная по быстродействию настройка всех нейронов регулятора пониженного порядка.

Объединение преимуществ регуляторов пониженного порядка с широкими возможностями искусственных нейронных сетей позволяет эффективно решать задачи управления нелинейными нестационарными объектами н условиях существенной априорной неопределенности.

С другой стороны, фаззи-системы управления в последние годы находят все более широкое применение (например, в ферросплавном производстве, [27, 30, 142, 143]). Одним из преимуществ нечетких систем является их способность работать с лингвистическими переменными. Это особенно важно, поскольку многие объекты управления могут характеризоваться лингвистическими описаниями. Наибольшее распространение получили схемы нечетких регуляторов, предложенные Мамдани [121] и Сугено [120]. В то время как первая из них отличается интуитивностью процесса синтеза регулятора и лучше приспособлена для представления знаний эксперта в лингвистической форме, вторая является более эффективной с точки зрения вычислительных затрат и позволяет использовать методы оптимизации и адаптации.

Рассмотрим метод синтеза нечеткого регулятора на основе схемы Сугено.

Пусть объект управления описывается следующим уравнением:

$$\begin{aligned} &y(k) = F(y(k-1),...,y(k-n),u(k-1),...,u(k-m)) + \xi(k), \\ &M\{\xi(k)\} = 0, \quad M\{\xi^2(k) = P_w < \infty\}, \end{aligned} \tag{5.22}$$

где $y(k)$ - выход объекта управления в дискретный момент времени k;

$u(k)$ - сигнал управления;

$\xi(k)$ - случайное возмущение типа белого шума;

F - некоторая функциональная зависимость, неизвестная в общем случае;

$M\{\bullet\}$ - символ математического ожидания.

Задача управления заключается в нахождении в каждый момент времени k такого значения $u(k)$, что

$$y(k+1) = y^*(k+1) + w(k+1), \tag{5.23}$$

где задающий сигнал $y^*(k)$ вычисляется с использованием значения уставки $r(k)$ (или желаемого установившегося значения выхода объекта управления в момент времени k), и дискретной передаточной функции $W_R(z)$ формирующего фильтра задающего сигнала

$$y^*(z) = W_R(z)r(z), \quad W_R(z) = \frac{p_1 z^{-1} + \ldots + p_k z^{-k}}{1 + q_1 z^{-1} + \ldots + q_l z^{-l}} \tag{5.24}$$

(здесь z^{-1} обозначает элемент запаздывания на один такт).

Разделим область изменения $y(k)$ и $u(k)$ на N и M интервалов соответственно. Тогда пространство состояний модели (5.22) будет разбито на N^n и M^m областей. Предположим, что в i-й области модель (5.22) может быть аппроксимирована линейным уравнением с постоянными параметрами:

$$\begin{aligned} &y(k) = a_{1,i}y(k-1) + \ldots + a_{n,i}y(t-n) + b_{1,i}u(t-1) + \ldots \\ &+b_{m,i}u(t-m) + \varepsilon(k) + C_i, \quad i = 1,\ldots,N^n \times M^m, \end{aligned} \tag{5.25}$$

где C_i – константа-смещение.

Используем для управления линеаризованной системой (5.25) нечеткий регулятор Сугено [120]. Правила, реализуемые таким регулятором, имеют вид

$$\begin{aligned} &IF\ x_1\ is\ A_1\ AND\ x_2\ is\ A_2\ AND\ \ldots\ AND\ x_n\ is\ A_n \\ &THEN\ u = f(x_1, x_2, \ldots, x_n), \end{aligned} \tag{5.26}$$

где $x_1, \ldots, x_n$ входные переменные.

В большинстве приложений f является линейной комбинацией входных переменных (система Сугено 1-го порядка), или просто константой (система Сугено 0-го порядка).

Учитывая (5.23), получим модификацию одношагового алгоритма управления для системы (5.25):

$$\begin{aligned} u(k) = \frac{1}{b_{1,j}}[&y^*(k+1) - a_{1,i}y(k) - \ldots - a_{n,i}y(k-n+1) \\ &-b_{2,i}u(k-1) - \ldots - b_{m,i}u(k-m+1) - C_i]. \end{aligned} \tag{5.27}$$

Выбирая соответствующие параметры передаточной функции $W_R(z)$ в уравнении (5.24). можно добиться необходимого качества управления.

Запишем уравнение (5.27) в виде правила (5.20):

$$
\begin{aligned}
&IF\ y^*(k+1)\ is\ Y_{1,i}\ AND\ t(k)\ is\ Y_{2,i}\ AND\ldots\\
&AND\ \ y(k-n+1)\ is\ Y_{n+1,i}\ AND\ \ u(k-1)\ is\ U_{1,i}\ AND\ldots\\
&AND\ u(k-m+1)\ is\ U_{m-1,i}\\
&THEN\ u_i=\alpha_{1,i}y^*(k+1)+\alpha_{2,i}y^*(k)+\alpha_{3,i}y^*(k-1)+\ \ldots+\\
&\alpha_{n,i}y^*(t-n)+\beta_{1,i}u(k-1)+\ldots+\beta_{m,i}u(k-m+1)+\gamma_i
\end{aligned}
\tag{5.28}
$$

где $a_{1,i}$=1 / $b_{1,i}$, $a_{j,i}=-a_{j-1,i}/b_{1,i}$, $j=\ 2,\ldots,n+1$,

$b_{l,i}=-b_{l+1,i}/b_{l,i}$, $k=1,\ldots,m-1$, $\gamma_i=-C_i$,

i – номер правила;

$u_i(k)$ - выход i-го правила и $b_{1,i}\neq 0$.

Выход регулятора вычисляется с использованием процедуры дефаззификации:

$$
u(k)=\frac{\sum w_i u_i(k)}{\sum w_i},\quad i=1,...,N_R, N_R=N^{n+1}\times M^{m-1},
\tag{5.29}
$$

где w_i - степень выполнения i-го правила; N_R - количество правил.

Структура контура управления показана на рис. 5.7.

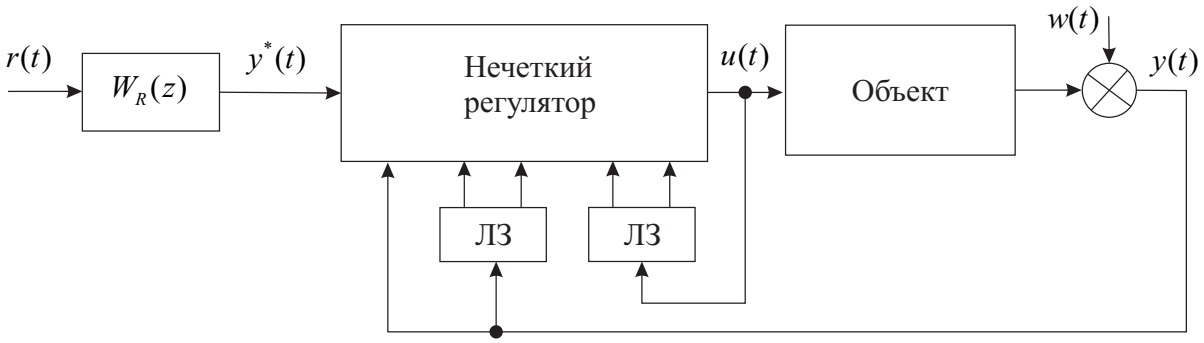


Рис. 5.7 - Структура контура управления

Запишем закон управления, реализуемый нечетким регулятором (5.28), в общем виде:

$$
\begin{aligned}
&u(k)=F_u(X(k)),\\
&X(k)=(y^*(k+1),y(k),...,y(k-n+1),u(k-1),...,u(k-m+1)),
\end{aligned}
\tag{5.30}
$$

где F_u - некоторая неизвестная функциональная зависимость;

$X(k)$ - вектор входных переменных нечеткого регулятора в момент времени k.

Для идентификации зависимости F_u используем схему адаптивной нейро-фаззи-системы Для настройки с помощью адаптивной нейро-фаззи-системы необходима обучающая выборка данных, которая имеет вид в рамках принятых нами обозначений

$$D = \{X'(k-1), u(k-1) \mid k = 1,..., N\}, \tag{5.31}$$

где

$$\begin{aligned} u(k-1) &= F_u(X(k-1)), \\ X(k-1) &= (y(k), y(k-1),..., y(k-n), u(k-1),..., u(k-m)). \end{aligned} \tag{5.32}$$

Адаптивное (самонастраивающееся) нечеткое управление является популярным методом, который успешно применяется в управлении процессами [129]. В дальнейшем адаптивное нечеткое управление значительно расширилось с использованием нейронных сетей. Разнообразие нейронечетких регуляторов усовершенствовало нечеткое управление. Однако вычисление правил и обучение нейронной сети обычно занимает продолжительное время. Применение новых гибридных нейро-фаззи-систем и их алгоритмов обучения дает возможность повысить скорость обучения и качество управления.

Проведя анализ, можно сказать, что задача интеллектуального управления на основе гибридных нейро-фаззи-систем интенсивно развивается и по сей день, что подтверждается множеством публикаций. Но те или иные подходы к сожалению обладают рядом недостатков, которые не позволяют реализовать управление в последовательном режиме (в текущем дискетном времени).

Таким образом, на данный момент актуальной проблемой является разработка интеллектуальных регуляторов, структура которых должна быть проста в технической реализации, а с другой стороны давать качественные управляющие воздействия.

5.6. Модели нестационарных сигналов, временных рядов и предобработка массивов входных данных

За последнее время, важность интеллектуального анализа нестационарных временных рядов в области технических, экономических, биомедицинских и других исследований постоянно возрастает, о чем свидетельствует множество публикаций в ведущих журналах, как Украины, так и за рубежом. В обрабатывающей промышленности и промышленном производстве, особый интерес представляет прогнозирование временных рядов, когда на основании некоторой предыстории, прогнозируются будущие поведение систем. Это очень важно в процессе производства и контроля, а также в оптимальном управлении объектами различной природы и назначения, особенно с нелинейной динамикой

В инженерной практике, значения временного ряда получают от разного вида датчиков, из чего следует, что полученные значения подвержены зашумлению, и, таким образом, временной ряд будет состоять из детерминированной составляющей сигнала и стохастической компоненты.

Анализ временных рядов, прежде всего, направлен на изучение внутренней структуры ряда (автокорреляция, тренд, сезонность и т.д.) для того,

чтобы получить скрытые зависимости процесса, при котором данные временного ряда формируются [122, 123].

Понятие анализ временных рядов включает в себя ряд задач:

– определение, классификация и описание временных рядов;

– построение модели временного ряда;

– прогнозирование будущих моментов временного ряда по заданной предыстории;

– сегментирование и кластеризация;

– обнаружение разладок в реальном времени и другие [20].

Если время измеряется непрерывно, временной ряд называется непрерывным, если же время фиксируется дискретно (т.е. через фиксированный интервал времени), то временной ряд дискретен. В дальнейшем будем рассматривать только дискретные временные ряды. Они получаются двумя способами:

– выборкой из непрерывных временных рядов через регулярные промежутки времени - такие временные ряды называются моментными;

– накоплением переменной в течение некоторого периода времени - в этом случае временные ряды называются интервальными.

В зависимости от характера данных временные ряды могут быть: стационарными, нестационарными, хаотическими, сезонными и несезонными, линейными и нелинейными, одномерными и многомерными, периодическими, апериодическими и квазипериодическими.

На практике реальные временные ряды могут сочетать в себе несколько из выше перечисленных характеристик. Например, линейные ряды могут быть стационарными, сезонными и т.д. Рассмотрим вкратце каждый подвид временных рядов.

Линейные временные ряды формируются путем наблюдения линейных процессов, которые математически определяются линейной моделью вида [4,5]

$$y(t) = \sum_{j=-\infty}^{\infty} \alpha_j x(t-j), \tag{5.33}$$

где коэффициенты α_j подчиняются ограничениям вида

$$\sum_{j=-\infty}^{\infty} \left|\alpha_j\right| < \infty. \tag{5.34}$$

С другой стороны, многие временные ряды в различных областях науки и техники требуют нелинейного моделирования [122,124]. Такие процессы могут быть представлены в ряде случаев в виде билинейного временного ряда вида

$$x_t = z_t + \sum_{i=1}^{p} a_i x_{t-i} + \sum_{j=1}^{q} b_j z_{t-j} + \sum_{i=1}^{r}\sum_{j=1}^{s} c_{ij} z_{t-i}. \tag{5.35}$$

Одномерные временные ряды представляют собой выборку наблюдений одновременного процесса, например, значения одной физической переменной или одного нестационарного сигнала с одинаковым промежутком квантования. Таким образом, в одномерном временном ряду параметр времени является неявной переменной, которая обычно заменяется на индекс-переменную.

В случае одномерного временного ряда, его математическая модель может быть всегда построена и носит название детерминированной. В противном случае, если временной ряд может быть только представлен в терминах функции распределения вероятностей, то такой временной ряд называется недетерминированым или стохастическим [124].

С другой стороны, широко распространены многомерные временные ряды, которые генерируются одновременно при наблюдении двух или более процессов. Такие наблюдения распространены в металлургии и машиностроении, где две или более физических величины (температура, давление, расход и т.д.) должны быть одновременно обработаны для построения модели динамической системы.

В многомерных временных рядах анализу подаются не только скрытые зависимости внутри одного процесса, а и зависимости между отдельными процессами, которые объединены в многомерный временной ряд. Для обработки такого рода временных рядов может быть использован многофакторный анализ, являющийся разделом математической статистики.

При анализе временных рядов традиционно различают разные виды динамики. Эти виды динамики могут, вообще говоря, комбинироваться. Тем самым задается разложение временного ряда на составляющие (компоненты).

Перечислим наиболее важные:

– тенденция соответствует медленному изменению, проходящему в некотором определенном направлении, которое сохраняется в течение значительного промежутка времени. Тенденцию называют также *трендом* или долговременным движением;

– циклические колебания – это более быстрая, чем тенденция, квазипериодическая динамика, в которой есть фаза возрастания и фаза убывания. В экономике, например, наиболее часто цикл связан с флуктуациями экономической активности;

– сезонные колебания соответствуют изменениям, которые происходят регулярно в течение года, недели или суток. Они связаны с сезонами и ритмами человеческой активности, природы, производственной необходимостью;

– календарные эффекты это отклонения, связанные с определенными предсказуемыми календарными событиями, такими как праздничные дни, количество рабочих дней за месяц, високосность года и т.п.;

– случайные флуктуации беспорядочные движения относительно большой частоты. Они порождаются под влиянием разнородных событий на изучаемую величину (несистематический или случайный эффект);

– выбросы – это аномальные движения временного ряда, связанные с редко происходящими событиями, которые резко, но лишь очень кратковременно отклоняют ряд от общего закона, по которому он движется;

– структурные сдвиги это аномальные движения временного ряда, связанные с редко происходящими событиями, имеющие скачкообразный характер и меняющие тенденцию.

Случайные составляющие временных рядов в основном делятся на две категории:

– действительно случайные, т.е. наблюдения, характеризующиеся функцией распределения плотности вероятностей или статистическими моментами, такими, как средние, дисперсия, фазовый сдвиг и т.д.

– хаотические, характеризующиеся значениями с некоторым, как правило, гиперболическим распределением, но на самом деле наблюдаемый процесс является полностью детерминированным.

Последнее подтверждается тем, что во многих областях науки и практики, связанных с физикой, биологией, химией, экономикой, медициной и др., достаточно часто присутствует широкий класс детерминированных нелинейных систем, хаотическое поведение которых выглядит как случайное, хотя по сути им не является. Более того, статистический анализ сигналов, генерируемых такими системами (вторые моменты, автокорреляционные функции, спектр) показывает, что это широкополосный случайный процесс, порождаемый детерминированным объектом, что само по себе парадоксально.

Такие системы называются хаотическими, и последние два десятилетия они являются предметом пристального внимания как теоретиков, так и специалистов совершенно различных областей [6,7,11,17].

Хаотический процесс, порождаемый нелинейной детерминированной системой, хотя внешне очень похож на стохастический, все же таковым не является. Основной его особенностью является крайняя чувствительность к начальным условиям, т.е. если одна и та же система стартует из начальных условий $x(0)$ и $x(0)+\varepsilon$, где ε – очень малая величина, то ее траектории движения экспоненциально расходятся во времени, стремясь к совершенно различным областям притяжения, называемыми странными аттракторами. Используя более строгое определение, можно сказать, что странный аттрактор – это притягивающее множество в фазовом пространстве, в котором движутся хаотические траектории, не являющееся при этом ни положением равновесия, ни предельным циклом.

В принципе, будущее поведение хаотической системы в силу её детерминированности полностью определяется ее прошлым, но на практике любая неопределенность или неточность в выборе начальных условий резко усложняет задачу анализа, для решения которой в последнее время все чаще используются искусственные нейронные сети и системы нечеткого вывода, благодаря своим универсальным аппроксимирующим свойствам и способности обучаться в процессе обработки информации [28, 126,132,134,143].

Кроме собственно хаотического движения, с нелинейными динамическими системами связан ряд типов поведения близких к хаосу и прежде всего:

– переходный хаос, представляющий собой движение, которое на конечном временном интервале выглядит как чисто хаотическое, т.е. траектория сначала развивается по странному аттрактору, но затем выходит на периодическое или квазипериодическое движение;

– квазипериодические колебания, представляющие собой колебания с двумя или более некратными частотами;

– бифуркации, представляющие собой резкие изменения характера движения (изменение траектории) при малом изменении одного или нескольких параметров системы;

– смеси типа "хаос + квазипериодические колебания" и т.п.

Общим для всех отмеченных типов поведения является их фрактальная структура [1, 2, 9, 22], т.е. самоподобие анализируемых процессов при различных пространственных и временных масштабах. В принципе любая существенно нелинейная динамическая система при определенном сочетании ее параметров может демонстрировать хаотическое поведение, однако на практике было исследовано и используется достаточно ограниченное число таких структур, среди которых можно выделить наиболее популярные:

– логистическое уравнение, описывающее рост биологической популяции,

$$x(k+1) = wx(k)(1-x(k)), \quad 0 \le w \le 4, \quad 0 \le x(0) \le 1, \tag{5.36}$$

где $k = 0,1,2,\ldots$ текущее дискретное время;

– модификации (5.36) типа

$$\begin{aligned} x(k+1) &= w_1 x(k) + w_2 x^2(k), \\ x(k+1) &= x^2(k) + \theta, \\ x(k+1) &= 1 - w_1 x^2(k) + w_2 x(k-1), \\ x(k+1) &= w x^3(k) + (1-w)x(k), \end{aligned} \tag{5.37}$$

где w, w_1, w_2, θ - некоторые скалярные параметры;

– модель Мэки- Гласса

$$x(k+1) = w_1 x(k) + \frac{w_2 x(k-\tau)}{1 + x^{10}(k-\tau)}, \tau \ge 17, \tag{5.38}$$

где τ – временная задержка;

– уравнения Мандельброта (Mandelbrot)

$$\begin{cases} x(k+1) = x^2(k) - y^2(k) + \theta_x, \\ y(k+1) = 2x(k)y(k) + \theta_y \end{cases} \tag{5.39}$$

и множество других.

Как уже отмечалось, хаос внешне очень напоминает случайный процесс, хотя им и не является, в связи с чем возникла задача моделирования и идентификации сигналов, состоящая в определении по имеющемуся набору данных их природы: случайной или детерминированной хаотической [6, 7].

Одним из признаков хаотической системы является появление на ее выходе широкого спектра частот при подаче на вход гармонического или постоянного сигнала. И хотя этот спектр внешне напоминает спектр белого шума, автокорреляционная функция хаотического процесса в отличие от дельта-функции белого шума имеет протяженный характер.

Не менее важным вопросом остается и предобработка данных для проведения дальнейшего анализа, что позволяет привести в соответствие с требованиями, определяемыми спецификой решаемой задачи.

Предобработка данных включает два направления: очистку и оптимизацию. Очистка производится с целью исключения факторов, снижающих качество данных и мешающих работе алгоритмов. Она включает обработку дубликатов, противоречий и фиктивных значений, восстановление и заполнение пропусков, сглаживание и очистку данных от шума, подавление и редактирование аномальных значений. Кроме этого, в процессе очистки восстанавливаются нарушения структуры, полноты и целостности данных, преобразуются некорректные форматы.

Оптимизация данных, как элемент предобработки, включает снижение размерности входных данных, выявление и исключение незначащих признаков. Основное отличие оптимизации от очистки в том, что факторы, устраняемые в процессе очистки, существенно снижают точность решения задачи или делают работу аналитических алгоритмов невозможной. Проблемы, решаемые в процессе оптимизации, адаптируют данные к конкретной задаче и повышают эффективность их анализа.

Нормализация данных – финальный процесс предобработки данных для дальнейшего использования их в обучении нейронных сетей.

Широко распространенным методом нормирования данных является их логарифмирование с последующим формированием дополнительных временных рядов из первых или вторых разностей. Первые разности представляют собой приближенный дискретный аналог первой производной, а вторые разности – второй производной. Часто из значений ряда вычитают среднее, для того чтобы получить возможность работать с отклонениями, а не полными значениями переменных.

Для анализа данных на основе нейро-фаззи-технологий, в зависимости от используемых функций активации или принадлежности, необходима нормировка данных, которую можно провести на основе следующих выражений:

$$1.\ x_i^{new} = \frac{x_i}{x_{i,\max}};$$

$$2.\ x_i^{new} = \frac{x_i}{x_{i,\max} - x_{i,\min}};$$

$$3.\ x_i^{new} = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}}, x_i^{new} \in [0,1];$$

$$4.\ x_i^{new} = \frac{x_i - x_{i,mean}}{x_{i,\max} - x_{i,\min}}, x_i^{new} \in [-1,1];$$

$$5.\ x_i^{new} = \frac{x_i - x_{i,mean}}{\sigma_{x_i}};$$

$$6.\ x_i^{new} = \frac{x_i}{\|x\|^2} - \textit{нормировка на гипершар}.$$

Еще одна важная проблема – это формирование обучающей и тестовой выборки для нейро-фаззи-систем. На сегодняшнее время окончательно не выработаны терминальные алгоритмы разбиения выборки данных на два подмножества. В большинстве случаев это разбиение выборки в процентном отношении 70% на 30%. С. Хайкин в своей работе [125] предложил, что количество точек N в обучающей последовательности должно быть $N = N_w / e$ точек, где e – желаемая ошибка прогноза, классификации, N_w – количество весовых коэффициентов в нейро-фаззи-сети.

Тем не менее, независимо от выбранного отношения между выборками, внимание должно быть уделено обеспечению того, чтобы обучающая выборка была репрезентативной и количество точек достаточно, чтобы обучить систему. Во всех случаях, а особенно таких как МГУА-нейронные сети*, кроме обучающей (training set) и тестовой (testing, checking set) для проверки процесса обучения на переобучение необходима и проверочная выборка (validation set).

Таким образом, применение того или другого метода нормирования, структурирования и генерации обучающей, тестовой и проверочной выборок данных определяется в каждом случае по-своему для соответствующей решающей задачи.

5.7. Нейро-фаззи-системы в задачах динамического интеллектуального анализа данных

Отличительной особенностью нейронных сетей является их способность к обучению, которая дает возможность создания интеллектуальных систем распознавания образов, классификации, диагностики, прогнозиро-

* МГУА – метод группового учёта аргумента [69].

вания и т.д. Фаззи-системы способны интерпретировать неточные данные, которые могут быть полезными в принятии решений. Таким образом, актуальной является проблема синтеза гибридных систем вычислительного интеллекта, которые вобрали бы в себя преимущества каждого подхода и были бы способны решать разнообразные сложные проблемы в условиях неопределенности.

Постоянно растущий интерес к объединению интеллектуальных технологий, в частности, объединение нейро- и фаззи- технологий, привел к созданию нейро-фаззи- или фаззи-нейронных структур, что, в свою очередь, расширило возможности обеих технологий.

На данный момент существует большое количество различных архитектур гибридных систем, например, нейроны на основе фаззи-логики (fuzzy logic based neurons), фаззи-нейроиы, нейронные сети с нечеткими весами, нейро-фаззи-адаптивная модель и др. Предложенные архитектуры добились успеха в решении различных инженерных проблем таких, как идентификация, моделирование нестационарных процессов, диагностика, когнитивное моделирование, классификация, распознавание образов, обработка изображений, проектирование, обработка сигналов, прогнозирование и т.д.

Существует несколько методов для реализации технологии нейро-фаззи моделирования. Одной из первых предложенных технологий объединения была замена сигналов вход/выход или весовых коэффициентов нейронных сечей па функции принадлежности фаззи-множеств, получение так называемых фаззи-нейронов. Ряд исследователей в своих работах предложили внутреннюю структуру фаззи-нейронов [28, 30].

В общем случае, гибридизацию нейро- и фаззи- подходов можно провести двумя способами

– синтез нейронных сетей, которые бы обладали возможностью обработки нечеткой информации, так называемых, фаззи-нейронных сетей (fuzzy- neural network);

– синтез фаззи-систем, дополненных способностями обучения и ассоциации нейронных сетей для улучшения их характеристик таких, как гибкость, скорость и приспособляемость: так называемых, нейро-фаззи-систем (neural-fuzzy systems).

Нейронные сети с фаззи-нейронам, их ещё называют фаззи-иейронными сетями, потому что они также способны обрабатывать нечеткую информацию. Нейро-фаззи-системы, с другой стороны, предназначены для реализации процесса нечеткого вывода, в которых связь весов сети соответствует параметрам фаззи-вывода. Первая модель (рис. 5.8) состоит из блока нечеткого вывода, и следующим за ним нейросетевым блоком, состоящим из многослойной нейронной сети с прямой передачей данных, на вход которой подается информация после фаззи-вывода. Используемая нейронная сеть может быть адаптирована и обучена на основе обучающей выборки в процессе обработки данных с помощью различных алгоритмов обучения. Во второй модели (рис. 5.9) блок нейронных сетей управляет

системой фаззи-вывода для получения соответствующего решения. Таким образом, первая модель получает лингвистические входы и генерирует числовые выходы, а вторая модель имеет числовые входы и генерирует лингвистические выходы [139,142].

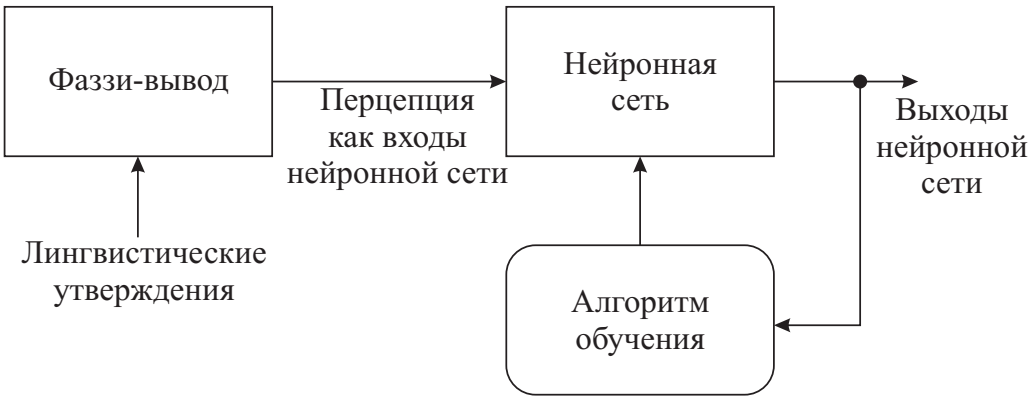


Рис. 5.8 - Фаззи-нейронная система (первая модель)

Кроме того, второй подход заключается в использовании фаззи-функций принадлежности для предварительной обработки или постобработки сигналов с помощью нейронных сетей, как показано на рис.5.10. Фаззи-система может легко закодировать знания эксперта, используя правила с лингвистическими метками.

На практике, для оптимального выбора функций принадлежности требуются ряд навыков. Комбинируя ту же самую систему с нейронной сетью, можно использовать способность к обучению, и выполнить настройки функций принадлежности и постепенно повысить производительность всей гибридной системы.

Авторы работы [127,128] предложили модель нейронной сети на основе фаззи-управления, состоящей из нейронной сети с прямой передачей информации, узлы которой в скрытых слоях системы реализуют функции принадлежности и фаззи-правила, составляющие систему фаззи-вывода с распределенным представлением и алгоритмом обучения нейронной сети. В данном случае параметры функций принадлежности определяются с помощью любого подходящего алгоритма обучения. Разработана модель, которая используется для мульти-спектрального анализа изображений. Наконец, Б. Коско [84] в своих работах предложил ряд усовершенствованных нейро-фаззи моделей с фаззи-ассоциативной памятью, которые дали возможность повысить качество обработки данных.

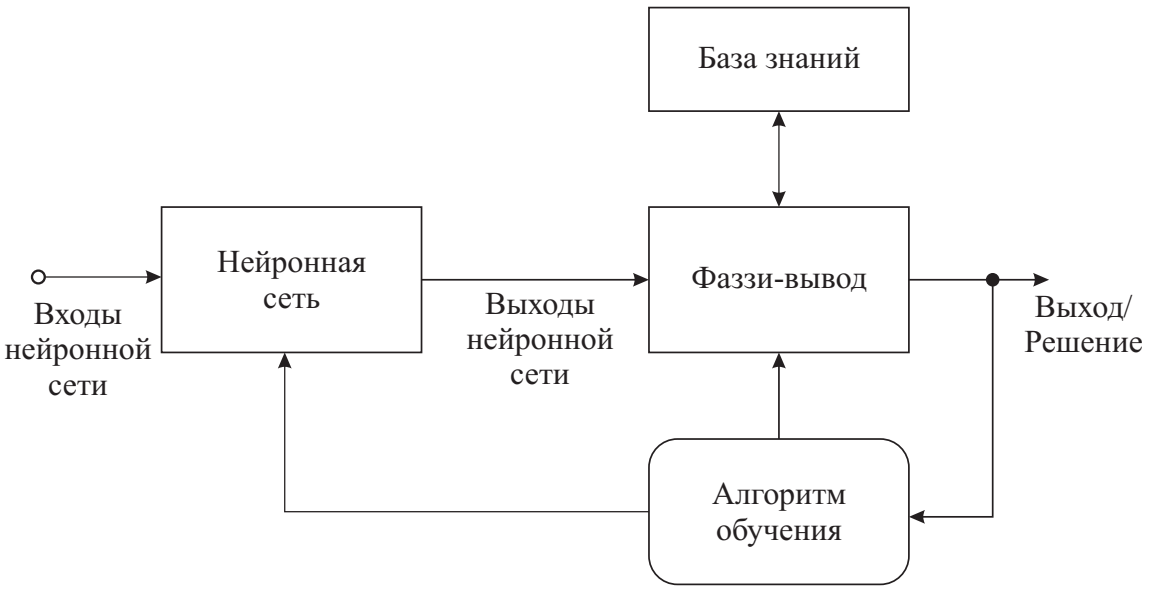


Рис. 5.9 - Фаззи-нейронная система (вторая модель)

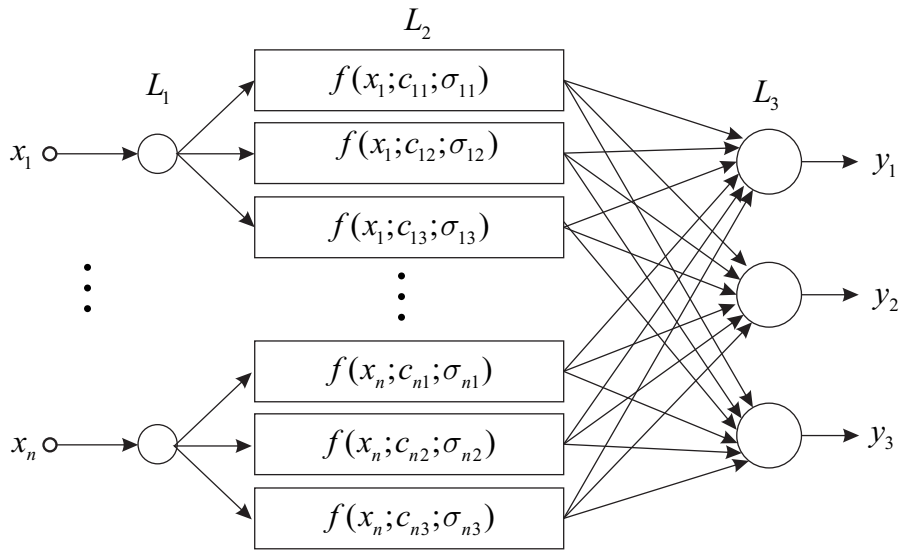


Рис. 5.10 - Фаззи-нейронная модель с настраиваемыми функциями принадлежности

Нейро-фаззи-модель ANFIS (адаптивная нейро-фаззи-система) имеет структуру, представленную на рис. 5.11. Такая система включает в себя пять слоев, осуществляя фаззи-вывод Такаги-Сугено. Предложенная модель имеет относительно сложную архитектуру для большого количества входов, что позволяет обрабатывать большое количество нечетких правил. Для обучения линейных параметров правил используется метод наименьших квадратов, а для оптимальной настройки функций принадлежности используется метод обратного распространения ошибки.

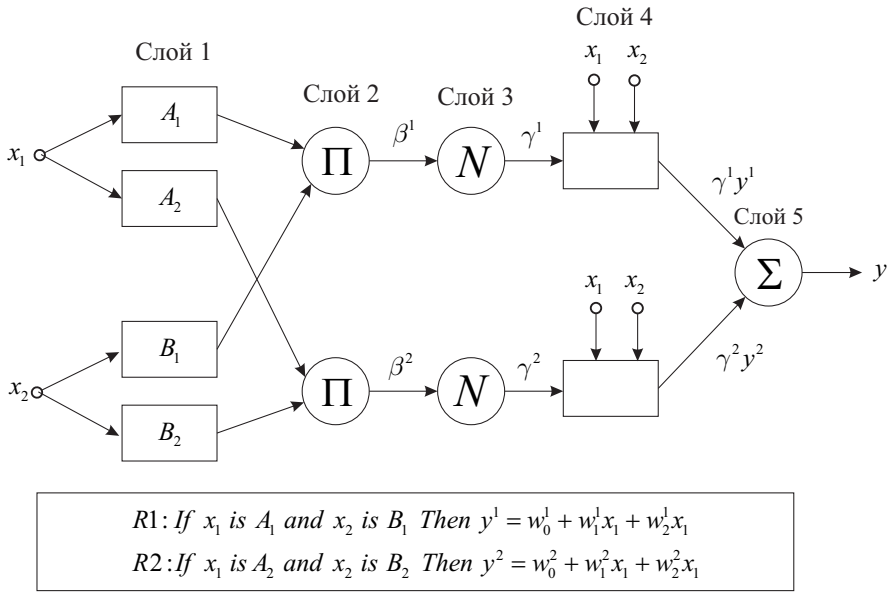


Рис. 5.11 - ANFIS архитектура на основе системы фаззи-вывода Такаги-Сугено с двумя правилами

Среди рассмотренных нейро-фаззи-архитектур можно выделить ряд недостатков: во- первых, это медленно сходящиеся алгоритмы обучения, во-вторых, это эмпирический подбор параметров и выбор вида функций принадлежности, в-третьих, это неспособность обрабатывать нестационарные временные ряды, зашумленные выбросами неизвестной природы. Таким образом, актуальным вопросом является разработка новых гибридных вэйвлет-нейро-фаззи-архитектур и их методов обучения, которые бы смогли учесть выше перечисленные недостатки [123, 130].

5.8. Интеллектуальный анализ данных на основе систем индуктивного моделирования и эволюционных фаззи-систем

В ряде реальных задач возникает проблема обработки очень больших массивов информации. Преодолеть эту проблему можно, разбивая тем или иным образом исходную задачу на множество подзадач низкой размерности и объединяя далее некоторым образом получаемые решения для достижения требуемого результата. Существует также проблема обработки временных рядов с короткой выборкой, так как не каждая нейронная сеть успевает качественно обучить свои параметры, а зацикливание выборки приводит к потере аппроксимирующих и экстраполирующих свойств сети. С вычислительной точки зрения наиболее удобным в этой ситуации представляется Метод Группового Учета Аргументов (МГУА), продемонстрировавший свою эффективность при решении множества различных практических задач.

МГУА является семейством методов индуктивного моделирования и одним из наиболее эффективных методов структурно-параметрической идентификации сложных объектов, которые рассматриваются как одно из направлений вычислительного интеллекта.

МГУА основывается па переборе моделей, которые постепенно усложняются, и их оценивании по внешнему критерию. Структура модели и степень влияния параметров на выходную величину определяются автоматически. Наилучшей является та модель, которая приводит к минимальному значению внешнего критерия.

МГУА был предложен А.Г. Ивахненко [69] для моделирования сложных систем, прогнозирования, идентификации и аппроксимации многофакторных систем, диагностики, распознавания образов и кластеризации выборки. Аналитически доказано, что в результате применения только этого индуктивного метода самоорганизации неточных, зашумленных и / или коротких выборок данных может быть найдена оптимальная математическая модель.

Отличие алгоритмов МГУА от других алгоритмов структурной идентификации и селекции лучшей регрессии состоит в следующих свойствах:

– использование внешнего критерия, который основывается на разделении выборки данных на обучающую, тестовую и проверочную;

– уменьшение требований к объему первичной информации; большое разнообразие структур: использование путей полного или уменьшенного перебора вариантов структур и применение многорядных итерационных процедур;

– высшая степень автоматизации достаточно ввести только первичные данные и задать внешний критерий.

Необходимо обратить внимание на разницу между оригинальными алгоритмами МГУА и "МГУА-подобными" алгоритмами. Первые находят минимум внешнего критерия и таким образом реализуют объективный выбор оптимальной модели. Этот метод основывается на индуктивном подходе: оптимальные модели находят путем перебора возможных вариантов и их оценки по так называемому внешнему критерию. Он вычисляется на отдельной части выборки, которая не используется для построения модели. Наилучшая модель может быть выбрана по двум критериям: первый отбирает лучшие модели на каждом ряде перебора в процессе структурной идентификации, а другой находит оптимальную модель. Процедура селекции останавливается, когда найдено минимальное значение критерия.

Некоторые "МГУА-подобные" алгоритмы работают по идее "чем сложнее модель тем она точнее". Это требует введения определенных порогов для весовых коэффициентов в формуле внешнего критерия при субъективном оценивании модели. Но значительные проблемы чаще всего связаны с короткими или существенно зашумленными выборками данных.

Решение практических задач и развитие теории МГУА привели к разработке широкого спектра алгоритмов. Каждый из них соответствует определенным условиям применения. Алгоритмы МГУА отличаются, главным образом, способом генерации моделей-кандидатов, подлежащих перебору по внешнему критерию. Выбор алгоритма зависит от характера решаемой проблемы, уровня дисперсии помех (возмущений), информативности и репрезентативности выборки данных.

Приведем классификацию алгоритмов МГУА для непрерывных и дискретных переменных:

1. Непрерывные переменные:

1.1. Алгоритмы МГУА параметрические:

– комбинаторный (COMBI);

– многорядный комбинаторный (MIA);

– объективного системного анализа (OSA);

– гармоничный (GA);

– двухуровневый (ARIMAD);

– мультипликативно-аддитивный.

1.2. Алгоритмы МГУА непараметрические:

– объективно-компьютерной кластеризации (ОСС);

– алгоритм кластеризации "Указательный перст" (PF);

– комплексирование аналогов (АС).

2. Дискретные переменные:

2.1. Алгоритмы МГУА параметрические:

– гармоничной редискретизации.

2.2. Алгоритмы МГУА непараметрические:

– вероятностный алгоритм на основе многорядной теории статистических решений (MTSD).

Можно выделить следующие достоинства МГУА:

1. Возможность восстановления неизвестной достаточно сложной зависимости по ограниченной выборке данных. Количество неизвестных параметров модели может быть больше нежели количество точек в выборке.

2. Возможность адаптации параметров модели при получении новых экспериментальных данных (в частности, используя рекуррентный метод наименьших квадратов).

Обобщенный итерационный алгоритм (ОИА) построения модели сложных систем на основе МГУА приведен на рис. 5.12.

Обобщенный итерационный алгоритм - это множество итерационных и итерационно-комбинаторных алгоритмов, которые определяются компонентами вектора трех индексированных множеств: DM (Dialogue Model), IC (Iterative- Combinatorial), MR(Multilayered-Relaxative), т.е. любой итерационный алгоритм определяется как отдельный случай обобщенного ОИА. При этом параметр DM принимает три значения: 1 – стандартный автоматический режим, 2 – плановый автоматический режим, 3 – интерактивный режим; IC принимает два значения: 1 – итерационный и 2 – итерационно-комбинаторный алгоритм; параметр MR принимает три значения: 1 – классический итерационный, 2 – релаксационный, 3 – комбинированный алгоритм.

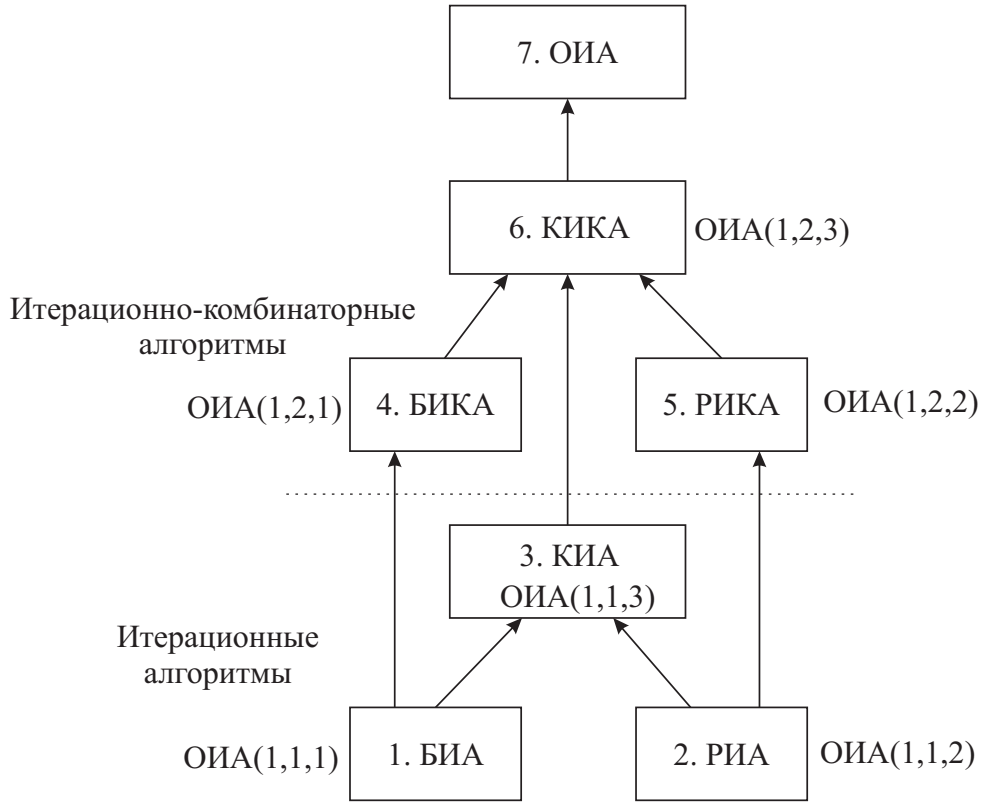


Рис. 5.12 - Иерархия архитектур итерационных алгоритмов МГУА

Модифицированный алгоритм с комбинаторной селекцией и ортогонализацией переменных, подход к рекуррентному алгоритму расчета коэффициентов и критериев селекции в релаксационном алгоритме МГУА, а также обобщенный релаксационный итерационный алгоритм МГУА [83].

При построении моделей сложных систем по экспериментальным данным довольно часто встречаются ситуации, когда входные данные заданы в виде интервалов неопределенности. Это привело к развитию и обобщению МГУА на основе фаззи-множеств. Нечеткий МГУА, дает не только точечные оценки прогнозов, а и ширину интервалов неопределенности для прогноза, что дает возможность непосредственно оценивать качество прогноза.

Исследования МГУА также привели к созданию нейронных сетей с активными нейронами, в которых реализована удвоенно-многорядная структура: нейроны с многорядной структурой объединены в многорядную сеть. В данном случае нейронами являются многорядные алгоритмы. Это дает возможность оптимизировать множество входных параметров на каждом уровне в процессе увеличения точности. Такие активные нейроны могут определять в процессе самоорганизации какие связи необходимо минимизировать согласно заданной целевой функции нейрона. Нейронная сеть такого вида рассматривается как средство для повышения точности решения задач.

Представляется целесообразным объединение идей МГУА и систем вычислительного интеллекта для получения качественно новых результатов в области динамического интеллектуального анализа данных (Dynamic

Data Mining), интеллектуального управления и других научных направлений в условиях недостаточности данных и системы пониженной размерности.

В настоящее время достаточно широко известна МГУА-нейронная сеть, узлами которой является двухвходовые N-Adalines, каждая из которых содержит набор настраиваемых синаптических весов, определяемых с помощью стандартного метода наименьших квадратов, и обеспечивает квадратичную аппроксимацию восстанавливаемого нелинейного отображения.

Для обеспечения требуемого качества аппроксимации такой ANN может потребоваться значительное количество скрытых слоев.

Таким образом, актуальной является разработка гибридных эволюционных вэйвлет-МГУА-нейро-фаззи-систем, которые бы вобрали в себя преимущества каждого из подходов, а именно: обработку данных в on-line режиме, подстройку не только параметров сети, но и её структуры во время обучения, а также возможность выявлять локальные особенности нестационарных сигналов.

5.9. Методы прогнозирования и эмуляции на основе вэйвлет-нейронных сетей и вэйвлет-нейро-фаззи-систем

Еще одним направлением развития систем вычислительного интеллекта являются вэйвлет-нейронные сети [130]. Вэйвлет-нейронная сеть была введена как аппроксиматор непрерывных функций на основе универсальных аппроксимирующих свойств вэйвлет-декомпозиции, со структурой, которая описывается с помощью алгоритма декомпозиции

$$g(x)=\sum_{i=1}^{N} w_i \varphi[D_i R_i (x-t_i)] + g^*, \tag{5.40}$$

где D_i – диагональная матрица, построенная из векторов растяжения и R_i – некоторой матрицы поворота. Параметр g^* вводит нормировку для функции с ненулевым средним, так как вэйвлет-функция φ имеет нулевое среднее, и в общем виде представляется выражением

$$\varphi_{\alpha,\beta}=\frac{1}{\sqrt{\alpha}}\varphi\left(\frac{t-\beta}{\alpha}\right). \tag{5.41}$$

Недостатком данной сети является, то, что при её реализации использован только алгоритм обучения весовых коэффициентов, а параметры вэйвлет-функции активации выбираются вручную.

Теорема Колмогорова-Арнольда, гласит, что пусть φ – ограниченная монотонно возрастающая непрерывная функция. Пусть I_{m_0} – m_0-мерный единичный гиперкуб $[0,1]^{m_0}$, а пространство непрерывных на I_{m_0} функций – $C(I_{m_0})$. Тогда для любой функции $f \in C(I_{m_0})$ и $\varepsilon > 0$ существует такое

целое число m_1 и множество действительных констант a_i, b_i, u w_{ij} где $i=1,\ldots,m_1$, $j=1,\ldots,m_0$, что выполняется равенство

$$F(x_1,...,x_{m_0})=\sum_{i=1}^{m_1}a_i\varphi\left(\sum_{j=1}^{m_0}w_{ij}x_j+b_i\right). \tag{5.42}$$

Которая представляет собой аппроксимацию многомерной функции $f\left(x_1,...,x_{m_0}\right)$ в виде суперпозиции одномерных при

$$\left|F(x_1,...,x_{m_0})-f(x_1,...,x_{m_0})\right|<\varepsilon. \tag{5.43}$$

С другой стороны, существуют аналогичные результаты вэйвлет-теории, в которой доказано, что произвольная функция может быть представлена в виде взвешенной суммы вэйвлет-функций с различными параметрами сдвига (центра) и растяжения (ширины)

$$f(x)=\sum_{i=1}^{N}w_i\det(D^{1/2})\varphi(D_ix-t_i). \tag{5.44}$$

При определенных предположениях можно увидеть близость результатов этих теорем.

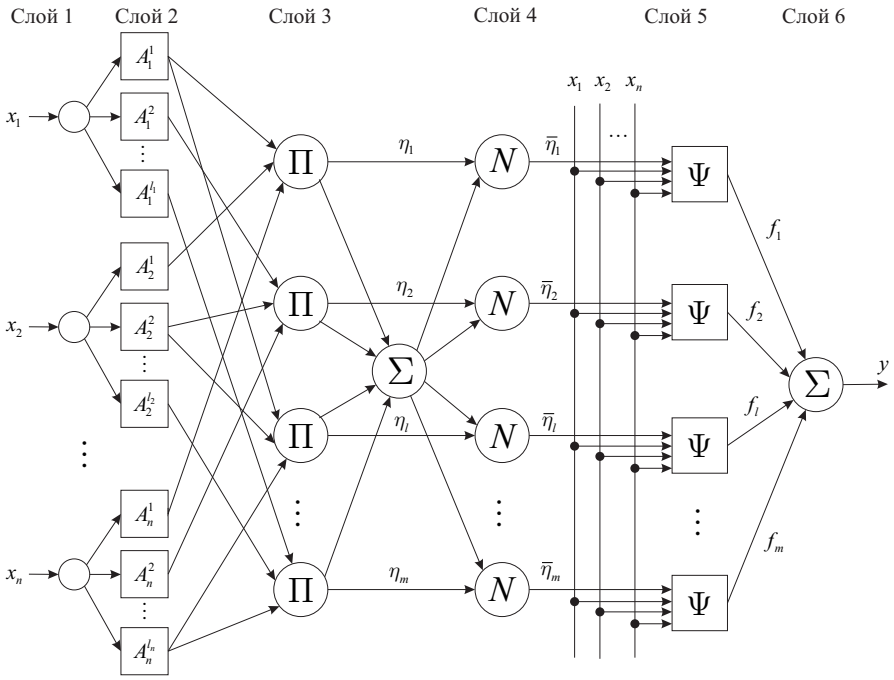


Рис. 5.13 - Фаззи-вэйвлет-нейронная сеть

Недостатком описанных выше сетей является сугубо эмпирический подбор шага градиентного обучения, что приводит к погрешности прогноза, а также низкая скорость сходимости алгоритмов обучения. С другой стороны архитектура таких сетей задана заранее и может быть как избыточна, так и недостаточна при обработки тех или иных данных.

Дальнейшее развитие этого направлении привело к созданию вэйвлет-иейро-фаззи-систем типа-2 [130, 145]. Такие системы представляют собой распараллеленную структуру (см. рис. 5.14), в слоях которой находятся вэйвлет-функции с различными начальными параметрами и далее выходы каждой из подсистем объединяются в блоке редукции модели.

Недостатком существующих фаззи-вэйвлет-нейронных сетей типа-2 являются низкие экстраполирующие свойства сетей за счет того, что блок редукции представляет собой либо простое усреднение выходов подсистем, либо элементарную оптимизационную процедуру, в которой коэффициенты подбираются эмпирическим образом, что влияет на качество получаемых результатов.

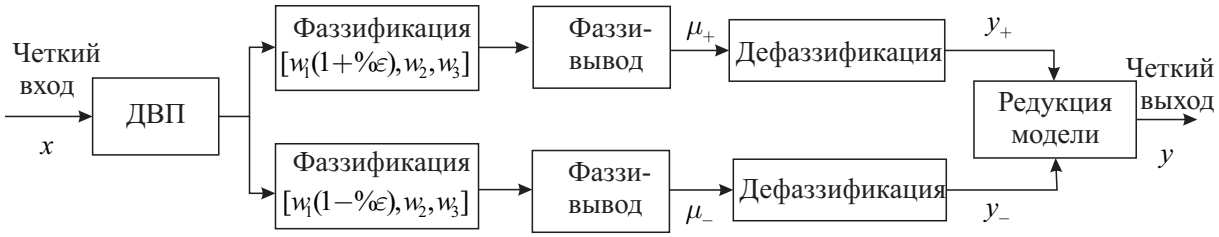


Рис. 5.14 - Вэйвлет-нейро-фаззи-система типа-2

Таким образом, актуальной проблемой является разработка новых гибридных вэйвлет-нейро-фаззи-систем с быстросходящимися методами обучения, которые бы позволили не только обучать синаптические веса, но и все параметры вэйвлет-функций активации-принадлежности, которые позволят сделать лингвистический нечеткий вывод, подобно тому, как это делается на функциях принадлежности, а также настраивать вид вэйвлет-функций, что позволит повысить в целом качество анализа и обработки нестационарных временных рядов произвольной природы и учтет выше перечисленные недостатки известных методов.

В настоящее время к методам динамического интеллектуального анализа данных предъявляются высокие требования, поскольку современные технологии позволяют съем и накопление достаточно больших массивов информации.

В общем случае задача, которую решает динамический интеллектуальный анализ данных, состоит в обнаружении скрытых закономерностей в данных различной природы. Более подробно они сочетают в себе регрессию, прогнозирование, эмуляцию, идентификацию, кластеризацию, компрессию, диагностику и т.п.

Рассмотренные выше существующие архитектуры нейро-фаззи-, вэйвлет-нейро-фаззи-сети являются универсальными не только с точки зрения универсальных аппроксимирующих возможностей, но и в смысле того, что различные входные процессы такими системами обрабатываются подобным образом. Это следует из гомогенности и полносвязаности рассмотренных архитектур. В результате имеем некий универсальный "черный ящик", способный решать широкий класс задач обработки нестацио-

нарных сигналов с определенной точностью. Однако при этом возникает ряд проблем – эта универсальность и точность достигаются либо за счет неограниченного роста числа нейронов и настраиваемых параметров сети, что влечет за собой необходимость больших выборок данных для обучения, а также снижение обобщающих и экстраполирующих свойств сетей, а в некоторых случаях, из-за необходимого переобучения, запоминание сетью обучающей выборки.

Еще одной проблемой в существующих нейро-фаззи-системах является неудачный выбор параметров и вида активационных функций, что приводит к некачественному решению поставленной задачи, а также к низкой скорости сходимости алгоритмов обучения, что не дает возможность использовать имеющиеся системы в реальном времени.

Именно эти недостатки породили необходимость создания динамического интеллектуального анализа данных, который ещё находится на начальной стадии своего развития. Динамический анализ данных позволяет осуществить синтез модели объекта при последовательном поступлении выборок данных в on-line режиме как во временной, так и в пространственной области.

Использование стандартных вэйвлет-нейро-фаззи-систем требует высокой квалификации пользователя, которому необходимо уметь задавать не только архитектуру сети, но и выбирать активационные функции и все их параметры для конкретно решаемой задачи. В этой связи для того, чтобы расширить круг решаемых задач пользователя, гибридная система должна уметь сама настраивать свою архитектуру (что позволяет теория эволюционных систем), выбирать вид функций активации-принадлежности (что позволяет теория фаззи-систем типа-2), настраивать параметры функций активации-принадлежности (что позволяет применение вэйвлет-нейро-фаззи-алгоритмов), а также задавать логическую интерпретируемость полученным результатам (что позволяет теория фаззи-систем типа-1) без вмешательства экспертов, которые, зачастую, могут просто отсутствовать.

Из вышеизложенного следует, что на сегодняшний день в теории вычислительного интеллекта, а именно: в динамическом интеллектуальном анализе данных, остро стоит научная проблема построения гибридных эволюционных систем, а именно гибридных эволюционных адаптивных вэйвлет-нейро-фаззи-систем, которые бы позволили учитывать априорную информацию о свойствах моделируемых сигналов, влияющих на них факторов, локальных особенностях сигналов, а также разработка новых методов обучения таких систем в последовательном on-line режиме и разработка нового класса адаптивных вэйвлет-функций активации - принадлежности, чтобы свести до минимума эмпирический подбор параметров системы человеком.

Контрольные вопросы к разделу V:

1. Что такое динамический интеллектуальный анализ данных и отличие от стандартного интеллектуального анализа данных?
2. Основные типовые подзадачи компрессии многомерных массивов информации?
3. В чем состоит алгоритм сегментации нестационарных временных рядов?
4. Особенности задачи прогнозирования на основе искусственных нейронных сетей?
5. В чем состоят нейросетевые методы обнаружения изменения свойств стохастической последовательности?
6. Преимущества и недостатки интеллектуального управления на основе систем вычислительного интеллекта?
7. Приведите классификация временных рядов?
8. Зачем нужна предобработка массивов входных данных?
9. Отличие нейро-фаззи-систем и фаззи-нейро-систем, преимущество и недостатки?
10. В чем состоит алгоритм МГУА?
11. Привести структурную схему нейро-фаззи сети ANFIS.
12. В чём состоит суть фаззи-вэйвлет нейронной сети?
13. Теорема Колмогорова-Арнольда и её значение для развития динамического интеллектуального анализа данных.
14. Привести классификацию алгоритмов МГУА для непрерывных переменных.
15. Привести классификацию алгоритмов МГУА для дискретных переменных.
16. В чём отличие МГУА и МГУА-подобных алгоритмов?
17. Привести архитектуру фаззи-вэйвлет нейронной сети, её достоинства и недостатки.
18. Вэйвлет-нейро-фаззи системы типа-2, преимущества над системами типа-1.

ПРИЛОЖЕНИЕ. *БАЗОВАЯ ИНФОРМАЦИЯ*

В данном приложении освещаются вопросы, относящиеся к базовым понятиям линейной алгебры и теории вероятности, представляющие собой математическую основу рассматриваемых в данном вводном курсе подходов и методов по интеллектуальной обработке данных.

П.1 Элементы линейной алгебры

В данном подразделе изложена информация, необходимая при выявлении в исходных данных ассоциаций и закономерностей (обозначение номеров рисунков, таблиц, формул по шаблону (П.1, П.2,..,П.N)). [46,51,144]

Напомним, что для матрицы A размером $n \times m$ под транспонированием понимаем замену строк столбцами с теми же номерами, в частности, если

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \\ x_{4,1} & x_{4,2} \end{bmatrix},$$

то

$$X^T = \begin{bmatrix} x_{1,1} & x_{2,1} & x_{3,1} & x_{4,1} \\ x_{1,2} & x_{2,2} & x_{3,2} & x_{4,2} \end{bmatrix}.$$

Важной характеристикой, используемой в дальнейшем, является скалярное произведение двух векторов

$$\langle X, Y \rangle = X^T Y = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n = \sum_{i=1}^{n} x_i y_i ,$$

где

$$X^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \text{ и } Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

С другой стороны, замечая, что $\langle X, Y \rangle = |X||Y|\cos\theta$, где θ угол между векторами X и Y, имеем

$$\cos\theta = \frac{X^T Y}{|X||Y|}.$$

Таким образом, скалярное произведение характеризует отклонение вектора X от вектора Y. Евклидовой метрикой или длиной вектора называется число

$$|X| = \sqrt{\langle X, X \rangle} = \sqrt{\sum_{i=1}^{n} x_i^2} ,$$

а евклидовым расстоянием между двумя векторами назовем число

$$|X-Y| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

Векторы $X_1, X_2, \ldots, X_m$ называются линейно независимыми, если равенство

$$\alpha_1 X_1 + \alpha_2 X_2 + \ldots + \alpha_m X_m = 0$$

выполняется тогда и только тогда, когда все $\alpha_i = 0, i = 1,2,\ldots,m.$

Если выполнение этого условия возможно хотя бы при одном $\alpha_i \neq 0,$ то эта система векторов является линейно зависимой.

Множество всех векторов размерности n называется векторным пространством V той же размерности.

Множество векторов $\{U_1, U_2, \ldots, U_m\}$ называется базисом векторного пространства V, если для $\forall v \in V$ найдется такое множество $\{\alpha_i\}_{i=1}^{m}$, что

$$v = \alpha_1 U_1 + \alpha_2 U_2 + \ldots + \alpha_m U_m .$$

Базис $\{U_1, U_2, \ldots, U_m\}$ называется ортогональным, если $U_i \perp U_j$ для $\forall i \neq j$ (т.е. $\langle U_i, U_j \rangle = 0$), если же при этом $|U_i| = 1, i = 1,\ldots,m$, то базис называется ортонормированным.

Если для квадратной матрицы A размером $m \times m$ найдется ненулевой вектор X такой, что выполняется условие $AX = \lambda X$, то X называется собственным вектором матрицы A, а число λ называется собственным значением матрицы. Таким образом, линейное преобразование, реализованное матрицей A, переводит собственный вектор X в коллинеарный, направленный в ту же сторону, если λ>0, и в обратную, если λ<0.

Отметим несколько важных свойств собственных чисел.

- Если матрица A действительна и симметрична (то есть $A^T = A$), то все собственные значения действительны.
-
- Если матрица не сингулярная (то есть ее ранг равен числу строк (столбцов)), то ее собственные числа не нулевые.
-
- Если матрица A положительно определена (то есть квадратичная форма $X^T AX > 0$), то все ее собственные числа положительны.

П.2 Элементы теории вероятностей

Пусть Ω - множество всех возможных исходов некоторых событий, и S – алгебра событий, то есть S совокупность подмножеств множества Ω, для которого выполнены следующие условия:

1. S содержит невозможное и достоверное события.
2. Если события $A_1, A_2, \ldots$ (конечное или счетное множество) принадлежит S, то S принадлежит сумма, произведение и дополнение этих событий.

Вероятностью называется функция $P(A)$, определенная на S, принимающая действительные значения и удовлетворяющая аксиомам:

- Аксиома неотрицательности: $\forall A \in S : P(A) \geq 0.$
- Аксиома нормированности: вероятность достоверного события равна единице: $P(\Omega) = 1.$
- Аксиома аддитивности: вероятность суммы несовместных событий равна сумме вероятностей этих событий: если $A_i \cap A_j = \varnothing (i \neq j),$ то
- $$P\left(\bigcup_k A_k\right) = \sum_k P(A_k).$$

Приведем свойства вероятности.

1. $P(\varnothing) = 0;$
2. $P(A) \leq 1;$
3. $A \subset B \Rightarrow P(A) < P(B);$
4. $P(A \cup B) = P(A) + P(B) - P(A \cap B).$

Вероятность того, что произойдет событие A при условии, что произошло событие B, называется условной вероятностью $P(A|B)$ и вычисляется следующим образом

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}.$$

В случае, если события A и B независимы, то $P(A \cap B) = P(A)P(B)$, и для условной вероятности можно записать

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)P(B)}{P(B)} = P(A).$$

Пусть

$$A = (A \cap B_1) \cup (A \cap B_2) \cup (A \cap B_3) \cup \ldots \cup (A \cap B_n),$$

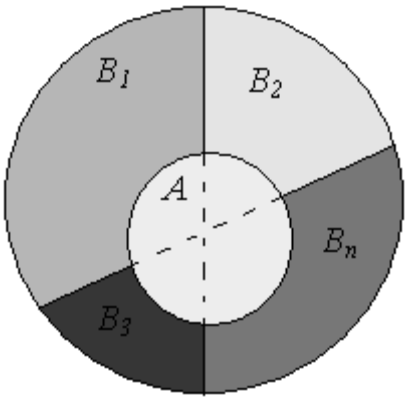


Рис.П.1 - Иллюстрация формулы полной вероятности

тогда из формулы условной вероятности получаем

$$P(A) = P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + ... + P(A|B_n)P(B_n),$$

то есть

$$P(A) = \sum_{k=1}^{n} P(A \mid B_k)P(B_k).$$

Важную роль в дальнейших рассуждениях играет формула Байеса или теорема гипотез[*]. Это утверждение позволяет переоценить вероятность гипотез B_i, принятых до опыта (события) и называемых априорными (a priori – до опыта) по результатам уже проведенного опыта, то есть используя апостериорные (a posteriori - после опыта) вероятности. Пусть $B_1, B_2, ..., B_n$ составляющие множества S, тогда вероятность того, что событие A приведет к событию B_i будет равна

$$P(B_i \mid A) = \frac{P(B_i \cap A)}{P(A)} = \frac{P(A \mid B_i)P(B_i)}{\sum_{k=1}^{n} P(A \mid B_k)P(B_k)}.$$

Случайной величиной X называется функция, определенная на множестве событий Ω, которая каждому элементарному событию ω ставит в соответствие число $X(\omega)$. При этом случайная величина может быть дискретной и непрерывной с точки зрения её наблюдения.

Любое правило, позволяющее находить вероятности произвольных событий $A \subseteq S$, называется законом распределения случайной величины, и при этом говорят, что случайная величина подчиняется данному закону распределения.

Функцией распределения случайной величины X называется функция $F(x)$, которая для любого $x \in R$ равна вероятности события $\{X \le x\}$

$$F(x) = P\{X \le x\}.$$

Функция распределения обладает следующими свойствами

1. $0 \le F(x) \le 1$.
2. $F(x)$ неубывающая функция, то есть, если $x_2 > x_1$, то $F(x_2) \ge F(x_1)$.

[*] Байес Томас Реверенд (1702 — 17.04.1761) — английский математик, пресвитерианский священник XVII века.

3. $F(-\infty)=0, F(+\infty)=1.$
4. $P\{a \le X < b\} = F(b) - F(a).$

Для дискретной случайной величины X положим

$$p(x) = P(X = x),$$

тогда

$$F(x) = P(X \le x) = \sum_{a \le x} P(X = x) = \sum_{a \le x} p(x).$$

Для непрерывной случайной величины функцию $f(x) \ge 0$ назовем функцией плотности распределения вероятностей, если

$$F(a) = P(X \le a) = \int_{-\infty}^{a} f(x)dx\,.$$

Тогда

$$P(a \le x \le b) = \int_{a}^{b} f(x)dx.$$

Заметим, что производная от функции распределения есть функция плотности распределения.

$$\frac{d}{dx}F(x) = f(x),$$

кроме того,

$$P(x = a) = \int_{a}^{a} f(x)dx = 0, \text{ и } P(-\infty \le x \le \infty) = \int_{-\infty}^{\infty} f(x)dx = 1.$$

Рассмотрим некоторые важные характеристики случайной величины. Первый момент случайной величины называется математическим ожиданием. Для дискретного случая математическое ожидание будет иметь вид

$$\mu = E(X) = \sum_{x} xp(x)\,,$$

для непрерывного случая

$$\mu = E(X) = \int_{-\infty}^{\infty} xf(x)dx.$$

Величина

$$\sigma^2 = \mathrm{var}(X) = E\left((X - E(X))^2\right)$$

называется вариацией (variance) или дисперсией. Это число характеризует разброс случайной величины, а σ называется среднеквадратичным отклонением (СКО) случайной величины от математического ожидания.

Особую роль в теории вероятностей играет нормальный закон (закон Карла Гаусса), что обусловлено, прежде всего, тем фактом, что он является предельным законом, к которому приближаются (при определенных условиях) другие законы распределения (в соответствии с центральной предельной теоремой).

Будем говорить, что непрерывная случайная величина X распределена по нормальному закону $N(\mu,\sigma)$, если ее плотность распределения имеет вид (функция Гаусса)

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), x\in R.$$

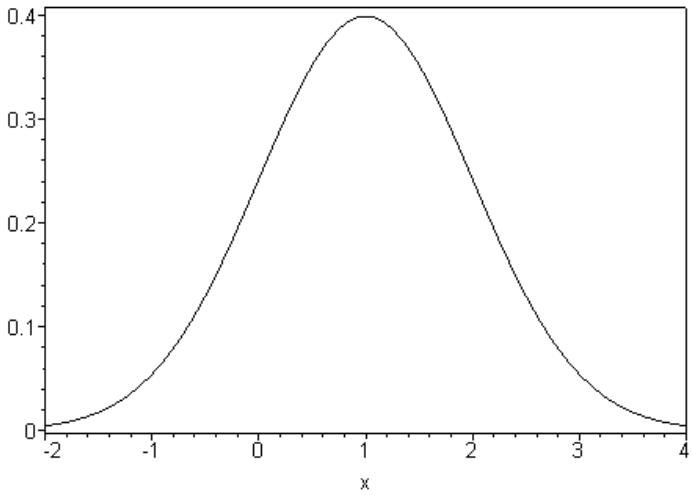


Рис.П.2. - График функции плотности нормального распределения (функция Гаусса) для $\mu=1,\sigma=1.$

Заметим, что, в соответствии с нормальным законом распределяются самые различные величины, например, ошибки измерений, износ деталей в механизмах, вес плодов и животных, рост человека, колебания курса акций и многое другое.

Несмотря на важность одномерного случая, более актуальным является рассмотрение случая многих переменных.

Упорядоченный набор $(X_1,X_2,...,X_n)$ случайных величин X_i $(i=1,2,...,n)$, заданных на одном и том же множестве Ω называется n-мерной случайной величиной. Одномерные случайные величины $X_1,X_2,...,X_n$ называются компонентами *n*-мерной случайной величины. Компоненты удобно рассматривать как координаты случайного вектора $X=(X_1,X_2,...,X_n)$ в пространстве *n* измерений.

Упорядоченная пара (X,Y)двух случайных величин называется двумерной случайной величиной. Полной характеристикой системы (X,Y) является ее закон распределения вероятностей, указывающий область возможных значений системы случайных величин и вероятностей этих значений.

Функцией распределения двумерной случайной величины (X,Y) называется функция $F(x,y)$, которая для любых двух действительных чисел *x* и *y* равна вероятности совместного выполнения двух событий $\{X\le x\}$ и $\{Y\le y\}$, то есть

$$F(x,y)=P\{X\le x,Y\le y\}=P(\omega\in\Omega,X(\omega)\le x,Y(\omega)\le y).$$

Для дискретной случайной пары (X,Y) в качестве функции плотности положим

$$p(x,y) = P(X = x, Y = y),$$

в непрерывном случае $f(x,y) \geq 0$ назовем функцией плотности распределения вероятностей, если

$$F(a,b) = P(X \leq a, Y \leq b) = \int_{-\infty}^{a}\int_{-\infty}^{b} f(x,y)dxdy\,.$$

Тогда

$$P(a \leq x \leq b, c \leq y \leq d) = \int_{a}^{b}\int_{c}^{d} f(x,y)dxdy$$

Заметим, что

$$\frac{\partial^2}{\partial x \partial y} F(x,y) = f(x,y).$$

Важную роль в дальнейших исследованиях играет смешанный момент второго порядка, называемый ковариацией

$$\operatorname{cov}(X,Y) = E\Big(\big(X - E(X)\big)\big(Y - E(Y)\big)\Big) = E(XY) - E(X)E(Y) = \mu_{xy} - \mu_x\mu_y.$$

В координатной форме ковариацию можно записать в виде

$$\operatorname{cov}(X,Y) = \sum_{i=1}^{n}\sum_{j=1}^{m}(x_i - \mu_X)(y_j - \mu_Y)p_{i,j}\ ,$$

где $p_{i,j} = p(x_i, y_j).$

Если для двух случайных величин X и Y при возрастании одной из них наблюдается тенденция к возрастанию другой, то $\operatorname{cov}(X,Y) > 0$, а если при возрастании одной случайной величины наблюдается тенденция к убыванию другой, то $\operatorname{cov}(X,Y) < 0$.

Если же поведение не предсказуемо, тогда $\operatorname{cov}(X,Y) = 0$. В этом случае говорят, что случайные величины некоррелируемы, но это не значит, что они независимы, хотя и для независимых случайных величин $\operatorname{cov}(X,Y) = 0$. *Нормализованную ковариацию называют корреляцией или коэффициентом корреляции*:

$$-1 \leq cor(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sqrt{\operatorname{var}(X)\operatorname{var}(Y)}} \leq 1.$$

Пусть $X = (X_1, X_2, ..., X_n)$ вектор, координаты которого являются случайными величинами, тогда

$$\operatorname{cov}(X) = \operatorname{cov}(X_1, X_2, ..., X_n) = \Sigma = E\Big((X-u)(X-u)^T\Big) =$$

$$= \begin{pmatrix} E\big((X_1-\mu_1)(X_1-\mu_1)\big) & \cdots & E\big((X_n-\mu_n)(X_1-\mu_1)\big) \\ \vdots & \ddots & \vdots \\ E\big((X_1-\mu_1)(X_n-\mu_n)\big) & \cdots & E\big((X_n-\mu_n)(X_n-\mu_n)\big) \end{pmatrix}.$$

При этом матрица Σ называется ковариационной.

Для случая многих переменных непрерывная *n*-мерная случайная величина **X** распределена по нормальному закону $N(\mu,\Sigma)$, если ее плотность распределения имеет вид

$$f(x)=\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}\left((x-\mu)^T\Sigma^{-1}(x-\mu)\right)\right)=$$

$$=\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}\left((x_1-\mu_1,...,x_n-\mu_n)\Sigma^{-1}\begin{pmatrix}x_1-\mu_1\\ \vdots\\ x_n-\mu_n\end{pmatrix}\right)\right),$$

Здесь Σ ковариационная матрица, |Σ| - ее определитель и Σ^{-1} матрица, обратная к ковариационной.

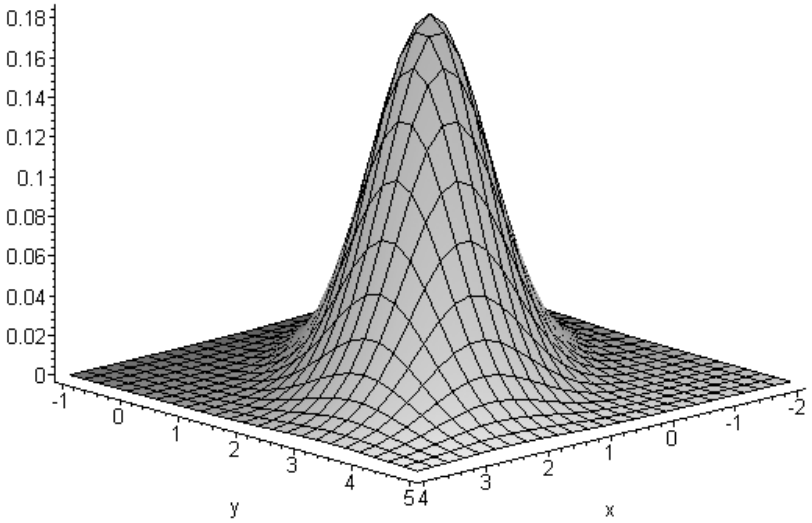


Рис. П.3. - График функции плотности

$$N\left(\begin{pmatrix}1\\2\end{pmatrix},\begin{pmatrix}1 & 0.5\\ 0.5 & 1\end{pmatrix}\right)$$

Если все величины $(X_1,X_2,...,X_n)$ независимы, то функция плотности будет иметь вид

$$f(x)=\prod_{i=1}^{n}\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right).$$

Пусть Φ матрица, столбцы которой являются нормированными собственными векторами матрицы Σ, тогда (в силу ортонормированности)

$$\Phi^{-1}=\Phi^T.$$

Если $\boldsymbol{\Sigma\Phi=\Phi\Lambda}$, то Λ диагональная матрица с соответствующими собственными значениями ковариационной матрицы Σ на диагонали, тогда $\Sigma=\Phi\Lambda\Phi^{-1}$ и, следовательно, $\Sigma^{-1}=\Phi\Lambda^{-1}\Phi^{-1}$. Через $\Lambda^{-1/2}$ обозначим матрицу, такую, что $\Lambda^{-1/2}\Lambda^{-1/2}=\Lambda^{-1}$, тогда $\Sigma^{-1}=\left(\Phi\Lambda^{-1/2}\right)\left(\Phi\Lambda^{-1/2}\right)^T=\Xi\Xi^T$.

Таким образом,

$$\left((x-\mu)^T \Sigma^{-1}(x-\mu)\right) = \left((x-\mu)^T \Xi\Xi^T (x-\mu)\right) = \left(\Xi^T (x-\mu)\right)^T \left(\Xi^T (x-\mu)\right) = \left|\Xi^T (x-\mu)\right|^2 .$$

Замечая, что матрица Ξ представляет собой матрицу преобразований (поворотов и растяжений), получаем, что точки x, удовлетворяющие условию $\left|\Xi^T (x-\mu)\right|^2 \equiv const$ лежат на эллипсе.

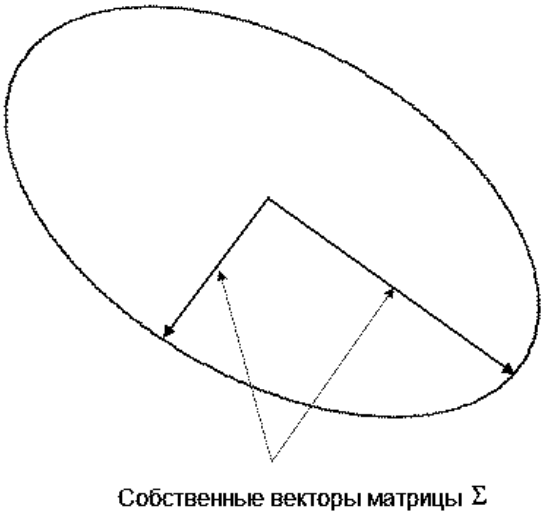


Рис.П.4 - Множество точек, равноудаленных от центра в смысле расстояния Махаланобиса

Число $\sqrt{\left((x-\mu)^T \Sigma^{-1}(x-\mu)\right)}$ называется расстоянием Махаланобиса (Mahalanobis) между x и μ. В частности, если все величины $(X_1, X_2, ..., X_n)$ независимы, то в этом случае расстояние Махаланобиса вырождается в Евклидово расстояние $\sqrt{\left((x-\mu)^T (x-\mu)\right)}$.

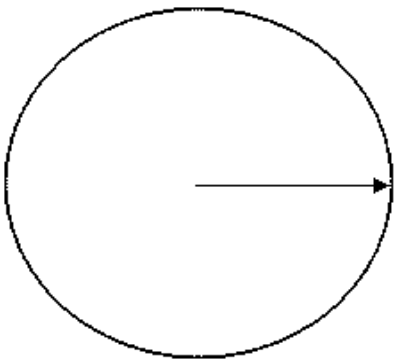


Рис.П.5 - Множество точек равноудаленных от центра в смысле расстояния Евклида

Приведем пример двумерной функции Гаусса.

Пусть $\mu = (1,2)$, $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$, тогда линии уровня соответствующей функции Гаусса будут иметь вид

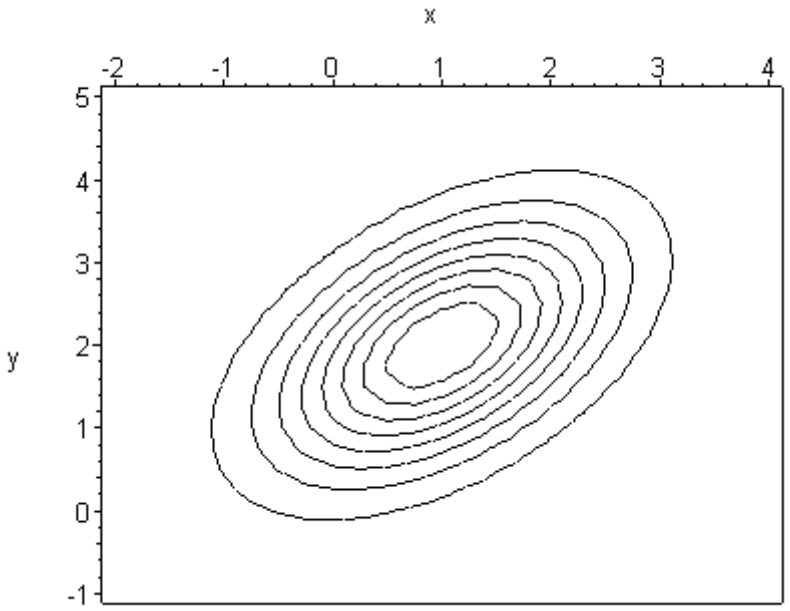


Рис.П.6 - Изолинии функции Гаусса при $\mu = (1,2)$, $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$

Если $\mu = (1,2)$, $\Sigma = \begin{pmatrix} 1 & 0. \\ 0 & 1 \end{pmatrix}$, то

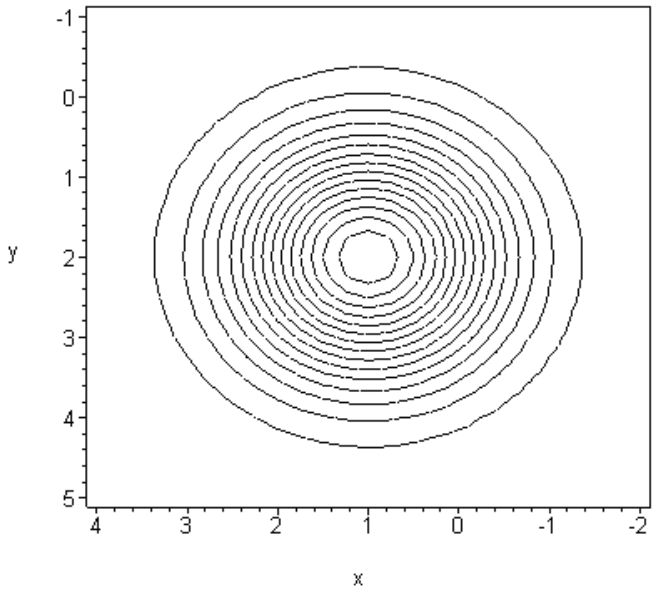


Рис.П.7 - Изолинии сферической функции Гаусса при $\mu = (1,2)$, $\Sigma = \begin{pmatrix} 1 & 0. \\ 0 & 1 \end{pmatrix}$

Если *n*-мерная случайная величина ***X*** имеет плотность $N(\mu,\Sigma)$, то величина AX имеет плотность $N(A^T\mu, A^T\Sigma A)$.

Таким образом, для любой случайной величины **X** можно подобрать преобразование, переводящее в случайную величину со сферической плотностью распределения.

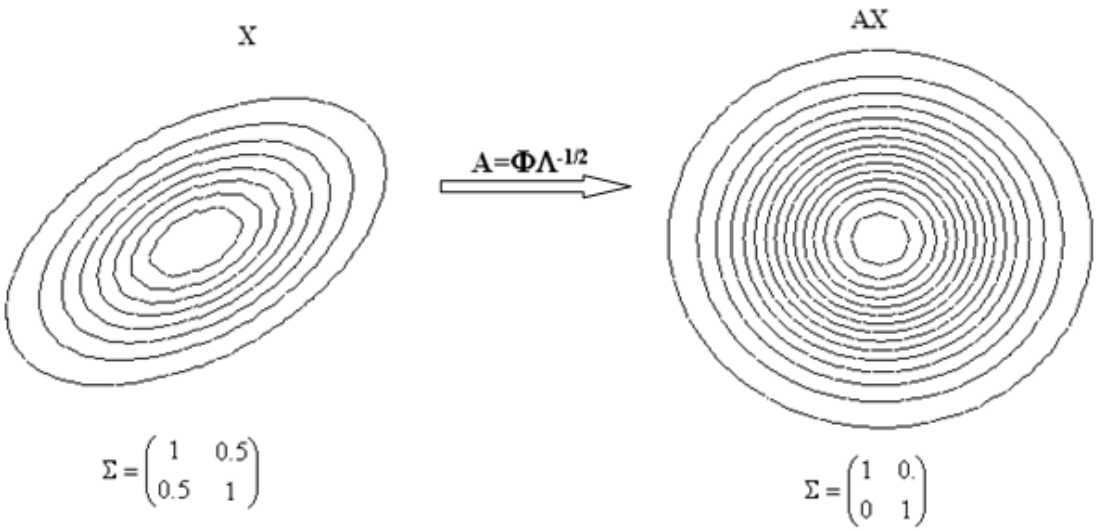


Рис.П.8 - Преобразование случайной величины с плотностью распределения при $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$ *в случайную величину со сферической функцией плотности.*

ЛИТЕРАТУРА

1. Михалев А.И. Цифровая обработка данных: от Фурье к Wavelets. – Днепропетровск: Системные технологии, 2007. – 200 с.
2. Сергиенко А.Б. Цифровая обработка сигналов. 2-ое издание. – СПб: Питер, 2006. – 751 с.
3. Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа. - М.: Наука, 1976. - 542 с.
4. Бендат Дж., Пирсол А. Измерение и анализ случайных процессов. – М.: Мир, 1974. – 464 с.
5. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление. – Т.1,2. - М.: Мир, 1974.
6. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. — М.: Мир, 1989. — 448 с.
7. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления свертки. — М.: Радио и связь, 1985. — 248 с.
8. Цифровые процессоры обработки сигналов: Справочник / А.Г. Остапенко, С.И. Лавлинскин, А.Б. Сушков и др. — М.: Радио и связь, 1994. —264 с.
9. Коршунов Ю.М., Бобиков А.И. Цифровые сглаживающие и преобразующие системы. — М.: Энергия, 1969. — 128 с.
10. Кузьмин С.З. Цифровая обработка радиолокационной информации. — М.: Сов. радио, 1974. — 432 с.
11. Signal processing handbook. /Ed. by C.H-Chen. - New York: Dekker, 1989. — 818 p.
12. Макс Ж. Методы и техника обработки сигналов при физических измерениях: В 2-х томах. Пер. с франц. – М.:Мир,1983.
13. Оппенгейм А.В., Шафер Р.В. Цифровая обработка сигналов. 2-ое переработанное издание. — М.: Техносфера, 2006. — 856 с.
14. Малла С. Вейвлеты в обработке сигналов.–М.: Мир, 2005. -671 с., ил.
15. Дьяконов В.П. Вейвлеты. От теории к практике. – М.: СОЛОН-Р, 2002. – 448 с.
16. Karhunen H. On Linear Methods in Probability Theory /English translation by Selin I., 1947 //The Rand Corporation, Doc. T-131. - August 1960.
17. Котельников В.А. О пропускной способности эфира и проволоки в радиосвязи. — М.: Изд.-во Всесоюзного Энергетического Комитета: МГУ, 1933.
18. Линник В.И. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений / В.И. Линник .− М. : Физматгиз, 1962.

19. Мандель И.Д. Кластерный анализ / И.Д. Мандель .− М. : Финансы и статистика, 1988 .− 176 с.
20. Бодянський Є.В., Михальов О.І., Плісс І.П. Адаптивне виявлення розладнань в об'єктах керування за допомогою штучних нейронних мереж. – Дніпропетровськ: Системні технології, 2000. – 140 с.
21. Прикладная статистика: Классификация и снижение размерности / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин .− М. : Финансы и статистика, 1989 .− 450 с.
22. Poncelet P. Data Mining Patterns: New Methods and Applications / P. Poncelet, M. Teisseire, F. Masseglia // Information science reference, Hershey .− New York, 2008 .− 307 p.
23. Айвазян С.А. Классификация многомерных наблюдений / С.А. Айвазян, Э.Н. Бежаева, О.В. Староверов .− М., 1974 .− 238 с.
24. Айвазян С.А. Прикладная статистика: Основы моделирования и первичная обработка данных / С.А. Айвазян, И.С. Енюков, Л.Д. Мешалкин .− М. : Финансы и статистика, 1983 .− 471 с.
25. Wu C.F.G. On the convergence properties of the EM algorithm / C.F.G. Wu // The Annals of Statistics .− 1983 .− no. 11 .− P. 95–103. (http://citeseer.ist.psu.edu/78906.html.)
26. Kleinberg J. Two algorithms for nearest-neighbor search in high dimensions / Jon Kleinberg. (http://citeseer.ist.psu.edu/kleinberg97two.html)
27. Михалёв А.И., Бабенко Ю.В. Оценка работы генетического алгоритма с модифицированными операторами мутации и генерации начальной популяции //Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 2 (79). – Днепропетровск, 2012. – С. 124-129
28. Jain A.K. Data Clustering / A.K. Jain, M.N. Murty, P.J. Flynn // A Review (http://www.csee.umbc.edu/ nicholas/clustering/p264-jain.pdf)
29. Mitchell M. An Introduction to Genetic Algorithms / M. Mitchell .− Cambridge, MA : The MIT Press, 1996.
30. Вапник В.Н. Восстановление зависимостей по эмпирическим данным / В.Н. Вапник .− М. : Наука, 1979.
31. Технология АКС-анализ клиентских сред / ЗАО "Форексис" .− 2005. (http://www.forecsys.ru/cea.php)
32. Cramer N.L. A representation for the adaptive generation of simple sequential programs / N.L. Cramer // Proceedings of an International Conference on Genetic Algorithms and the Applications .− Pittsburg, PA, : Carnegie-Mellon University, 1985 .− P. 183-187.
33. Лем С. Солярис. Эдем. Непобедимый / С. Лем .− М. : АСТ, 2003.
34. Ghazanfar M.A. An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering / Mustansar Ali Ghazanfar, Adam Prugel-Bennett // Proceeding of the International Multi

Conference of Engineers and Computer Scientists .− Hong Kong .− 2010 .− Vol 1.

35. Башмаков А.И. Интеллектуальные информационные технологии: Учебное пособие / А.И. Башмаков, И.А. Башмаков .− М. : Изд-во МГТУ им. Н.Э. Баумана, 2005 .− 304 с.
36. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности /Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев .− Харьков : Основа, 1997 .− 112 с.
37. Бодянский Е.В. Методы вычислительного интеллекта для анализа данных / Бодянский Е.В. // Радіоелектронні і комп'ютерні системи. – Харьков: ХАИ, 2007. – №5 (24). – С. 66–76.
38. Верченко А.П. Рекуррентные средства кластеризации в применении к задачам сегментации изображений / Верченко А.П., Кириченко Н.Ф., Лепеха Н.П. // Проблемы управления и информатики. – 2005. – №5. – С. 62–71
39. Земсков В.Н. Сжатие изображений на основе автоматической классификации / Земсков В.Н., Ким И.С. // Известия вузов. Электроника, 2003. – № 2. – С. 50–56.
40. Мельник Р.А. Кластерізація мікрообразів для кодування зображень / Мельник Р.А., Алексеев О.А. // Праці 7–ї Всеукраїнської міжнародної конференції «Оброблення сигналів і зображень та розпізнавання образів»(УкрОБРАЗ'2004). – Киев: Кібернетичний центр НАН України, 2004. – С. 81–84.
41. Yao J. Automatic Segmentation of Colonic Polyps in CT Colonography Based on Knowledge-Guided Deformable Models / Yao J., Miller M., Franaszek M., Summers R. // Proceedings of SPIE, Medical Imaging 2003: Physiology and Function: Methods, Systems, and Applications, Vol. 5031. – 2003. – P. 370–380.
42. Hammah R.E. Validity measures for the fuzzy cluster analysis of orientations / Hammah, R.E., Curran, J.H. – Pattern Analysis and Machine Intelligence, IEEE Transactions. – Volume 22, Issue 12, Dec 2000. –P. 1467–1472.
43. Moller-Levet C.S. Clustering of unevenly sampled gene expression time-series data / Moller-Levet C.S., Klawonn F., Cho K.-H., Yin H., Wolkenhauer O. //Fuzzy Sets and Systems 152. – 2005. – P. 49–66.
44. Гибридные нейро-фаззи модели и мультиагентные технологии в сложных системах /Е.В. Бодянский, В.Е. Кучеренко, Е.И. Кучеренко, А.И. Михалев, В.А. Филатов //Под ред. Е.В. Бодянского. – Днепропетровск: Системные технологии, 2008. – 403 с.
45. Hartigan J.A. Clustering algorithms / J.A. Hartigan. – N.Y.: Wiley, 1975. – 386 p.
46. Гайдышев И. Анализ и обработка данных: специальный справочник / Гайдышев И. – СПб: Питер, 2001. – 752 с.

47. Васильев Н. Метрические пространства / Васильев Н. // Квант. – М.:МНЦМО, 1990. – С. 17–23.
48. Скворцов В.А. Примеры метрических пространств / Скворцов В.А. – М.: МЦНМО, 2002. – 24 с.
49. Карпов Л.Е. Методы добычи данных при построении локальной метрики в системах вывода по прецедентам / Л.Е. Карпов, В.Н. Юдин. – М.: ИСП РАН, препринт № 18, 2006. – 20 с.
50. Александров В.В. Алгоритмы и программы структурного метода обработки данных / Александров В.В., Горский Н.Д. – Л.: Наука, 1983. – 208 с.
51. Браверман Э.М. Структурные методы в обработке эмпирических данных / Браверман Э.М., Мучник И.Б. – М.: Наука, 1983. – 464 с.
52. Жамбю М. Иерахический кластер-анализ и соответствия / Жамбю М. – М.: Финансы и статистика, 1988. – 342 с.
53. Дюран Б. Кластерный анализ / Дюран Б., Оделл П. – М.: Финансы и статистика, 1977. – 176 с.
54. Классификация и кластер. – М.: Мир, 1980. – 392 с.
55. Duda O.R. Pattern classification / Richard O. Duda, Peter E. Hart, David G. Stork. Second edition. Wiley Interscience – 637 p.
56. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. / Рутковская Д., Пилиньский М., Рутковский Л. – М.: Горячая линия–Телеком, 2004. – 452 с.
57. Радивоненко О.С. Принципы интеллектуальной поддержки начальных стадий проектирования авиационных двигателей на основе технологий DATA MINING /РадивоненкоО.С. // Вісник Харківського Національного Університету. - Харків: ХНУ,2002. – Вип. 551. - Ч.1. - С.229-233.
58. Яхъяева Г.Э. Нечеткие множества и нейронные сети / Яхъяева Г.Э. – М.: БИНОМ, 2006. – 316 с.
59. Jang J.-S. R. Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence / Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani. – Prentice-Hall, 1997. – 614 p.
60. Xu B. Automatic color identification in printed fabric images by a fuzzy-neural network / Xu B., Lin Sh. – AATCC Review, 2(9), 2002. –P. 42–45.
61. Джонс М.Т. Программирование искусственного интеллекта в приложениях / Джонс М.Т. – М.: ДМК Пресс, 2004. – 312 с.
62. Halpern J.Y. Reasoning about Uncertainty / Joseph Y. Halpern. – The MIT Press, Cambridge, Massachusetts, London, England, 2003. – 483 p.
63. Вятченин Д.А. Прямые алгоритмы нечеткой кластеризации, основанные на операции транзитивного замыкания и их применение к обнаружению аномальных наблюдений / Вятченин Д.А. // Искусственный интеллект. – 2007. – № 3. – С. 205-216.

64. Соколов А.Ю. Алгебраическое моделирование лингвистических динамических систем // Проблемы управления и информатики / Соколов А.Ю. – 2000. – №2. – С. 141–148.
65. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / Под ред. В.М. Курейчика - 2-е изд., испр. и доп. - М.: ФИЗМАТЛИТ. 2006. - 320 с.
66. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации / Орловский С.А. – М.: Наука. Главная редакция физико-математической литературы, 1981. – 208 с.
67. Corsini P. A system based on a modified version of the FCM algorithm for profiling Web users from access log / Corsini P., Dosso L., Lazzerini B., Marcelloni F // P. 725–729.
68. Заде Л. Основы нового подхода к анализу сложных систем и процессов принятия решений // Математика сегодня. – М.: Знание, 1974. – С. 5–49.
69. Ивахненко А.Г. Моделирование сложных систем (информационный подход) / Ивахненко А.Г. – К.: Вища школа, 1987. – 63 с.
70. Интеллектуальные системы принятия проектных решений / [А.В. Алексеев, А.Н. Борисов, Э.Р. Вилюмс, Н.Н. Слядзь, С.А. Фомин]. – Рига: Зинатне, 1997. – 320 с.
71. Молчанов А.А. Моделирование и проектирование сложных систем / Молчанов А.А. – К.: Выща школа, 1988. – 359 с.э
72. Bellman R.E. Decision making in fuzzy environment / Bellman R.E., Zadeh L.A. // Management Sciense. – 1970. – №17. – P.141–164.
73. Takagi T. Fuzzy identification of systems and its application to modeling and control / Takagi T., Sugeno M. // IEEE Trans. On Syst. Man. Cybern. – 1985. – Vol. 15. – P. 116–132.
74. Земсков В.Н. Сжатие изображений на основе автоматической классификации / Земсков В.Н., Ким И.С. // Известия вузов. Электроника, 2003. – № 2. – С. 50–56.
75. Blake C. UCI repository of machine learning databases / C. Blake and C. J. Merz. – Irvine, CA: University of California, Department of Information and Computer Science, 1998. – [http://www.ics.uci.edu/mlearn/MLRepository.html] .
76. Вятченин Д.А. Нечеткие методы автоматической классификации. – Мн.: УП «Технопринт», 2004. – 219 с.
77. Бочарников В.П. Fuzzy-технология: Математические основы. Практика моделирования в экономике / Бочарников В.П. – СПб: «Наука» РАН, 2001. – 328 с.
78. Data Mining: A heuristic approach / [edited by] Hussein Aly Abbass, Ruhul Amin Sarker, Charles S. Newton. – Idea Group Publishing. Hershey-London-Melbourne-Singapore-Beijing, 2002. – 300 p.

79. Заде Л. Роль мягких вычислений и нечеткой логики в понимании, конструировании и развитии информационных интеллектуальных систем /Заде Л.А. – Новости Искусственного Интеллекта. – №2-3, 2001. – с. 7–11.
80. Заде Л. Понятие лингвистической переменной и ее применение к принятию приближенных решений / Заде Л. – М.: Мир, 1976. – 167 с.
81. Zadeh L.A. The Calculus of Fuzzy If-Then Rules // AI Expert. – 1992. – Vol. 7. – P. 23–27.
82. Konar A. Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain. – CRC Press LLC, Boca Raton United States of America 2000. – 788 p.
83. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / [Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев]. – Х.: Основа, 1997. – 112 с.
84. Kosko B. Fuzzy systems as universal approximators / Kosko B. // IEEE Transactions on Computers,vol. 43, No. 11,November 1994.–P. 1329–1333.
85. Cordon O. A General study on genetic fuzzy systems / Cordon O., Herrera F. // Genetic Algorithms in engineering and computer science, 1995. –- P. 33–57.
86. Дюбуа Д. Теория возможностей. Приложения к представлению знаний в информатике: Пер. с фр. / Дюбуа Д., Прад А. – М.: Радио и связь, 1990. – 228 с.
87. Agrawal R. Fast Algorithms for Mining Association Rules / Rakesh Agrawal and Ramakrishnan Srikant: In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September, 1994. – P. 487–499.
88. Han J. Discovery of multiple-level association rules from large databases / J. Han and Y. Fu: In Proc. of the 21st Int'l Conference on Very Large Databases. – Zurich, Switzerland, September, 1995. – P. 420–431.
89. Прикладная статистика: Классификация и снижение размерности. Справ. изд./ [С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин]. – М.: Финансы и статистика, 1989. – 608 с.
90. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999. — 270 с.
91. Четыркин Е.М. Статистические методы прогнозирования. 2-е изд. М., Статистика, 1977.
92. Brown R.G. Smoothing forecasting and prediction of discrete time series. N.Y., 1963.
93. Brown R.G., Meyer R.F. The fundamental theorem of exponential smoothing. – Oper. Res., 1961, vol. 9 n. 5.
94. Wade R.C. A technique for initializing exponential smoothing forecasts. – Management Science, 1967, vol. 13, n.7.

95. Шумейко А.А., Сотник С.Л. Интеллектуальный анализ данных (Введение в Data Mining). – Днепропетровск: Белая Е.А., 2012. – 212 с.
96. Cohen G.D. A note on exponential smoothing and autocorrelated inputs.- Oper. Res., 1963, vol. 11, n. 3.
97. Cox D.R. Prediction by exponentially weighted moving averages and related methods.- J. of the Royal Stat. Soc., 1961, vol. 23, n. 2.
98. Лукашин Ю.П. Адаптивные методы краткосрочного прогнозирования. – М.: Статистика, 1979. – 254с., ил.- (Мат. Статистика для экономистов)
99. Лоули Д. Факторный анализ как статистический метод / Д. Лоули,А. Максвелл. - М.: Мир,1967. - 144 с.
100. Abonyi J. Fuzzy clustering based segmentation of time-series / J. Abonyi, B. Feil, S.Z. Nemeth, P. Arva // Proc. 5th International Symposium on Intelligent Data Analysis. - Berlin, Germany. - 2003. - P. 275-285.
101. Abonyi J. Cluster analysis for data mining and systems identification / J. Abonyi, B. Feil. - Basel-Boston-Berlin: Birkheueser Verlag AG. - 2007. – 303 p.
102. Hoeppner F. Fuzzy clustering of sampled functions / F. Hoeppner,F. Klawonn // Proc. 19-th Int. Conf. North American Fuzzy Information Processing Society (NAFIPS). - Atlanta, USA, 2000. - P. 251-255.
103. Gorshkov Ye. Robust recursive fuzzy clustering-based segmentation of biological time series / Ye. Gorshkov,I. Kokshenev, Ye. Bodyanskiy, V. Kolodyazhniy and others // Proc. 2006 Int. Symp. on Evolving Fuzzy Systems (EFS'06). Ambleside, Lake District, UK, 7-9 Sep., 2006. - IEEE Press, 2006. –P. 101-105.
104. Connor J.T. Recurrent neural networks and robust time series prediction / J.T. Connor, R.D. Martin, L.E. Atlas // IEEE Trans, on Neural Networks.- 1994. - 5. - P. 240 - 254.
105. Saxen H. Nonlinear time series analysis by neural networks. A case study / H. Saxen // Int. J. Neural Systems. - 1996. - 7. - P. 195-201.
106. Madhavan P.G. A new recurrent neural network learning algorithm for time series prediction / P.G. Madhavan // J. of Intelligent Systems. - 1997. – N 7. - P. 103 - 116.
107. Conway A.J. Delayed time series predictions with neural networks / A.J. Conway, K.P. Macpherson, J.C. Brown //Neurocomputing. -1998. – N 18.- P. 81-89.
108. Бодянський Є.В. Рекурентна прогнозуюча штучна нейронна мережа: архітектура та алгоритми навчання / Є.В. Бодянський, Н.Є. Кулішова, О.Г. Руденко // Адаптивні системи автоматичного управління. - Дніпропетровськ: Системні технологи, 1999. - 2 (22). - С. 129-137.
109. Wan E. Temporal backpropagation for FIR neural networks / E. Wan // Int. Joint Conf. on Neural Networks. - V.l. - San Diego, 1990. - P. 575 - 580.

110. Wan E.A. Time series prediction by using a connectionist network with integral delay lines / E. Wan // Time Series Prediction. Forecasting the Future and Understanding the Past [Eds. by A. Weigend,N. Gershenfeld]. - SFI Studies in the Sciences of Complexity. - V. XVII. - Reading: Addison-Wesley, 1994.- P.195-218.
111. Sandberg I.W. Uniform approximation of multidimensional myopic maps / I.W. Sandberg // IEEE Trans, on Circuits and Systems. - 1997. - 44. - P. 477-485.
112. Back A.D. A unifying view of some training algorithms for multilayer perceptions with FIR filter synapses / A.D. Back,E.A. Wan,S. Lawrence, A.C. Tsoi // Neural Networks for Signal Processing [Eds. by J. Vlontzos, J. Hwang, E. Wilson]. - N.Y.: IEEE Press, 1994. - 4. - P. 146-154.
113. Никифоров И.В. Последовательное обнаружение изменения свойств временных рядов / И.В. Никифоров. - М.: Наука,1983. - 199 с.
114. Isermann R. Process fault detection based modelling and estimating methods -a survey / R. Isermann // Automatica. - 1984. - 20(4). - P. 387-404.
115. Kerestencioglu F. Change detection and input design in dynamical systems / F. Kerestencioglu. - Taunton, UK: Research Studies Press. - 1993. - 152 p.
116. Бодянский Е.В. Искусственные нейронные сети: архитектуры, обучение, применения / Е.В. Бодянский, О.Г. Руденко. - Харьков: ТЕЛЕТЕХ, 2004. -369 с.
117. Omatu S. Neuro-control and its applications / S. Omatu,M. Khalid,R. Yusof. -London: Springer-Verlag, 1995. - 255 p.
118. Badr A.Z. Neural network based adaptive PID controller / A.Z. Badr // Proc. Conf. Control of Industrial Systems ”Control for the Future of the Youth”.- Berfort-France,1997. - V. 1/3. - P. 359 - 365.
119. Бодянский Е.В. Робастный гибридный двойной вэйвлет-нейрон в задачах обработки динамики показателей гомеостаза при остром стресс-повреждении /Е.В. Бодянский, Е.А. Винокурова // Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту»: зб. праць по матеріалам міжнар. наук. конф. - Євпаторія-Херсон. - Т. 3(1). - 2008.- С. 56-59.
120. Sugeno M. An introductory survey of fuzzy control / M. Sugeno // Information Sciences. - 1985. - 36. - P. 59 - 83.
121. Mamdani E.H. Application of fuzzy algorithms for control of simple dynamic plants /E.H. Mamdani // Proc. of IEEE. - 1974. - 121 (12). - P. 1585-1588.
122. Box G.E.P. Time series analysis: forecasting and control / G.E.P. Box, G.M. Jenkins, G.S. Reinsel. - New York: Wiley. - 2008. - 784 p.
123. Винокурова Е.А. Робастные адаптивные нейро-фаззи и вэйвлет-нейро-фаззи системы вычислительного интеллекта / Е.А. Винокурова // Проблеми інформатики та моделювання: зб. наук, праць за

матеріалами 9-оі міжнар. науково-технічної конф. - Х.: НТУ ”ХПГ,2009. - С. 24.

124. Dillon W.R. Multivariate analysis: methods and applications / W.R. Dillon, M. Goldstein. - New York: Wiley. - 1985. - 445 p.
125. Haykin S. Neural networks. A comprehensive foundation / S. Haykin. - Upper Saddle River, NJ: Prentice Hall. - 1999. - 842 p.
126. Bezdek J.C. Editorial-fuzzy models: What are they and why / J.C. Bezdek // IEEE Trans, on Fuzzy Systems. - 1993. - 1. - P. 1-5.
127. Gupta M.M. Fuzzy neural networks: Theory and Applications / M.M. Gupta // Proc. of SPIE. - 1994. - 2353. - P. 303-325.
128. Buckley J.J. Fuzzy neural networks: a survey / J.J. Buckley, Y. Hayashi // Fuzzy Sets and Systems. - 1994. - 66. - P. 1-13.
129. Brown M. Neuro-fuzzy adaptive modelling and control / M. Brown, C. Harris. -New York: Prentice Hall. - 1994. - 528 p.
130. Винокурова Е.А. Гибридные эволюционные адаптивные вэйвлет-нейро-фаззи-системы для динамического интеллектуального анализа данных // Диссертация д-р техн. наук: 05.13.23, Харьковский национальный университет радиоэлектроники. – Х., 2012. – 387 с.
131. Михальов О.І., Водолазський Ю.О. Вейвлет-мультифрактальний аналіз складних зображень //Вісник ВПІ. – Вінницький національний технічний університет. –2009. -№2. – С. 84-87.
132. Быковец В.В., Бойко М.М., Власова Е.Н., Михалёв А.И. Прогнозирование прочности железорудных окатышей на основе нечеткого вывода //Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 3 (74). - Днепропетровск, 2011. - С.164 - 168.
133. Бабенко Ю.В., Михалёв А.И. Исследование и сравнительный анализ классического и хаотического генетических алгоритмов в задачах глобальной оптимизации // Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 3 (74). - Днепропетровск, 2011. - С.138 - 144.
134. Новікова Л.Ю., Михальов О.І. Проектування нечіткої системи впливу складу і якості шихти на показники процесу агломерації // Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 6 (71). - Днепропетровск, 2010. - С.148 - 153.
135. Бабенко Ю.В., Михалев А.И. Исследование модифицированного оператора мутации генетического алгоритма в задачах глобальной оптимизации //Материалы международной конференции «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта» (ISDMCI’2011). - Том 2. – Херсон: ХНТУ, 2011. – С. 178.
136. Новикова Е.Ю., Михалев А.И. Оценка рисков потребительского кредитования с применением методов DATA Mining //Тезисы Международной научно-практической конференции «Современные ин-

формационные технологии на транспорте, промышленности и образовании». – ДНУЖТ-ДИИТ: Днепропетровск, 2011. – С. 69.

137. Михалев А.И., Прядко Н.С., Сухомлин Р. А. Исследование хаотических акустических ритм-сигналов и их вейвлет анализ //Материалы международной конференции «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта» (ISDMCI'2011). - Том 2. – Херсон: ХНТУ, 20011. – С. 233-234.
138. Новікова К.Ю, Михальов О.І. Алгоритм нечіткої кластерізації в задачах аналізу металографічних зображень // Материалы международной конференции «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта» (ISDMCI'2011). - Том 2. – Херсон: ХНТУ, 2011. – С. 236-237.
139. Михалев А.И., Прядко Н.С., Сухомлин Р.А. Вейвлет-анализ акустических сигналов процесса струйного измельчения // Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 3 (80). - Днепропетровск, 2012. - С.122 - 127.
140. Новікова К.Ю., Михальов О.І. Дослідження алгоритмів нечіткої кластеризації в задачах аналізу металографічних зображень // Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 4 (81). - Днепропетровск, 2012. - С.110 –119.
141. Бабенко Ю.В., Михалев А.И. Модифицированный генетический алгоритм для оптимизации многоэкстремальных функций /Тези доповідей III Всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління (САІУ-2012)». – Запоріжжя, 14-16.03.2012. – Запоріжжя: КПУ, 2012. – С. 20-21.
142. Оптимизация параметров ферросплавного производства с использованием методов нечеткого вывода // Михалев А.И, Лысая Н.В., Лысый Д.А., Гладких В.А., Лысенко В.Ф. – Днепропетровск: Системные технологии, 2008. – 130 с.
143. Бодянський Є.В., Кучеренко Є.І., Михальов О.І., Філатов В.О., Гасик М.М., Куцин В.С. Методи обчислювального інтелекту в системах керування технологічними процесами феросплавного виробництва/ Монографія (Наукове видання). - Дніпропетровськ: Національна металургійна академія України, 2011. - 420 с.
144. Dorigo M. Optimization, Learning and Natural Algorithms / M. Dorigo // PhD thesis .− Italy : Politecnico di Milano, 1992.
145. Mendel J.M. Advances in type-2 fuzzy sets and systems / J.M. Mendel // Information Sciences. - 2007. - 177. - P. 84-110.
146. Гибридные нейро-фаззи модели и мультиагентные технологии в сложных системах /Е.В. Бодянский, В.Е. Кучеренко, Е.И. Кучеренко, А.И. Михалев, В.А. Филатов //Под ред. Е.В. Бодянского. – Днепропетровск: Системные технологии, 2008. – 403 с.

147. Гаряев П.П. Волновой генетический код, М: ИЗДАТЦЕНТР, 1997. 108 стр.
148. Holland J. H. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975
149. Водолазский Ю.А., Михалев А.И., Клименко А.П. Мультифрактальный анализ микрошлифов колёсных сталей // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 6(65). – Днепропетровск, 2009. – С. 39 - 45.
150. Михайловская Т.В., Тутык В.А., Михалев А.И. Расчет технологических параметров обработки электронным пучком металлической поверхности с использованием клеточно-автоматной модели // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 5(65). – Днепропетровск, 2010. – С.75 - 81.
151. Михайловская Т.В., Михалев А.И., Гуда А.И., Новикова Е.Ю. Моделирование движения пассажиропотока с использованием клеточно-автоматного похода //Автомобільний транспорт. Вісник Харківського національного автодорожнього університету. - Вып. 25, 2009. - С. 250-253.
152. Новикова Е.Ю., Михалев А.И., Михайловская Т.В. Нечеткие алгоритмы исследования свойств электросталей для транспортных систем // Автомобільний транспорт. Вісник Харківського національного автодорожнього університету. - Вып. 25, 2009. - С. 246-249.
153. Михалев А.И., Гуда А.И., Деревянко А.И. Критерии идентификации параметров хаотической динамики управляемого объекта //Автомобільний транспорт. Вісник Харківського національного автодорожнього університету. - Вып. 25, 2009. – С. 254-257.
154. Михальов О.І., Гуда А.І., Дмитрієва І.С. Особливості моделювання та ідентифікації хаотичної системи Ресслера зі збуреннями. // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 2(67). – Днепропетровск, 2010. – С.114 – 118.
155. Михальов О.І., Каліберда Ю.О. Побудова IDS на основі штучної імунної мережі // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 3(68). – Днепропетровск, 2010. – С.106-110.
156. Горбонос А.А., Михалёв А.И. Модель Пенроуза как основа для построения фрактальных поверхностей квазикристаллов // Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 3(68). – Днепропетровск, 2010. – С.163 -168.
157. Михалёв А.И., Сухомлин Р.А. Оценивание хаотических ритм-сигналов в задачах диагностики динамических систем //Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 3 (74). - Днепропетровск, 2011. - С.145 - 151.

158. Новикова Е.Ю., Михайловская Т.В., Михалёв А.И. Fuzzy – идентификация теплофизических параметров при моделировании затвердевания // Системные технологии. Региональный межвузовский сборник научных работ. - Выпуск 4 (75). - Днепропетровск, 2011. - С.170 - 174.
159. Новикова Е.Ю., Михалев А.И. Оценка рисков потребительского кредитования с применением методов DATA Mining //Тезисы Международной научно-практической конференции «Современные информационные технологии на транспорте, промышленности и образовании». – (12.05.-13.05.2011). – ДНУЖТ-ДИИТ: Днепропетровск, 2011. – С. 69.
160. Михалев А.И., Прядко Н.С., Сухомлин Р. А. Исследование хаотических акустических ритм-сигналов и их вейвлет анализ //Материалы международной конференции «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта» (ISDMCI'2011). - Том 2. – Херсон: ХНТУ, 20011. – С. 233-234.
161. Михальов О.І., Крамаренко В.В., Ялової К.М., Новікова К.Ю. Структури даних та алгоритми: Навч. посібник з гріфом МОНУ. — Дніпродзержинськ, 2010. — 263 с.
162. Михалев А.И., Стовпченко И.В. Mimo-каскадное нео-фаззи моделирование процесса выплавки стали в кислородном конвертере // Тези доповідей міжнародної конференції з проблем використання інформаційних технологій в освіті, науці та промисловості. – Дніпропетровськ: вид.-во ДВНЗ Національний гірничий університет, 19-21 вересня 2011р. – С. 15-17.
163. Недоспасов А.А., Михалев А.И. Оценка степени самоподобия Интернет – трафика методами мультифрактального анализа //Системные технологии. Региональный межвузовский сборник научных работ. – Выпуск 2 (79). – Днепропетровск, 2012. – С. 111-117.
164. Нейросетевое моделирование и прогнозирование характеристик процесса выплавки ферросиликомарганца // А.И. Михалев, В.А. Гладких, Н.В. Лысая, Д.А. Лысый, В.Ф.Лысенко, А.И. Деревянко. - Нові технології. - Науковий вісник КУЕІТУ, № 2(24), 2009. - С.14-17.
165. Линник Ю.В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений. М.: ФИЗМАТЛИТ, 1958, 336 с.

Навчальне видання

Михальов Олександр Ілліч

Винокурова Олена Анатоліївна

Сотник Сергій Леонідович

КОМП'ЮТЕРНІ МЕТОДИ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ДАНИХ

Російською мовою

Формат 60х84 1/16 Ум. друк арк. 13,2. Тираж - 300.

Замовлення №2/14

ПП Усик Т.Л.

49000 м. Дніпропетровськ, пр.. Фестивальний, 24/85

Свідоцтво про реєстрацію ВО № 163448 від 15.08.2002

ISBN 978-966-2596-14-4