Государственное образовательное учреждение  
высшего образования Луганской Народной Республики  
«Донбасский государственный технический институт»

Факультет фундаментального и инженерного образования и инноваций

Кафедра управления инновациями в промышленности

УТВЕРЖДАЮ

Первый проректор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В. Кунченко

**Методические указания для проведения лабораторных работ по ДИСЦИПЛИНе**

«Моделирование систем и процессов*»*

*(шифр, наименование дисциплины)*

15.03.04 "Автоматизация технологических процессов и производств"

(код, наименование направления)

Профиль подготовки: «Управление и инновации в автоматизированных системах и технологических процессах»

(профиль подготовки, магистерская программа)

Квалификация \_\_\_\_\_бакалавр\_\_\_\_\_\_

(бакалавр/специалист/магистр)

Форма обучения \_\_\_\_очная /заочная\_\_\_\_

(очная/заочная)

Алчевск, 2022

УДК 62-52

В практикуме по математическому пакету Маткад в систематическом виде изложено решение многих задач в пакете Маткад. В модуле разобрано построение в Маткаде ряда математических моделей, а также приведено решение ряда математических задач..

.

Содержание Стр.

[Введение 4](#_TOC_250009)

[Лабораторная работа № 1 Программирование в Маткаде 4](#_TOC_250008)

[Лабораторная работа № 2.Решение задач методом Монте-Карло 7](#_TOC_250007)

[Лабораторная работа № 3.Задачи линейного программирования……………….. 12](#_TOC_250006)

[Лабораторная работа № 4.Оптимизация качества молочной колбасы………….. 17](#_TOC_250005)

[Лабораторная работа № 5.Выбор оптимальной траектории……………………… 19](#_TOC_250004)

Лабораторная работа № 6.Оптимальное созревание сырокопченой колбасы…… 24

[Лабораторная работа № 7.Оптимизация работы предприятия…………………… 28](#_TOC_250003)

Лабораторная работа № 8.Оценка качества пищевых продуктов……………… 31

[Лабораторная работа № 9.Оценка качества конфет «Зефир»… 35](#_TOC_250002)

[Лабораторная работа № 10.Нечеткие множества…………………………………. 39](#_TOC_250001)

Лабораторная работа № 11.Определение оптимального количества жира и белков 41

в молочной колбасе…………………………………………………………………

Лабораторная работа № 12.Распознавание образов по минимуму расстоя- 44

ния…………………………………………………………………………………….

Лабораторная работа № 13.Распознавание образов по коэффициенту корреляции 47 Лабораторная работа № 14.Распознавание образов по нечеткой мере сходства… 50 Лабораторная работа № 15.Нейронные сети. Персептрон……………………. 52

[Лабораторная работа № 16.Обучение однослойных нейронных сетей 55](#_TOC_250000)

Лабораторная работа № 18.Планирование экспериментов……………………. 58

Лабораторная работа № 19.Адаптивный фильтр Калмана……………………….. 62

Лабораторная работа № 21.Подбор ПИД регулятора…………………………….. 65

Литература…………………………………………………………………………. 68

ВВЕДЕНИЕ

Лабораторный практикум состоит из двадцати одной лабораторной рабо- ты. Большинство из них посвящено решению конкретных прикладных задач. Но так как предыдущий модуль №2 посвящен только работе со встроенными функциями, а некоторые абстрактные задачи требуют составления программы, они также помещены в этом модуле.

Первая лабораторная работа посвящена составлению программ в Маткаде. Затем рас- смотрено решение различных задач методом Монте-Карло. Группа лабораторных работ по- священа методам оптимизации. Здесь рассматриваются задачи линейного программирова- ния, задачи многошаговой оптимизации дискретным методом динамического программиро- вания для детерминированных и стохастических систем.Несколько лабораторных работ по- священо оценке качества пищевых продуктов путем обработки экспериментальных данных. При этом используется аппарат нечетких множеств, и многомерная множественная регрес- сия (встроенная функция Minerr).

В остальных лабораторных работах рассматриваются задачи распознавания образов с помощью различных мер сходства, нейронные сети, планирование экспериментов, и две за- дачи по методам управления динамическими системами.

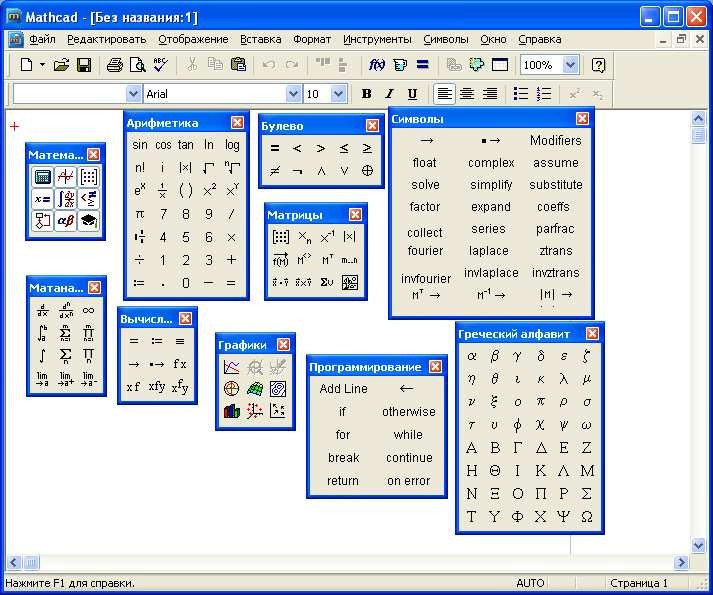
Так как в стандартных курсах вузовской математики не изучают некоторые разделы, задачи на которые рассмотрены в данном модуле, то во многих лабораторных работах перед решением задач кратко изложен теоретический материал.

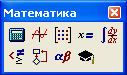
ЛАБОРАТОРНАЯ РАБОТА №1. ОСНОВЫ РАБОТЫ С MATHCAD. Программирование

**Общие сведения о маткаде**

Основное окно приложения имеет ту же структуру, что и большинство приложений Windows. Сверху вниз располагаются заголовок окна, строка меню, панели инструментов (стандартная и форматирования) и рабочий лист, или рабочая область, документа. Новый документ создается автоматически при запуске MathCAD. Файлы документов в MathCAD имеют расширение **.mcd**.

Большинство команд можно выполнить как с помощью меню (верхнего или контекстного), так и панелей инструментов или клавиатуры.



Панель Math (Математика) предназначена для вызова на экран еще девяти панелей, с помощью которых происходит вставка математических операций в документы. Чтобы

вызвать какую-либо из них, нужно нажать соответствующую кнопку на панели Математика.

В окне редактирования формируется документ MathCAD. Новый документ получает имя Untitled (Без названия) и порядковый номер. Одновременно открыто может быть до восьми документов.

Документ состоит из трех видов областей: формульных, текстовых и графических. Расположение нетекстовых блоков в документе имеет принципиальное значение. Области просматриваются системой, интерпретируются и исполняются. *Просмотр идет слева направо и сверху вниз. Для ввода текстового комментария* нужно выполнить команду Text Region (Текстовая область) из пункта меню Insert или нажать клавишу с двойной кавычкой (“), или нажать на кнопку текста на панели инструментов. Текстовая

область служит для размещения текста между формулами и графиками.

При этом в месте ввода появляется курсор в виде вертикального штриха, на место которого вводятся символы текста. Внутри текста курсор перемещается клавишами перемещения курсора. Переход на новую строку производится нажатием на клавишу Enter. Для окончания ввода нужно щелкнуть мышью вне текстовой области.

*Для ввода формулы* нужно установить указатель мыши в свободном месте окна редактирования и щелкнуть левой кнопкой мыши. Появится визир в виде красного крестика. Он указывает место, с которого начинается набор формулы.

### Константы и переменные

Константами называются поименованные объекты, хранящие некоторые значения, которые не могут быть изменены.

В MathCAD применяются десятичные, восьмеричные и шестнадцатеричные числовые константы. Десятичные константы могут быть целочисленными, вещественными, заданными с фиксированной точкой, и вещественными, заданными в виде мантиссы и порядка.

В MathCAD содержится особый вид констант - размерные. Помимо своего числового значения они характеризуются еще и указанием на то, к какой физической величине они относятся. Для этого указания используется символ умножения. В системе MathCAD заданы следующие основные типы физических величин: time (время), length (длина), mass (масса) и charge (заряд). При необходимости их можно изменить на другие.

Переменные являются поименованными объектами, которым присвоено некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д. Имена констант, переменных и иных объектов называют идентификаторами.

Имя переменной называется идентификатором. MathCAD различает в идентификаторах символы верхнего и нижнего регистров. Например: ABC и AbC имена разных переменных.

Идентификаторы MathCAD должны начинаться с буквы и могут содержать следующие символы:

латинские буквы любого регистра;

* арабские цифры от 0 до 9;
* символ подчеркивания (\_), символ процент (%) и символ (.);
* буквы греческого алфавита (набираются с использованием клавиши Ctrl или применяется палитра греческих букв).

### Определение переменных

Переменные должны быть предварительно определены пользователем, т. е. им необходимо хотя бы однажды присвоить значение. В качестве оператора присваивания используется знак :=, тогда как знак = отведен для вывода значения константы или переменной. Попытка использовать неопределенную переменную ведет к выводу сообщения об ошибке.

В MathCAD различают: локальные и глобальные переменные. Локальные переменные вводятся:

Имя\_переменной : выражение

На экране:

Имя\_переменной := выражение

Глобальные переменные вводятся:

Имя\_переменной ~ выражение

На экране:

Имя\_переменной  выражение

Если переменной присваивается начальное значение с помощью оператора

:=, такое присваивание называется локальным. До этого присваивания переменная не определена и ее нельзя использовать. MathCAD читает рабочий документ слева направо и сверху вниз, поэтому определив переменную, ее можно использовать в вычислениях везде правее и ниже равенства, в котором она определена. Однако с помощью знака ≡ (три горизонтальные черточки) можно обеспечить глобальное присваивание, т. е. оно может производиться в любом месте документа. К примеру, если переменной присвоено таким образом значение в самом конце документа, то она будет иметь это же значение и в начале документа.

Например:

Ввод с клавиатуры Вид на экране

local:137 local := 137 локальное определение переменной local; global~987.23 global ≡ 987.23 глобальное определение переменной

global.

Переменные могут использоваться в математических выражениях, быть аргументами функций или операндом операторов.

Переменные могут быть и размерными, т. е. характеризоваться не только своим значением, но и указанием физической величины, значение которой они хранят. Проведение расчетов с размерными величинами и переменными особенно удобно при решении различных физических задач.

### Предопределенные переменные

Предопределенные (системные) переменные – особые переменные, которым изначально системой присвоены начальные значения.

|  |  |  |  |
| --- | --- | --- | --- |
| Переменная | Ввод | Назначение | Значение  по умолчанию |
|  | Ctrl + Shift + p | Число  | 3.14159 |
| e | e | Основание натурального  логарифма | 2.718 |
|  | Ctrl + Shift + z | Системная бесконечность | 10307 |
| i или j | 1i или 1j | Мнимая единица |  |
| % |  | Процент | 0.01 |
| TOL |  | Погрешность численных  методов | 0.001 |
| ORIGIN |  | Нижняя граница  индексации массивов | 0 |

### Операторы

*Операторы* - элементы языка, с помощью которых можно создавать математические выражения. К ним, например, относятся символы арифметических и логических операций, знаки вычисления сумм, произведений, производной и интеграла и т. д.

Операторы, обозначающие основные арифметические действия, вводятся с панели Calculator (Калькулятор, Арифметика).

Вычислительные операторы вставляются в документы при помощи панели инструментов Calculus (Матанализ). При нажатии любой из кнопок в документе появляется символ соответствующего математического действия, снабженный несколькими местозаполнителями. Количество и расположение местозаполнителей определяется типом оператора и в точности соответствует их общепринятой математической записи.

Результатом действия логических, или булевых, операторов являются только числа 1 (если логическое выражение, записанное с их помощью, истинно) или 0 (если логическое выражение ложно).

Вычислительные операторы сгруппированы на панели Evaluation (Вычисления):

* Численный вывод (Evaluate Numerically) =
* Символьный (аналитический) вывод (Evaluate Symbolically) 
* Присваивание (Definition) :=
* Глобальное присваивание (Global Definition) .

|  |  |  |
| --- | --- | --- |
| Оператор | Клавиша | Назначение оператора |
| *X := Y* | *X : Y* | Локальное присваивание *X* значения *Y* |
| *X* *Y* | *X*  *Y* | Глобальное присваивание *X* значения *Y* |
| *X =* | *X =* | Вывод значения *X* |
| *X + Y* | *X + Y* | Сложение *X* с *Y* |
| *X - Y* | *X - Y* | Вычитание из *X* значения *Y* |
| *X Y* | *X \* Y* | Умножение *X* на *Y* |
| *X*  *Z* | *X / Z* | Деление *X* на *Z* |
| *X*  *Y* | *Ctrl + /* | Линейное деление |
| *a b*  *c* | *Ctrl + Shift + +* | Дробь (смешанный номер) |
| *zw* | *z ^ w* | Возведение *z* в степень *w* |
| *z* | *z \* | Вычисление квадратного корня из *z* |
| *n!* | *n !* | Вычисление факториала |
| *Bn* | *B [ n* | Ввод нижнего индекса *n* |
| *An,m* | *A [ n , m* | Ввод двойного нижнего индекса |
| *A<n>* | *A Ctrl + 6 n* | Ввод верхнего индекса (для векторов) |

### Ранжированные (дискретные) переменные

Ранжированная переменная – переменная, которая принимает ряд значений при каждом ее использовании.

Для определения ранжированной переменной общего вида используется выражение:

Имя\_переменной := начальное\_значение, начальное\_значение + шаг .. конечное\_значение.

Если шаг равен 1, тогда ранжированную переменную можно задавать следующим образом:

Имя\_переменной := начальное\_значение.. конечное\_значение.

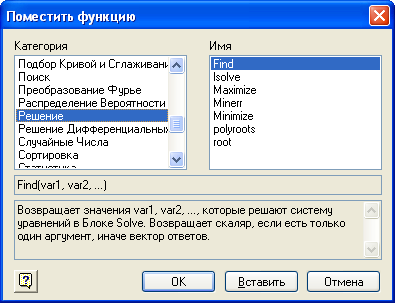
Любое выражение с ранжированными переменными после знака равенства (=) создает таблицу вывода.

### Определение функций

Функция – выражение, согласно которому проводятся некоторые вычисления с его аргументами и определяется его числовое значение.

Функции в пакете MathCAD могут быть встроенные и определенные пользователем.

В MathCAD имеется множество встроенных функций. Для их ввода используется команда меню Вставка  Функция или кнопка на панели инструментов . В диалоговом окне нужно выбрать Категорию и соответствующую функцию.



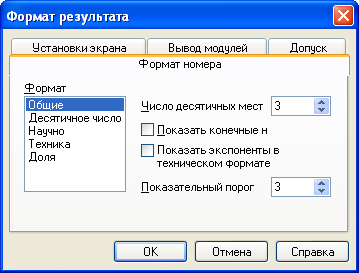
*Функция пользователя* вначале должна быть определена, а затем к ней может быть произведено обращение. Функция пользователя определяется следующим образом:

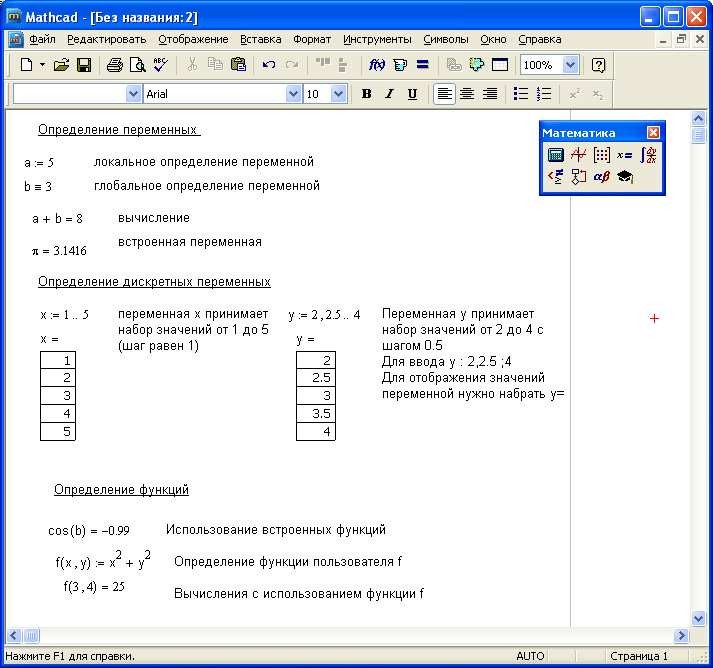
Имя\_функции(Переменная1, Переменная2, …) := Выражение

Задается имя функции, в скобках указывается список аргументов функции - это перечень используемых в выражении переменных, разделяемых запятыми. Затем записывается знак присваивания, справа от которого записывается выражение. Выражение - это любое арифметическое выражение, содержащее доступные системе операторы и функции с операндами и аргументами, указанными в списке аргументов.

Примеры задания функций одной и двух переменных:

f(x):= 10 - exp(x) mult(x, у) := x\*y

Обращение к функции осуществляется по ее имени с подстановкой на место аргументов констант, переменных, определенных до обращения к функции, и выражений. Например:

f(3), sin(1), mult(2,3).

### Форматирование результатов

Способ, которым MathCAD выводит числа, называется форматом результата. Формат результата может быть установлен для всего документа (глобальный формат) или для отдельного результата (локальный формат).

Глобальный формат устанавливается командой меню ФорматРезультат. В диалоговом окне, появляющемся после выбора этой команды, устанавливается выводимая точность числа, диапазон показателя степени (если вывод чисел нужен в форме с плавающей запятой) и точность

нуля. После внесения требуемых изменений нужно нажать кнопку ОК.

Для установки формата отдельного числа нужно: щелкнуть мышью на выражении, результат которого нужно переформатировать; вызвать команду форматирования и проделать вышеописанные действия.

### Построение графиков

Для построения графика используется команда меню ВставкаГрафики. Для создания декартового графика:

1. Установить визир в пустом месте рабочего документа;
2. Выбрать команду Вставка  График  Х-У график, или нажать комбинацию клавиш Shift + @, или щелкнуть кнопку  панели Графики. Появится шаблон декартового графика;
3. Введите в средней метке под осью Х первую независимую переменную, через запятую – вторую и так до 10, например: х1, х2, …;
4. Введите в средней метке слева от вертикальной оси Y первую независимую переменную, через запятую – вторую и т. д., например: у1(х1), у2(х2), …, или соответствующие выражения;
5. Щелкните за пределами области графика, чтобы начать его построение.

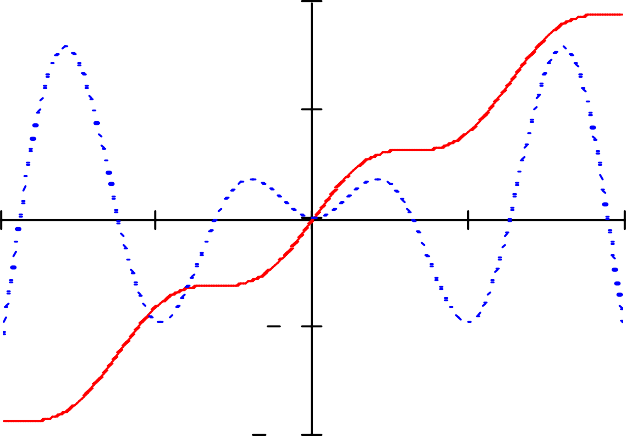
Можно построить несколько зависимостей на одном графике. Для этого нужно ввести соответствующие функции у вертикальной оси (оси ординат). Чтобы разделить описания функций, вводить их нужно через запятую.

Пример. Фрагмент рабочего документа MathCAD

Зададим функцию пользователя f

f(x)  x sin (x)

10



5

f( x) x sin( x)

10 5

0 5 10

5

10

x

ПРОГРАММИРОВАНИЕ В МАТКАДЕ.

В Маткаде имеется встроенный язык программирования. Это язык более высокого уровня, чем Бейсик и Паскаль, он позволяет производить **объектно-ориентированные** про- граммы.

1. При программировании в Маткаде в программе участвуют **локальные переменные**, действие которых распространяется только на программу, а не на весь документ в целом, частью которого является программа. Переменные, дей- ствующие во всем документе, называются **глобальными**.

Для составления программ необходимо, прежде всего, вызвать **панель программиро- вания.** Для этого следует нажать на математической панели кнопку с изображением стре- лок , прямоугольника и ромба между кнопками интегрирования и греческого алфавита.

Появится новая панель – **панель программирования**, состоящая из кнопок:

Кнопка **ADD LINE- ДОБАВЬ СТРОКУ.** При ее нажатии возникает вертикальная линия, объединяющая два оператора в блок с одним входом и одним выходом. Для объеди- нения большего числа операторов кнопку следует нажимать несколько раз.

Кнопка

- это оператор присвоения, например

A  B

Локальной переменной А присваивается значение В.

Кнопка **IF** аналогична оператору условного перехода в языках Бейсик

и Паскаль, например, выражение

A  B

if C  0

означает , что , если С > 0, то A присваивается значение B.

Кнопка **OTHERWISE** дает возможность сделать выбор (аналог ELSE в Бейсике и Паскале).

C  D if A  B

E  F

otherwise

Если A>B, то С присваивается значение D, в противном случае E

присваивается значение F.

Кнопка **FOR** вводит в программу цикл с параметром (когда заранее известно, сколько циклов необходимо выполнить).Количество циклов задается несколькими способа- ми:

*FORA*  5,4,7,8

*FORi* 1..10

*FORA* *V*

Кнопка **WHILE** - образует заголовок цикла с предусловием. Такой цикл использует- ся, если мы заранее не знаем, сколько циклов нам необходимо сделать для решения задачи

(Аналогичные операторы имеются в Бейсике - оператор WHILE - WEND и в Паскале -

оператор WHILE - DO).

В Маткаде набирается:

WHILE < логическое условие> < операторы, которые должны выполняться>. Ниже приводятся элементы программ в Маткаде. Их необходимо прогнать.

**Пример 1.** Задано значение х. В зависимости от этого значения z принимает значение 0

или 3. При изменении х меняется Z

x  1

z 

###### 0 if

x  0

###### 3 otherwise z  3

**Пример2.** Задано значение х. Значение у по-прежнему зависит от х, но вариантов здесь уже три.

x 11

y x if x2 if

x 0

0 x 10

ex otherwise

y  5.987 104

**Пример 3.** Задано найти сумму первых десяти натуральных чисел. До начала следует присвоить сумме s нулевое значение. Так как число циклов известно, используем оператор FOR.

s s 0

for x 1 10

s s x

s  55

В процессе решения примера

1. измените наибольшее значение x до 100 **,**
2. Суммируйте квадраты x

**Пример 4.** Сумма составляется в зависимости от величины x.

s s 0

for x 1 10

s s x if x 5

s s sin(x) if 5 x 8

s s cos ( x) otherwise

s  9.161

**Пример 5.** Применение оператора WHILE. Суммировать натуральный ряд следует до тех пор, пока сумма не превысит число 30. Здесь используются две линии ADD LINE. Преж- де всего устанавливаются начальные значения s и x. Затем пишется оператор WHILE и во втором цикле - само накапливание. Нижняя буква s показывает, по какой переменной произ- водится операция.

Порядок вычислений следующий:

1. Устанавливаются начальные значения s и x.
2. Осуществляется проверка условия. Так как оно выполняется, производится операция

s=s + x = 0+1.

1. Проверяется условие s<30
2. Производится вычисление s.

И так далее. То есть проверка производится ДО вычислений. Поэтому результат пре- вышает заданный. Проверка то проводилась ДО вычисления, тогда было s<30, а после вы- числения стало s>30 на очередной x.

Оператор WHILE проверяет условие ( s<=30) **ДО ОЧЕРЕДНОГО ЦИКЛА.**

s s 0

x 1

while

s 30

s s x

x x 1

s

s  36

После решения заданного примера измените предельное значение s на 15,20, 50.

Составить программы для решения следующих задач:

**Задача 1**. Найти сумму 25 натуральных чисел

S=1+2+3+4+. +25

**Задача 2**. Найти сумму 25 членов числового ряда

S=1-2+4-8+16-32+..........

ПОДСКАЗКА. Здесь каждый следующий член ряда равен предыдущему, умноженному на -2. Un+1= U n(-2).

**Задача 3.** Суммировать 25 членов ряда

S= (3+4)/2 +(6+3)/4 +(12+2)/6 + (24+1)/8 + ............

ПОДСКАЗКА. Здесь следует представить общий член ряда в виде (a+ b )/c и опреде- лить закономерности изменения каждой составляющей.

**Задача 4**. Как известно, индийский владетель расплатился с изобретателем шахмат следующим образом: на первую клетку шахматного поля было положено одно зерно, на вто-

рую - два, на третью - четыре (22 ), на четвертую - восемь (23 ) и т. д. На последнюю, 64- ую клетку было положено 263 зерен. Сколько зерна получил изобретатель шахмат, если одно зерно весит 0,3 г.?

Все вышеприведенные задачи имели в ответе скаляр. Маткад позволяет получать ответ в виде вектора и матрицы. В задаче 7 ответы получаются в виде вектора.

**Задача 7**. Составить циклическую программу заполнения нижеприведенного v вектора числами :А) v=(1,2,3) , В) v=(3,2,1) , С) v=( 1.4.9).Ниже приведено решение варианта А.

vektor 

ORIGIN  1

 0 

v   0 

 0 

ORIGIN:=1 означает, что счет начинается с 1, а не с 0.

for

i  1  3

vi  i

 v1 

 1 

###### 

vektor   v2 

vektor

  2 

 3 

 v3 



vektor

Здесь проведено различие между ГЛОБАЛЬНЫМИ и ЛОКАЛЬНЫМИ переменными. Программа в МАТКАДЕ является обычно частью большой задачи, переменные кото-

рой называются ГЛОБАЛЬНЫМИ.

Переменные внутри программы называются ЛОКАЛЬНЫМИ. Иногда они могут сов- падать.

В данной задаче VEKTOR - глобальная , а V, I - локальные переменные.

В программе приведена связь между ними.Определено начальное значение вектора V.

Слово VEKTOR в нижней части программы определяет, по какой переменной проис- ходит вычисление.

ЛАБОРАТОРНАЯ РАБОТА №2. РЕШЕНИЕ ЗАДАЧ МЕТОДОМ МОНТЕ- КАРЛО

Во многих задачах исходные данные носят случайный характер, поэтому для их ре- шения должен применяться статистико-вероятностный подход. На основе таких подходов построен ряд численных методов, которые учитывают случайный характер вычисляемых или измеряемых величин. К ним принадлежит и метод статистических испытаний, называемый также методом Монте-Карло, который применяется к решению некоторых задач вычисли- тельной математики, в том числе и для вычисления интегралов.-

Метод Монте-Карло состоит в том, что рассматривается некоторая случайная величина ξ , математическое ожидание которой равно искомой величине х:

*Mξ=x*

Проводится серия n независимых испытаний, в результате которых получается (гене- рируется) последовательность n случайных чисел ξ1, ξ2,….., ξn, и по совокупности этих зна- чений приближенно определяется искомая величина

*ξ = (ξ*1 *,+ξ2 + ξn )* / *n*  *x*

1 *n* 1 *n nx*

*Mξ = M( n*  *ξi ) = n M*  *ξi = n = x*

*i=*1 *i=*1

Интегрирование одномерных интегралов.

Интегрирование одномерных определенных интегралов методом Монте-Карло произ- водится по написанному ниже алгоритму.

*b*

*I =* 

*a*

*N*

*f(x)dx= (b*  *a)* / *N*  *f(xi )*

*i=*1

, (1)

где xi - равномерно распределенная случайная величина, f(x) - подынтегральная функция, N - количество - количество случайных аргументов xi, b и a -.верхний и нижний пределы интегрирования. В Маткаде имеется встроенная функция **rnd(x**), возвращающая равномерно распределенную случайную величину в диапазоне 0 - x. Для решения задач ме- тодом Монте-Карло необходимо составить программу с использованием этой функции.

**Пример1**. Вычислить методом Монте-Карло приведенный ниже интеграл. Приведено решение методом Монте-Карло и численным методом

2

5 *x*3 *+* 2x *+*8

3 *x*5 *+* 3x4 *+*8x *+* 9



0

w 

y  0

N  1000000

for

i 1  N

x  rnd(2)

2

 5 x3  2x  8



 dx  1.142

5 x3  2x  8

y  y 

3

x5  3x4  8x  9

 3

 x5



0

 3x4  8x  9

y2

w  N

w  1.142

**Пример 2**. Вычислить методом Монте-Карло интеграл

10



y  

 3

x2  3x  8 dx

Задача отличается от предыдущей тем, что нижний предел не равен нулю, а функция

rnd(х) вычисляет случайные числа в пределах 0-х. Поэтому здесь вычисляются два интеграла

10





 3

x2  3x  8

dx  582.833

m  1000000

0 10

*y =* *(x*2 *+*3x *+*8 *)dx+* *(x*2 *+*3x *+*8 *)dx*

3 0

y1  y2  0

y3  0

for i 0  m

x1  rnd(10)

y2  y2  x12  3x1  8 x2  rnd(3)

y3  y3  x22  3x2  8

i

y2  10y2

m

y3  3y3

m

y1  y2  y3 y1

y  582.833

**Задача 2.** Вычислить методом Монте-Карло и с помощью встроенных функций Маткада нижеприведенные интегралы. Записать ошибку :

3

x3 dx

0

10

sin*(x)ex*  2 *x*2 *+*1*dx =*

3

Интегрирование многомерных интегралов.

Многомерные интегралы

*b*1 *b*2 *bm*

  ........ 

*f* (*x*1, *x*2,........*xm* )*dx*1*dx*2 *dxm*

*a*1 *a*2 *am*

вычисляются методом Монте - Карло по алгоритму

*(b*1  *a*1 *)(b*2  *a*2 *)*...*(bm*  *am ) n*

*n*

 *f(x*1 *, x2 , xm )*

*i=*1

где хi-случайная равномерно распределенная величина, - *f(x*1 *, x2 ,*...*xm )* подынтегральная функция, bi, ai (i=1,2,…n)- верхний и нижний пределы ин- тегрирования.

Многомерные интегралы вычисляются в Маткаде методом Монте-Карло с помощью встроенной функции runif(m,a,b), возвращающей вектор из m равномерно распределенных случайных чисел в пределах от a до b.

**Пример3** Вычислить методом Монте-Карло и обычным численным интеграл

2 2 2

*(x1+ x2+ x3)dx*1*dx*2*dx*3

Данный интеграл имеет 0 0 0 одинаковые пределы, что

облегчает задачу.

w  y  0

for i  1 N

2 2 2

v  runif(3 0 2)

y  y  v0  v1  v2 i

w  22 2y

N

  

  

0 0 0

(x1  x2  x3) dx1dx2dx3  24

w w  24.093

**Пример 4.** Вычислить методом Монте-Карло и обычным численным методом инте-

грал

5 8

*y=*∫ ∫

*x1x2dx* 1 *dx* 2

5 8

1

w 

2

y  0

N  10000

  x1x2dx1dx2  360

 

1 2

for

i  1 N

v  runif(2 0 1) x1  4 v0  1

x2  6 v1  2

y  y  (x1x2) i

w  360.177

w  46y

N

w

Так как интегралы имеют различные пределы, мы формируем

функцией runif вектор v с диапазоном вычислений от нуля до единицы, а затем преобразуем составляющие этого вектора, так, чтобы выдерживались заданные пределы.

Для многомерных интегралов большой размерности вычисление методом Монте-Карло происходит значительно быстрее, чем при использовании встроенной функции интегрирова- ния. Так, интеграл

2 10 8 7 9 8 5 8 9 7

*y=*∫ ∫ ∫ ∫ ∫ ∫ ∫ ∫ ∫ ∫

1 5 2 3 4 1 0 4 6 5

*x1x2x3x4x5x6x7x8x9x*10 *dx* 1 *dx* 2 *dx* 3 *dx* 4 *dx* 5 *dx* 6 *dx* 7 *dx* 8 *dx* 9 *dx* 10

вычисляется на компьютере с тактовой частотой 2 гигагерца встроенной функцией ин- тегрирования более получаса, а методом Монте-Карло- при N=100000 менее двух минут.

**Задача 3.** Вычислить нижеприведенные интегралы с помощью встроенных функций Маткада и методом Монте-Карло. Сравнить результаты

5 9 200 6

*y =*   

0 3 100 4

5 9 3

*y =*  *x12* *(x2+ x3)*3 *dx*1*dx*2*dx*3

*x12 + x23*  *x35*

1 3 0

Поиск глобального экстремума функции в заданной области методом Монте –

**Карло.**

Нелинейные функции в большинстве случаев имеют не один, а несколько экстремумов. Самый большой максимум или минимум называется глобальным экстремумом, остальные – локальными. При решении задачи , как правило, необходимо определение глобального экс- тремума. Ниже дается решение такой задачи в Маткаде для заранее заданной области. Вне нее глобальный экстремум может быть другим.

Пусть задана многоэкстремальная функция

y(x)  x   2x  sin(15  x)

e

Рассмотрим ее графики при различных пределах изменения аргумента x:

Из рис.1 видно, что для области изменения аргумента -2<=x<=4 видно, что глобаль- ный экстремум находится в районе x=-2 и равен примерно 75.

x  2 1.99 4

100

50

y(x) 2

0 2 4

50

100

150

x

Рис.1

Из рис.2 следует, что для другой области изменения аргумента глобальный экстремум находится районе x=0.5

**x**  0  0.01 

**y** ( **x**)

0.2

0.15



0.1

0.05

 0.05

 0.1

00.5

1 1.5

2 2.5 3

Рис.2

Найдем глобальный минимум этой функции в пределах изменения аргумента 0<x<3 , используя метод Монте-Карло. Сформируем два вектора X и Y, присвоив их нулевым эле- ментам значение нуль.

Зададимся количеством случай- **X**0  0

ных чисел N, которые мы будем ис-

**Y**0  0

пользовать для вычисления минимума. Чем больше это количество, тем точнее будет

**N** 

#### 10000

С помощью функции rnd(x) создадим вектор случайных значений элементов х i . Функ- ция rnd(x) генерирует равномерно распределенные случайные числа в интервале 0-х. Из гра- фика видно, что нам достаточен интервал 0-3.

**i** 

1  **N**

**Xi** 

**rnd** 3

Теперь в векторе Х помещено 100000 случайных чисел. Вычислим значения функции

от них и поместим их в вектор Y.

**Yi**  **y****Xi**

Величину максимального элемента этого вектора найдем, используя функцию max.

**Y**0 

**max****Y**

**Y**0 

###### 0.184

Мы вычислили (приближенно) максимальное значение заданной функции. Теперь не- обходимо определить значение аргумента, соответствующее минимальному значению функ- ции. .Для этого составим небольшую программку и вычислим по ней ответ:

**X0** 

**for**

**i**  **1**  **N X0**  **Xi if X0**

**Y0 Yi**

**X**0 

###### 0 .5

**Задача 4.** Вычислить максимумы и минимумы нижеприведенных функций:

*y = x4*  2x3 *+* 5

 2 *< x <* 2

*y =*

0 *< x <* 3

3 *(x*3 2x 2 *)*

*y=*sin2x− *x*

– *π* / 2 *<x<π* /2

ЛАБОРАТОРНАЯ РАБОТА № 3. ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВА- НИЯ

Характерной тенденцией в построении современных систем управления является стремление получать системы, которые являются наилучшими.

При этом задачи управления сводятся к нахождению наилучшего процесса из множе- ства возможных процессов, т.е. относятся к классу задач ***оптимального управления***.

В тех случаях, когда цель управления может быть достигнута несколькими различны- ми способами, на способ управления можно наложить добавочные требования, степень вы- полнения которых может служить основанием для предпочтения одного способа управления всем другим.

Во многих случаях реализация процесса управления требует затраты каких-либо ресур- сов: затрат времени, расхода материалов, топлива, электроэнергии. Следовательно, при вы- боре способа управления следует говорить не только о том, достигается ли поставленная цель, но и том, какие ресурсы придется затратить для достижения этой цели. В этом случае задача управления состоит в том, чтобы из множества решений, обеспечивающих достиже- ние цели, выбрать одно, которое требует наименьшей затраты ресурсов.

В других случаях основанием для предпочтения одного способа управления другому могут служить иные требования, налагаемые на систему управления: стоимость обслужива- ния, надежность, степень близости получаемого состояния системы к требуемому, степень достоверности знаний и т. п.

***Математическое выражение, дающее количественную оценку степени выполне- ния наложенных на способ управления требований, называется, критерием качества управления.***

Наиболее предпочтительным или оптимальным способом управления будет такой, при котором критерий качества управления достигает минимального (иногда максимального) значения.

**Различные виды задач управления.** Различные виды задач оптимального управления отличаются друг от друга способом и последовательностью выполнения этих операций.

**Одношаговые задачи.** В одношаговых задачах не рассматриваются методы реализа- ции принятого решения, т. е. определяются не величина и характер управляющего воздейст- вия, а непосредственно значение переменной состояния системы, которое обеспечивает наи- лучшее достижение цели управления. В одношаговых задачах критерий качества называют обычно **целевой функцией** или **функцией выигрыша** или **функцией потерь**. Методы ре- шения одношаговой задачи называются методами ***математического программирования***.

Математическое программирование представляет собой не аналитическую, а числен- ную форму решения, т.е. дает не формулу, выражающую конечный результат, а указывает лишь вычислительную процедуру, которая приводит к решению задачи.

Поэтому методы математического программирования эффективны лишь при использо- вании ЭВМ.

**Задача линейного программирования** является простейшим случаем задачи матема- тического программирования. Метод решения таких задач разработал советский математик **Канторович,** за что он получил Нобелевскую премию. Задача линейного программирования состоит в следующем:

Дана система *т* линейно независимых уравнений с n неизвестными x1, x2 ,…….x n , назы- ваемая *системой ограничений* задачи линейного программирования:

*a11x1+…a1nx n=b1*

*………………* (1)

*am1x1+…amnx n=bm*

где, не уменьшая общности, можно считать *b I >=0 , i=1…m.*

Характерной особенностью данной задачи является то, что число уравнений меньше числа неизвестных, т.е. m<n*.* Требуется найти неотрицательные значения переменных (*xi>=0, i=1…n*)*,* которые удовлетворяют уравнениям (1) и обращают в минимум (максимум) критерий оптимальности, который в данном случае называют **целевой функцией**.

*q=c1x1 +….c n x n. (2)*

**З**адачи линейного программирования можно решать в различных пакетах. Для этого не нужно знать математический метод их решения, но нужно уметь поставить задачу.

**Задача 1.** Кондитерский цех может производить три вида карамели: А, В, С. Затраты сырья, запасы сырья в цеху и прибыль от реализации каждого вида карамели приведены в таблице.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид сырья | Затраты сырья т/т | | | Запасы сырья  т. |
| Сах. Песок | 0,8 | 0,5 | 0,6 | 800 |
| Патока | 0,4 | 0,4 | 0,9 | 600 |
| Фруктов. Сироп | - | 0,1 | 0,1 | 120 |
| Прибыль руб /т | 108 | 112 | 126 |  |

Составить план производства карамели в цеху, при котором прибыль будет максимальной.

Это задача на нахождение оптимального решения. Опишем задачу математически.

Обозначим через Х1 – количество производимой цехом карамели А, Х2- количество производимой цехом карамели В,

Х3- количество производимой цехом карамели С, F- прибыль от производства всех трех видов карамели.

Прибыль цеха от производства карамели всех трех видов будет описываться уравнени- ем F=108\*Х1+112\*Х2+126\*Х3

И нам требуется найти максимум этой функции. Это - критерий оптимальности задачи (целевая функция). По-видимому, при производстве нельзя использовать сырья больше имеющихся в цехе запасов, т.е.

0,8\*Х1+0,5\*Х2+0,6\*Х3<=800

0,4\*X1+0,4\*X2+0,3\*X3<=600

0,1\*X2+0,1\*X3<=120

Кроме того, количество произведенных конфет не может быть отрицательным (это зна- чило бы, что цех не производит, а покупает конфеты). Поэтому,

Х1>=0, X2>=0, X3>=0.

Теперь мы имеем полное математическое описание задачи. Так как все уравнения и не- равенства в этой задаче линейны, мы имеем задачу ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ.

Решение задачи в Маткаде.

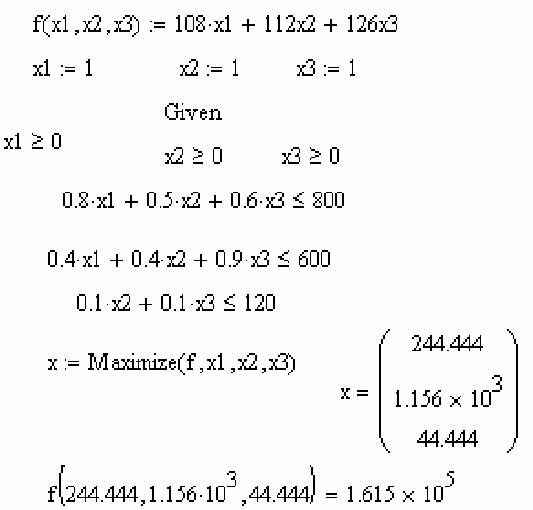


Рис.1 Решение задачи линейного программирования в Маткаде

**Задача 2:** Для поддержания нормальной жизнедеятельности человеку необходимо по- треблять в день не менее 118 г белков, 56 г жиров, 500 г углеводов и 8 г минеральных солей. Эти питательные вещества содержатся в разных количествах в разных пищевых продуктах. В таблице приведено количество питательных веществ в различных продуктах в г/кг и цена этих продуктов за 1 кг. Необходимо составить дневной рацион, содержащий минимальную суточную норму питательных веществ при минимальной их стоимости.

Пит. вещества Продукты

Мясо Рыба Молоко Масло Сыр крупа Картоф

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Белки | 180 | 190 | 30 | 10 | 260 | 130 | 21 |
| Жиры | 20 | 3 | 40 | 865 | 310 | 30 | 2 |
| Углеводы | - | - | 50 | 6 | 20 | 650 | 200 |
| Минеральные | со- 9 | 10 | 7 | 12 | 60 | 20 | 10 |
| ли |  |  |  |  |  |  |  |
| Цена руб. /кг | 35 | 18 | 5 | 45 | 42 | 12 | 2 |

Обозначив через

Х1 –количество мяса, Х2- количество рыбы, Х3- количество молока, Х4- количество масла, Х5- количество сыра,

Х6- количество крупы и Х7- количество картофеля,

потребляемых человеком в день, можем составить уравнение общей стоимости F пи- тания в день:

*F =* 35*X*1 *+*18*X* 2 *+* 5X3 *+* 45*X* 4 *+* 42 *X* 5 *+*12 *X* 6 *+* 2X7

Нам нужно найти минимум F.

сюда

Суммарное количество белков в рационе человека должно быть не меньше 118 г. От-

180 *X*1 *+*190 *X* 2 *+* 30 *X* 3 *+*10 *X* 4 *+* 260 *X* 5 *+*130 *X* 6 *+* 21*X* 7  118

Такие же неравенства составляем для жиров, углеводов и солей. Имеем:

20 *X*1 *+* 3X2 *+* 40 *X* 3 *+*865*X* 4 *+* 310*X* 5 *+* 30 *X* 6 *+* 2X7  56

50 *X* 3 *+* 6X4 *+* 20 *X* 5 *+* 650 *X* 6 *+* 200 *X* 7  500

9X1 *+*10 *X* 2 *+* 5X3 *+*12 *X* 4 *+* 60 *X* 5 *+* 20 *X* 6 *+*10 *X* 7  8

Кроме того, так как человек потребляет, а не выделяет продукты , ни один из аргументов Х не может быть отрицатель- ным. Отсюда:

*X* 1≥ 0

*X* 2≥ 0

*X* 3≥ 0

*X* 4≥ 0

*X* 5≥ 0

*X* 6≥ 0

*X* 7≥ 0

Теперь математическое описание задачи составлено полностью. Решить задачу в Маткаде по аналогией с предыдущей..

Ориентировочный ответ таков: Чтобы получить все необходимые питательные веще- ства при минимальной стоимости продуктов человек не должен есть ни мяса, ни рыбы, ни молока, ни сыра. Он должен потреблять около 20-40 г жиров в день и 500-800 г крупы или 800-900 г картофеля. При этом стоимость дневного рациона составит около 12 рублей.

**Задача 3**При откорме животных каждое животное ежедневно должно получить не ме- нее 60 ед. питательного вещества А, не менее 50 ед. вещества В и не менее 12 ед. вещества С. Указанные питательные вещества содержат три вида корма. Содержание единиц питатель- ных веществ в 1 кг каждого из видов корма приведено в следующей таблице:

Составить дневной рацион, обеспечивающий получение необхо- димого количества питательных ве- ществ при минимальных денежных затратах, если цена 1 кг корма I вида составляет 9 руб., корма II вида — 12 руб. и корма III вида — 10 руб.

|  |  |  |  |
| --- | --- | --- | --- |
| Питатель- ные  вещества | Количество единиц питательных ве- ществ в 1 кг корма вида | | |
| I | 11 | 111 |
| А | 1 | 3 | 4 |
| В | 2 | 4 | 2 |
| С | 1 | 4 | 3 |

Составить математические зависимости и решить задачу самостоятельно.

**Задача 4.**Четыре предприятия данного района для производства продукции используют одно и тоже сырье. Потребности в сырье каждого из предприятий соответственно равны 120,50,190 и 110 единиц. Сырье сосредоточено в трех местах его получения, а запасы соот- ветственно равны 160, 140,170 единиц. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и за- даны матрицей Здесь

 7 8

A   4 5

 9 3

1 9 

9 8 

2 6 

A1  1  7

A1  2  8 т. д.

- цена завоза сырья из пункта 1 в пункт 1;

- цена завоза сырья из пункта 1 в пункт 2, и

Требуется составить такой план перевозок, при котором об- щая стоимость перевозок является минимальной.

Это так называемая транспортная задача.

Нарисуем схему возможных маршрутов перевозок:

1 2 3 4

Куда перевозят



Откуда перевозят

1 2 3

Рис.2 Транспортная задача Перевозка из третьего источника сырья на схеме не показана.

Обозначим количество груза, перевезенного из пункта 1 в пункт 1 через Х11,

из пункта 1 в пункт 2 через Х12,

|  |  |  |  |
| --- | --- | --- | --- |
| из | 1 | 3 | Х13, |
| из | 2 | 1 | Х21 |

и т.д.

Нам необходимо минимизировать общую стоимость перевозок F. Воспользовавшись матрицей стоимости С, составим уравнение этой стоимости. В общем виде

F= а 11Х 11 +а 12 Х 12 + а 13Х 13 + а 14Х 14 + а 21Х 21+а 22Х 22 +а 23Х 23 + +а 24Х 24 + а 31 Х 31+а 32Х 32+а 33Х 33 + а 34Х34

или в цифрах: F=7X 11+8X 12+1X 13+2X 14 +4X 21+5X 22+9X 23+8X 24+9X31+2X 32+ +3X

33+X34

Запасы сырья в каждом пункте отправления и потребности в нем в каждом пункте переработки ограничены.

Для первого пункта переработки Х11+Х 21+ Х 31=120.

Соответственно, для остальных пунктов переработки имеем:

Х12+Х 22+Х 32 = 50,

Х 13+ Х 23+ Х 33=190,

Х 14+Х 24+Х 34 =110.

Для пунктов отправки сырья имеем:

Х 11+Х 12+Х 13+Х 14 =160

Х 21+Х 22+Х 23+Х 24= 140

Х 31+Х 32+Х 33+Х 34 =170

Кроме того, так как мы возим сырье только в одну сторону, все переменные должны быть положительными:

X 11>0 ,X 12> 0, X 13>0, X 14>0, X 21>0, X 22>0,X 23>0, X 24>0,X31>0, X 32>0,X 33>0,X 34>0.

Теперь мы полностью составили уравнения задачи и можем решать ее. Решить задачу в Маткаде.

Наберем начальные значения переменных X (по-прежнему равные 1), целевую функ- цию, все ограничения и все числа аналогично задачам 1 и 2. Получим решение.

Математическую запись решения можно записать более компактно, используя матричное исчисление.

Все переменные задаем в форме матрицы. Заметив, что целевая функция может быть представлена в виде следа ( суммы диагональных элементов) произведения АХТ, где Т – ин- декс транспонирования, записываем ее в такой форме. Все ограничения записываем также сокращенно. Ответ получен в виде матрицы z.

И так, следует перевезти из пункта 1 в пункт приема 3 160 единиц груза, из пункта 2 в пункт приема 1 – 120 единиц и в пункт 2 – 20 единиц.

Из пункта отправки 3 следует перевезти в пункты приема 2 и 3 по 30 единиц, а в пункт приема 4 – 110 единиц груза. При этом минимальные затраты составят 1000 единиц стоимо- сти.

ORIGIN  1

 7

A   4

 9

8 1 2 

5 9 8 

2 3 1 

 1 1 1 1 

X   1 1 1 1 

 1 1 1 1 

f(X)  trA  T

i  1  4 j  1  3

Given

X  0

X1  i 160 X2  i 140

X3  i 170

i

Xj  1 120

j

i

i

Xj  2 50 Xj  3 190 Xj  4 110

j j j



X

z  Minimiz(ef  X)

 0 0

160 0 

f(z)  1000

z   120 20 0 0 

 0 30 30 110 

Рис.3. Решение транспортной задачи в матричной форме

ЛАБОРАТОРНАЯ РАБОТА№4.ОПТИМИЗАЦИЯ КАЧЕСТВА МОЛОЧНОЙ КОЛБАСЫ

Согласно стандарту вареная молочная и столовая колбасы должны состоять из говяди- ны и свинины Возникла задача добавить в ее состав мясо птицы без ухудшения качества.

Сначала были изготовлены модельные колбасы «Молочная высшего сорта» и «Столо- вая первого сорта» без добавления мяса птицы и определены качественные характеристики фарша этих видов колбас. Bсeго было проведено 30 экспериментов (по 3 параллельных опы- та). Результаты эксперимента были подвергнуты математической обработке, полученные чи- словые характеристик приняты за опорные.

Опорные (без мяса птицы) безразмерные свойства фарша модельной молочной колба- сы высшего сорта были следующие:

Влага - 69,00 ± 1,20; жир - 14,50 ± 1,00; белок-15 +- 0,4, зола- 1,00 ± 0,07; водо - связы- вающая способность – 42, удельное напряжение сдвига - 5700,00 ± 100,00.

Введем следующие обозначения массовых долей фарша:

М1 содержание говядины 1-го сорта; M 2 содержание свинины полужирной;

М3 содержание мяса птицы механической обвалки; М4 - содержание молока сухого цельного;

М5- содержание яйца цельного (или крахмала).

Тогда условия, при которых фарш с мясом кур механической обвалки максимально приближался к опорному, можно описать в виде следующей системы неравенств:

1) 70,20  77,70 М1 + 66,00 М2 + 70,00 М3 + 4,00 М4 + 74,00 М5  65,00 - содержание

влаги;

2)13,50 7,00М1 + 16,00М2 + 16,00М3+25,00М4 + 11,50М5  15,00 - содержание жира;

3) 11,90  20,20 М1 +17,00 М2+13,00 М3+ 26,00 М4+12,70М5  16,10 - содержание белка;

4) 0,93  1,10 М1+ 0,80 М2 + 0,90 М3 + 0,40 М4+ 1,10 М5  1,07 - содержание золы;

5) 30,05  60,00 М1 + 32,50 М2 + 37,00М3 +55,00 М4 +11,00 М5 5 55,00 - показатель во-

досвязывающей способности;

6) 5600,00  7000,00 М1 + 6500,00 М2 + 4700,00 М3 + 370,00 М4 +120,00 М5 5800,00

- показатель предельного напряжения сдвига.

Для простоты во всех неравенствах введено обозначение М к => М к/100, где к=1  5.

При этом получается следующее естественное условие для массовых долей: 7)М1+М2+М3+М4+М5=1.

В качестве целевой функции сначала был выбран показатель относите5льной биологи- ческой ценности искомых фаршей.

Биологическая ценность характеризует пищевые свойства, вкусовые достоинства, энер- гоемкость и безвредность продукта, служит «надежным индикатором», по которому можно тестировать ту или иную технологию производства животного сырья и продуктов.

Однако при моделировании критерий биологической ценности во многих вариантах достигал одного и того же значения 190 – 192.

Это говорит о том, что после определенного уровня (190 для данной задачи) этот кри- терий становится менее чувствительным. Иными словами, существовало множество «опти- мальных» решений, которые по критерию биологической ценности практически не отлича- лись. В этих условиях более целесообразным было поддерживать значение этого критерия на достигнутом «квазиоптимальном» уровне, а в качестве различающего критерия выбрать кри- терий себестоимости фарша, который для рассматриваемой модели выглядит так:

. СБ =200М1+ 190М2+ 85М3+ 100М4+ 40М5.

Здесь 200– цена говядины за 1 кг. , 190 – цена свинины. 85 – цена 1 кг птицы, 100 – це- на молока сухого, 40 – цена яйца.

Теперь задачу можно сформулировать так:

Найти М1,М2,М3,М4,М5, для которых С.Б достигает минимума при ограничениях 1-7

и дополнительном ограничении на БЦ:

. 150.00М1+ 180ю00М2+ 260.00М3+ 100.00М4+ 125М5≥ 190.00

В соответствии с описанными условиями данная задача была решена в Маткаде. Её ре- шение приведено на рис.1

f(M1 M2 M3 M4 M5)  200M1  190M2  85M3  100M4  40M5

М1- количество мяса по цене 200р/кг, М2 - количество свинины по цене 190р/кг

М3- количество мяса птицы 85р/кг,

М4 - количество сухого молока 100р/кг М5 -количество яиц 40р/десяток

M1  1

M2  1 Given

M3  1

M4  1

M5  1

M1  0

M2  0

M3  0

M4  0

M5  0

77.7M1  66M2  70M3  4M4  74M5  70.2

77.7M1  66M2  70M3  4M4  74M5  65

7M1  16M2  16M3  25M4  11.5M5  13.5

7M1  16M2  16M3  25M4  11.5M5  15

20.2M1  17M2  13M3  26M4  12.7M5  13.9

20.2M1  17M2  13M3  26M4  12.7M5  16.1

1.1M1  0.8M2  0.9M3  0.4M4  1.1M5  0.83

1.1M1  0.8M2  0.9M3  0.4M4  1.1M5  1.07

60M1  32.5M2  37M3  55M4  11M5  30.0

60M1  32.5M2  37M3  55M4  11M5  55.0

 0.121

7000M1  6500M2  4700M3  370M4  120M5  5800

fQ  Q  Q  Q  Q   154.257

150M1  180M2  260M3  100M4  125M5  190

0 1 2 3 4

M1  M2  M3  M4  M5 1

Q  Minimize(f M1 M2 M3 M4 M5)

Таким образом, минимальная себестоимость колбасы 154.257 р.за 1 кг.без ухудшения ее качества будет достигнута при содержании

говядины -12,1%, свинины- 56,45, птицы -22,95, без сухого молока, при содержании яиц - 8,6%.

Сначала набраны выражение для целевой функции (себестоимости) и присвоены про- извольные начальные приближения переменным. Затем набирается ключевое слово GIVEN, открывающее блок решения систем уравнений, после чего набираются условия неотрица- тельности и ограничения переменных.

. В заключение c помощью встроенной функции Q= minimize(f, M1,M2,M3,M4,M5) вычисляется минимальная себестоимость f(Q0,Q1,Q2,Q3,Q4) и значения переменных в виде вектора Q.

ЛАБОРАТОРНАЯ РАБОТА №5. ВЫБОР ОПТИМАЛЬНОЙ ТРАЕКТОРИИ

**(Динамическое программирование).**

Системы, изменяющиеся во времени называются динамическими. В динамических сис- темах процесс управления разбивается на несколько последовательных шагов, причем реше- ние, принимаемое на каком либо шаге, зависит от результатов выполнения решения преды- дущего шага. Такие процессы называются ***многошаговыми процессами*** принятия решения.

**Понятие вариационной задачи.** Математическая формулировка динамических задач оптимального управления сводится к следующему. Имеется объект управления, состояние которого характеризуется системой дифференциальных уравнений.

x'  g(x u)  x(0) 

c(1)

Характер процессов в объекте управления можно изменять, используя то или иное уп- равление u из пространства допустимых управлений U. В общем случае управление u может быть также многомерной величиной. За критерий качества управления принимается ин- тегральная оценка вида

*T*

*J = Q [x(t),u(t)]dt*

(2),

1  1

0

имеющая физический смысл потерь, где Т – время протекания процесса управления, a Q1[x(t),u(t)]=q1(t) – мгновенные потери в момент t при состоянии системы x(t) и управлении u(t). Добавочными ограничениями могут быть ограничения, накладываемые на количество ресурсов или пределы изменения некоторых параметров, выражающиеся математически со- отношением

*T*

 *H*[ *x*(*t*), *u*(*t*)]*dt*  *K*

0

(3)

***Оптимальным называется такое управление из множества допустимых уп- равлений U, при котором для объекта, описываемого дифференциальным уравнением (1), и заданных ограничениях на используемые ресурсы (3) критерий качества управления (2)***

***принимает минимальное (максимальное) значение.***

Сформулированная подобным образом задача оптимального управления относится к классу ***вариационных задач***, решением которых занимается раздел математики, получивший название вариационного исчисления.

Величина J1(u), определяемая соотношением (2), получила название ***функционала***.

В отличие от функции, например f{x), численные значения которой задаются на мно- жестве значений аргумента х, численные значения функционала J1(u) задаются на множестве всевозможных управлений u(t).

Задача нахождения оптимального управления сводится к тому, чтобы из множества до- пустимых управлений U выбрать такое, при котором функционал J1(u) принимает мини- мальное численное значение.

При отыскании оптимального управления вариационными методами приходится стал- киваться с трудностями, ряд которых носит принципиальный характер:

1. Вариационные методы дают возможность находить только относительные максиму- мы и минимумы функционала J(u), тогда как интерес представляет нахождение абсолютного максимума или минимума.
2. На значения управляющих сигналов обычно бывают наложены ограничения, делаю- щие невозможным поиск оптимального управления вариационными методами.

Трудности, состоящие на пути решения задачи оптимизации управления, привели к ин- тенсивному изучению проблемы оптимальности рядом ученых в Советском Союзе и в США. Советские математики Л. С. Понтрягин и его ученики В. Г., Болтянский, Р. В. Гамкрелидзе, Е. Ф. Мищенко создали теорию оптимального управления, в основе которой лежит сформу- лированный Л. С. Понтрягиным **принцип максимума**. Этот принцип позволил поставить теорию оптимального управления на строгую математическую основу и открыл широкие возможности для ее практического применения в области автоматических систем управле- ния.

Американский математик Р.Беллман разработал иной подход к вычислению оптималь- ных процессов, получивший название **динамического программирования.** Метод динами- ческого программирования дает в руки специалиста эффективную вычислительную проце- дуру решения задач оптимизации управления, хорошо приспособленную к использованию ЭВМ. Для дискретных систем (а на ЭВМ все системы можно рассматривать как дискретные) математическая формулировка метода динамического программирования сводится к сле- дующему.

Имеется объект управления, состояние которого характеризуется многомерной дис- кретной переменной.

*xk = (xk(* 1 *) ,*.....*xk(N) ),k =* 0,1*,*...*n* 1 (4)

Характер процессов в объекте управления можно изменять, используя то или иное уп- равление из пространства допустимых управлений *U*. В общем случае управление uU мо- жет быть также многомерной величиной. Выбор управления *uk* производится только в дис- кретные моменты*, k=0,I, ., n-1.*

*uk = (uk(* 1 *) ,*...*,uk(R) )*

Характер движения объекта управления описывается системой разностных уравнений

*xk+*1 *= xk + g(xk ,uk );k =* 0,1*, ,n* 1, *x*0 *= c* (5)

За критерий оптимальности принимается сумма вида

*n*1



*Jn (u) = Q(xk ,uk ),* (6)

*k=*0

где *Q (xk, uk)* – потери на шаге *k,* зависящие от состояния объекта *xk* и управления на данном шаге *uk.* Оптимальным называется такое управление *u* из множества допустимых управлений *U,* при котором для объекта, описываемого разностным уравнением (4), крите- рий оптимальности(6) принимает минимальное (максимальное) значение. Теперь задача за- ключается в выборе таких управлений u0,u1,. un-1, которые обеспечивают минимальное зна-

чение суммы (6).

Это минимальное значение критерия n-шагового процесса будет зависеть только от на- чального состояния х0 и его можно обозначать fn(x0). По определению имеем:

*fn (x*0 *) =* min min ...min *[Q(x*0 *,u*0 *)+ Q(x*1 *,u*1 *)+*...*+ Q(xn*1 *,un*1 *)]*

(7)

*u*0 *u*1

*un*1

Заметим, что первое слагаемое этого выражения Q(x0,u0) зависит только от управления u0, тогда как остальные слагаемые зависят как от u0 , так и от управлений на других шагах. Так, Q(x1,u1) зависит от *u1*, но оно зависит и от *u0*, так как x1=T(x0,U0).

Аналогично обстоит дело и с остальными слагаемыми. Поэтому выражение (7) можно записать в виде

*fn (x*0 *) =* min *[Q(x*0 *,u*0 *)+ fn*1*(x*1 *)]* (8)

*u*

0

min ...min *[Q(x*1 *,u*1 *)+*...*+ Q(xn*1 *,un*1 *)]*

Заметим далее, что выражение

*u*1 *un*1

представляет собой минимальное значение критерия качества управления *(n–1)*- шагового процесса, имеющего начальное состояние *x1*. Эту величину можем обозначить че- рез fn-1(x1)

Таким образом, получаем:

*fn*1*(x*1 *) =* min *[Q(x*1 *,u*1 *)+ fn*2 *(x*2 *)]*. (9)

*u*1

Эти рассуждения можно повторить, если рассмотреть (n-1)-шаговый процесс, начи- нающийся с начального состояния x1. Продолжая рассуждения, получаем аналогичное выра- жение для (n-1) шагового процесса, начинающегося с состояния хl.

*fn**l (xl ) =* min *[Q(xl ,ul )+ fn**(l+*1 *)(xl+*1 *)]*

*ul* (10)

Уравнение (11), называемое уравнением Беллмана, представляет собой рекур- рентное соотношение, позволяющее последовательно определять оптимальное управ- ление на каждом шаге управляемого процесса.

Сама идея оптимизации управления на каждом шаге отдельно, если трудно оптимизи- ровать сразу весь процесс в целом, не является оригинальной и широко используется на практике. Однако при этом часто не принимают во внимание, что оптимизация каждого шага еще не означает оптимизацию всего процесса в целом. Особенностью метода динамического программирования является то, что оно совмещает простоту решения задачи оптимизации управления на отдельном шаге с дальновидностью, заключающейся в учете самых отдален- ных последствий этого шага.

Из основного свойства оптимального управления следует, что оптимизация управле- ния для произвольной стадии многошагового процесса заключается в выборе только после- дующих управлений.

Поэтому бывает удобным учитывать не те шаги, которые уже были пройдены, а те, ко- торые осталось проделать, для того, чтобы привести процесс в конечное состояние. С этой точки зрения уравнение (10) удобно записать в иной форме. В выражении (10) величина n-1 означает число шагов до тконца процесса. Обозначим эту величину через k. при этом вели- чины *xi* и *ui* будем обозначать просто через *x* и *u.*Они будут означать состояние объекта и примененное управление за к шагов до конца процесса. Последующее состояние, т.е. то, к которому объект переходит из состояния х при применении управления u, обозначим через х`. При этом соотношение (10) примет вид

*f (x)=* min *[Q(x,u)+ f*

*k*

*u*

*k* 1

*(x' )]*

(11)

**Пример задачи динамического программирования.** Динамическую систему необ-

ходимо перевести из точки А в точку В (см.рис.1). Движение возможно только по горизонтали

0

**0,0** B

**0,2**

14



**0,3**

18



8

**0,1**

### 4 6 8

**6 6 *7 4***

4

**1,0**

**1,2**

19

**1,3**

22



**1,1**

1

### 3 7 8

***2 5 6 1***



1

**2,1**

5

**2,0**

1

**2,2**



2

**2,3**

### 7 3 5

***4 2 8 2***



**3,3**



**3,2**

15



3,1

11

**3,0**

7

### 5 6 4

Рис. 1.

динамического программирования имеет вид:

справа налево (управление ) или по вертикали снизу вверх (управление u2). На рисунке задана цена перехода из одного узла в другой. Нужно выбрать такой путь, чтобы суммарная цена была минимальной. Внутри окружностей заданы координаты каждого узла. Как бы мы ни двигались, решение задачи требует шести шагов. Будем решать задачу « наоборот», двигаясь из точки В с координатами (0,0) в точку А с координатами (3,3). В этом случае возможно движение только вниз или справа налево.

Рекуррентная формула для решения задачи методом

*f (x)=* min *[Q(x,u)+ f*

*k*

*u*

*k* 1

*(x' )]*

Будем действовать в соответствии с этой формулой (рис.1). Рассмотрим переход в каж- дую точку нашей системы и, если возможны варианты переходов, выберем тот переход, для которого общая сумма потерь является наименьшей. Начинаем движение из точки (0,0). В этой точке еще никаких потерь нет, поэтому ставим в прямоугольник около узла число 0. В т.(1,0) можно перейти только из т.(0,0) с платой 4 единицы. Рисуем стрелку и ставим около узла число 4. В т. (2,0) можно перейти только из т.(1,0) с общими потерями 4+1=5. Рисуем стрелку и ставим это число около узла. Аналогично проводим стрелки и заполняем узлы (3,0), (0,1),(0,2) и (0,3). После этого рассмотрим переход в т. (1,1). В нее можно перейти из т. (1,0), при общих потерях 4+8 =12, и из т. (0,1) с общими потерями 8+7 =15. Выбираем пер- вый путь, проводим стрелку и ставим около узла число 12. В т. (1,2) можно перейти из т. (0,2) или из т.((1,1). Выбираем второй путь, так как при нем общие потери 12+7=19 меньше. Аналогичным образом просчитываем и выбираем переходы ко всем остальным точкам. В результате оказывается, что существует только один путь к т. (3,3). Он то и является опти- мальным. На рис.1 он выделен жирными стрелками.

До построения собственно программы задаем координаты начальной и конечной точек, объявляем о начале отсчета с 1 (ORIGIN:=1), задаем значения коэффициентов a,b

и

ORIGIN  1

x0  2

x1  12

y0  2

y1  12

выбираем шаг d (см. рис. 2).

Рис.2

a  5

b  30

d 

y1  y0 1000

V 

S  0 x  x0

Сама программа приведена на рис.3.

y  y0

[( x1  x0) 

( y1

 y0 )]

В доцикловой части программы задаем начальные значения перемен-

for

i  1 

d

ным x,y и функции S.

В начале цикла в операторе for

u  2

u  1

if S 

if S 

2

a  x  S 

a  x  S 

2

2

b  y 

b  y 

( y  y1 )

( x  x1)

определяем количество точек. Мы проведем вычисления для четырех точек. Соответственно, j=0 ..3 Оче- видно, что каким путем мы бы не

S  S  a  x S  S  b  y

if u 1 otherwise

шли, нам придется пройти (x1-x0)/d

точек по горизонтали и (y1-y0)/d то- чек по вертикали, т. е. всего [(x1-

x  x  d

y  y  d

if u 1

if u 2

x0)+(y1-y0)]/d точек.

Далее в цикловой точке мы,

i  1  x i  2  y i  3  u

V

V

V

V  4  S V  i

1

прежде всего, выбираем управление, минимизируя полную цену. Полная цена всего движения для данного уча- стка состоит из цены движения на всех предыдущих этапах S и прира- щения цены при движении на данном участке. При движении по горизонта- ли это приращение равно ax 2, при

1  6

V

движении по вертикали – by.

Рис.3. Программа выбора оптимальной траектории

Но так как движение системы не должно выходить за пределы квадрата x1-x0, y1- y0, то одновременно при выборе управления мы проверяем нахождение системы в этом квадрате. Для возможности движения по вертикали должно выполняться условие

y ≤ y1, а для возможности движения по горизонтали – условие x ≤ x1. Выбрав управление, мы подсчитываем новое значение полной цены S и определяем новые значе-

ния координат x и y.

13

12.267

11.533

10.8

10.067

9.333

8.6

Vj  2 7.867

7.133

6.4

5.667

4.933

4.2

3.467

2.733

2

Для сохранения в памяти и вывода на пе- чать вводим новые значения в столбцы матри- цы V. Программа составлена.

На рис.4 представлен график решения за-

дачи.

22.636.733344.656.733366.676.733388.696.73331100.1616.733132

Vj  1

Рис.4. График оптимальной траектории.

ЛАБОРАТОРНАЯ РАБОТА №6.ОПТИМАЛЬНОЕ СОЗРЕВАНИЕ

**СЫРОКОПЧЕНОЙ КОЛБАСЫ (Динамическое программирование).**

Рассмотрим задачу оптимального управления созреванием сырокопченых колбас. Процесс производства сырокопченых колбас состоит в составлении фарша из исходно-

го сырья ( говядины высшего сорта, полужирной свинины, шпика), который затем шприцу- ется в оболочку, и полученные батоны колбасы помещаются в камеру созревания.

. В процессе созревания происходят распад белков, накопление ароматобразующих ве- ществ и другие биохимические реакции. Все эти превращения приводят к накоплению мо- лочной кислоты в колбасе, что ведет к изменению показателя его кислотности рН.

С целью сокращения созревания сырокопченых колбас в них при составлении фарша вносятся стартовые культуры. Стартовые культуры способствуют подавлению жизнедея- тельности патогенной микрофлоры за счет деятельности молочнокислых бактерий.

Управлять процессом созревания сырокопченых колбас можно, варьируя управляющие величины: количество и состав стартовых культур, температуру t ◦ С и влажность воздуха ψ

% в камере созревания. Эти же величины являются и контрольными, так как их можно на- блюдать. Управляемой величиной является показатель кислотности pH.

Результаты экспериментальных исследований показали, что процесс созревания зани- мает 21 день и его можно расчленить на восемь шагов, в результате каждого из которых достигается определенные значения показателя кислотности полуфабриката.

1. ый шаг — начальный этап (составление фарша) и первый день созревания, рН1

=(5,60+5,80);

1. ой шаг — второй день созревания, pН2 = (5,50 +5,70); 3-ий шаг — третий день созревания, рН3 = (5,35 +5,5);

4-ый шаг—четвертый день созревания, рН4 = (5,20 +5,40); 5-ый шаг — пятый день созревания, рН5 =(5,10 +5,30);

1. ой шаг — шестой день созревания, рH 6 =(4,90+5,10);
2. ой шаг—с седьмого по двенадцатый день созревания, рН7 =(4,80+5,00);
3. ой шаг—с девятого/двенадцатого по двадцать первый день созревания, рН8

=(5,30).См таблицу 1.

Здесь t и ψ – практически используемые значения управляющих величин, pH – полу- чаемые при таком управлении значения показателя кислотности, pHmin и pHmax – мини

микрофлоры стартовых культур. В результате шагов 2-6 происходит подавление жиз- недеятельности патогенной микрофлоры, накопление молочной кислоты, что обусловливает сдвиг рН в «кислую» сторону. По существу на первом – шестом шагах выбираются допусти- мые управления, обуславливающие качество готовой продукции. Седьмой шаг приводит значение pH к минимуму. Восьмой шаг поддерживает термовлажностные параметры, необ- ходимые для получения продукта заданного качества с конечным результирующим значени- ем pH=5,3. На последних двух шагах управление не варьируется.

Таблица 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| День | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| pH | 5.6 | 5.55 | 5.5 | 5.4 | 5.3 | 5.2 | 5.1 | 5.0 | 4.9 | 4.85 | 4.  8 | 4.8 | 4.8 |
| PHmin | 5.6 | 5.6 | 5.5 | 5.35 | 5.2 | 5.1 | 4.9 | 4.8 | 4.8 | 4.8 | 4.  8 | 4.8 | 4.8 |
| pHma x | 5.8 | 5.8 | 5.7 | 5.5 | 5.4 | 5.3 | 5.1 | 5.0 | 5.0 | 5.0 | 5.  0 | 5.0 | 5.0 |
| t | 20 | 20 | 20 | 20 | 18 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| ψ | 90 | 88 | 86 | 84 | 82 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| шаги |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | | |

Про

должение таблицы 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 4.85 | 4.9 | 4.95 | 5.0 | 5.1 | 5.2 | 5.25 | 5.3 | 5.3 |
| 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 |
| 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 | 5.3 |
| 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |  |
| 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 |  |
| 8 | | | | | | | | |

В качестве математической модели процесса созревания сырокопченой колбасы выбе- рем авторегрессионую зависимость,

*pH i*  *ai*  *pH i* 1  *bi*  *ti*  *ci* *i*

(1)

Анализ таблицы 1 показывает, что коэффициенты « b» и «c» практически постоянны, поэтому примем b=-6\*10-4, а c=0,95\*10-4. Коэффициент «a» несколько более изменяется в процессе созревания колбасы, поэтому для этого коэффициента были приняты зависящие от номера шага Значения, приведенные в таблице 2.

Таблица 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | | |
| день | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| «а» | 1 | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 1 | 1 |

Продолжение таблицы 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | | | | | | | | |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 1.01 | 1.01 | 1.01 | 1.01 | 1.02 | 1.02 | 1.01 | 1.01 | 1 |

Созданную математическую модель необходимо, прежде всего, исследовать на адек- ватность. Для этого вычисленный по ней показатель кислотности сравнивался с показателем из таблицы 1 при значениях управляющих величин из этой же таблицы. Для этого была соз- дана программа в Маткаде, приведенная на рис.1.

PH 

PH

0

for

 PHtr

0

i  1 21

t  ttri

  tri

c  0.95   4

10

a  ai

b  6   4

10

PH  a PH  b  t  c  

i i1

i

PH

Рис.1. Программа проверки адекватности системы экспериментальным данным и ошибка модели в %

Результаты расчетов по программе приведены в таблице 3.

Таблица 3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | | |
| День | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| pH практ | 5.6 | 5.55 | 5.5 | 5.4 | 5.3 | 5.2 | 5.1 | 5.0 | 4.9 | 4.85 | 4.8 | 4.8 | 4.8 |
| pH мат | 5.6 | 5.484 | 5.371 | 5.313 | 5.20 | 5.09 | 4.99 | 4.89 | 4.79 | 4.74 | 4.69 | 4.69 | 4.69 |
| Ошибка  % | 0 | 0.174 | 1.35 | 1.61 | 1.8 | 1.9 | 1.0 | 0.12 | 0.87 | 0.87 | 0.84 | 0.81 | 0.74 |

Продолжение таблицы 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | | | | | | | | |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 4.74 | 4.78 | 4.83 | 4.87 | 4.97 | 5.08 | 5.12 | 5.17 | 5.3 |
| 4.837 | 4.884 | 4.931 | 4.979 | 5.077 | 5.177 | 5.228 | 5.279 | 5.277 |
| 0.74 | 0.68 | 0.63 | 0.59 | 0.57 | 0.58 | 0.59 | 0.61 | 0.7 |

Из таблицы 3 видно, что относительная ошибка не превышает 2%. Таким образом, соз- данная математическая модель хорошо отображает экспериментальные данные.

Мы хотим так изменить управление процессом созревания сырокопченой колбасы, чтобы хотя бы немного улучшить ее качество. Так как по мере созревания кислотность по- луфабриката должна снижаться, то для улучшения качества необходимо уменьшать кислот- ность. Для оценки качества управления процессом будем использовать функциональный критерий - выигрыш J в виде относительного суммарного количества патогенной микрофло- ры, прекратившей свою жизнедеятелъностъ в результате накопления массы молочнокислой микрофлоры, а также количества расщепленных белков и других эссенциальных веществ.

*Jt*   *pH k*  min

Запишем основное рекуррентное

уравнение динамического программирования, используя в качестве функции выигрыша

(2)

*J t* = − *pHt*

что обусловлено фактом снижения значения рН при увеличении

значения J. Так как на восьмом шаге происходит лишь поддержание параметров, то опти- мизацию необходимо проводить только для первых семи шагов.

Из выбранной функции выигрыша следует, что условную оптимизацию всех шагов возможно проводить, выбирая такие управления, при которых.

Как было *J*

стартовых культур, а *k*

 max сказано выше, управлением является количество и состав также температура и влажность в камере созревания. В

качестве множества допустимых управлений технологией было выбрано множество Ωу =

{18°С + 25°С} х {80% + 90%}, образованное путем комбинаций (прямого произведения) возможных значений температур и относительной влажности воздуха в камере созревания. Безусловная оптимизация управления соответствует таким наборам {t k , ψk} управляющих воздействий, при которых оптимальная кривая рН opt проходит ниже всех используемых на практике технологий, т.е. площадь под кривой рН будет минимальной.

Была составлена программа в Маткаде, которая приведена на рис.6.После ввода на- чальных значений температуры, влажности, показателя кислотности и критерия оптимально-

сти начинается варьирование управляющих па-

pH 

t0  20

f 0  90

pH0  5.6

J0  5.6

раметров.

В программе управление температурой t

и влажностью f ведется на первых шести шагах

( i= 1…6).

for

i  1  6

pH1i  ai  pHi 1  b  18  c  80 J1i  Ji1  pHi 1

На каждом из этих шагов выбираются пу- тем перебора значения t и f, минимизирующее критерий оптимальности J – сумму значений показателя кислотности на всех предыдущих

for

t1  18  25

шагах.

for

t1 ti  t1

f1  80  90

pH2i  ai  pHi 1  b  t1  cf1 J2i  Ji1  pHi 1

if J2i  J1i

Ji  J2i

pHi  pH2i

otherwise

pHi  pH1i Ji  J1i

f1

Для этого сформированы два вложенных цикла по температуре (t1) и влажности (f1), со- ответственно. Начиная с седьмого дня значе- ния температуры и влажности задаются посто- янными ( t=15 град. C, f=80%).

В результате расчетов получены необхо- димые для оптимального решения значения управляющих параметров, значения критерия оптимальности в случае оптимального и не оп- тимального решений, графики и таблицы зна- чений показателя кислотности для обоих случа- ев ( см. рис.7).

На этом рисунке в части А приведены оп- тимальные значения управляющих параметров

– температуры и влажности. В части Б рисунка приведены значения критерия оптимальности

for

pH

f i  f1 i

i  7  20 ti  15

f i  80

pHi  ai pHi 1  b  ti  cfi i

для случаев оптимального

Jopt =101.755 и традиционного Jtr=107.35 управлений, а также изменение этого критерия в результате оптимального решения

DJ= 5.21%

. В части В показаны значения показателя кислотности при оптимальном управлении, а в части Г по шагам графики показателя кислот- ности при оптимальном ( крестики) и традици- онном ( кружочки) управлении.

Рис.6. Маткад программа определения оптимальных параметров выдержки сырокопче- ной колбасы методом динамического программирования.

Следует заметить, что во-первых полученное улучшение качества колбасы ( величины показателей кислотности) незначительно, что указывает на хороший профессионализм практиков, и во-вторых оптимальное решение крайне чувствительно к значениям коэффи- циентов используемого разностного уравнения. режима созревания сырокопченой колбасы.

tT 

А)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 20 | 25 | 25 | 25 | 25 | 25 | 25 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |

fT 

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 |

Б)

20 20

Jopt   pH Jtr  

pHtrT

Jopt  101.755

i

i

i  1

i  0

Jtr  107.35

DJ  (Jtr  Jopt

Jtr

DJ  5.212%

В)

T



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 0 | 5.6 | 5.541 | 5.427 | 5.315 | 5.206 | 5.098 | 5.044 | 4.992 | 4.941 | 4.89 | 4.84 | 4.838 | 4.837 | 4.884 | 4.931 | 4.979 | 5.028 | 5.127 | 5.228 | 5.279 | 5.33 |

pH

Г)

pHi

5.6

5.51

5.42

5.33

5.24

pHtrT 5.15

i 5.06

4.97

4.88

4.79

4.7

0 1 2 3 4

5 6 7

8 9 10 11 12 13 14 15 16 17 18 19 20

i

Р

ис.7. Результаты расчета оптимального режима созревания сырокопченой колбасы.

А)- Необходимые для оптимального процесса значения температуры и влажности по шагам.

Б) Полученная экономия критерия оптимальности DJ= 5,212%.

В) Значения показателя кислотности при оптимальном процессе.

Г) График значений показателя кислотности при оптимальном (крестики) и стандарт- ном ( кружочки ) процессах.

ЛАБОРАТОРНАЯ РАБОТА № 7. ОПТИМИЗАЦИЯ РАБОТЫ ПРЕДПРИЯТИЯ.

**(Стохастическое динамическое программирование).**

**Стохастические задачи.** Типичным для задач управления является случай, когда имеющаяся информация бывает или недостаточна для точной оценки ситуации, или искаже- на посторонними факторами. Тем не менее, недостаточность информации не снимает задачи принятия решения. Особенность задач управления именно в том и состоит, что решение должно быть обязательно принято независимо от того, в состоянии ли мы точно оценить ре-

зультаты, к которым приведет принятое решение. Таким образом, в процессе управления возникает важная задача принятия решения в условиях, когда информация о сложившейся ситуации или недостаточна, или искажена. Данная задача получила название ***задачи приня- тия решения в условиях неопределенности.***

Метод динамического программирования позволяет решать вероятностные задачи.

**Задача 1**.Завод выпускает продукцию. Он может находиться в двух состояниях:

1. спрос на продукцию есть;
2. спроса на продукцию нет.

Работой завода можно управлять с помощью двух стратегий:

1. стратегия 1 – не тратиться на рекламу и научные исследования;
2. стратегия 2 – тратиться на них.

Для каждой стратегии задана своя стохастическая матрица ( матрица вероятностей пе- рехода из одного состояния в другое и своя матрица доходов.

 0.5

P1 

0.5 

 9 3 

 0.8

0.2 

 4 4 

 0.4

0.6 

R1 

 3

7 

P2 

 0.7

0.3 

R2 

 1

19 

Здесь P1, P2 – матрицы вероятностей переходов завода из одного состояния в другое. Например, P111 – вероятность того, что завод, находящийся в состоянии 1 останется в этом состоянии, P112 – вероятность того, что завод, находящийся в состоянии 1 перейдет в со- стояние 2, P1 22 – вероятность того, что завод, находящийся в состоянии 2 останется в этом состоянии и P121 - вероятность того, что завод, находящийся в состоянии 2 , перейдет в сосо- тяние1. Таковы же элементы матрицы P2.

В матрице R1 элемент R111 – ожидаемый доход от деятельности завода, если он оста- нется в состоянии1, R112 – ожидаемый доход завода при переходе из состояния 1 в состоя- ние 2, R122 - ожидаемый доход ( убыток,т.к. доход отрицателен) завода, остающегося в со- стоянии 2, R121 – ожидаемый доход завода при перходе из состояния 2 в состояние 1. Таков же смысл элементов матрицы R2.

Задача состоит в том, чтобы, в каком бы состоянии завод не находился, выбрать стра- тегию ( управление) ,приносящую максимальный ожидаемый доход.

Перед нами вероятностная задача: управление (выбор стратегии) изменяет вероятность перехода из одного состояния в другое. Переход происходит дискретно, т.е. это дискретный случайный процесс. При выборе стратегии вероятность перехода из одного состояния в дру- гое зависит только от состояния в настоящий момент времени. Следовательно, этот случай- ный процесс является Марковским.

Ожидаемый доход – это сумма вероятностей получения того или иного дохода. Рассмотрим первый шаг управления работой завода – первый выбор стратегии. Обо-

значим через F ожидаемый доход. Если завод находится в состоянии 1, то он может остать- ся в этом состоянии или перейти в состояние 2. При этом может быть использована как стратегия 1, так и стратегия 2. В первом случае F11 - ожидаемый доход при состоянии 1 и выборе стратегии 1 составляет: F11 =P111 R111 +P112 R112. После подстановки чисел получим: F11= 6.

Аналогично, при выборе стратегии 2 ожидаемый доход будет составлять F12= P211R211+P212R212=4.

Однако завод может находиться в состоянии 2 и для управления также может быть вы- брана одна из стратегий. В этом случае

F21=P122R122+P121R121= -3 при выборе первой стратегии и F22=P222R222+P221R221= -5 при выборе второй стратегии. Эти расчеты можно сделать выбором на первом шаге управле- ния. Для получения максимального ожидаемого дохода следует выбрать стратегию 1 ( не использовать научные исследования и рекламу) , в каком бы состоянии завод не находился.

На каждом следующем шаге «n » управления мы должны выбирать управление так, чтобы получить максимальный ожидаемый доход за все «n» шагов. Обозначим **макси-**

**мальный** ожидаемый доход за первый шаг при нахождении завода в состоянии 1 через f1, а при нахождении завода в состоянии 2 через f2. Из предыдущих расчетов следует, что f1=6 и f2=-3.

На втором шаге управления ожидаемый доход с учетом максимального ожидаемого дохода на первом шаге составит при нахождении завода в состоянии 1 и выборе стратегии1 F11= P122(R122+ f1) + P112 (R112+f2) =0,5(9+6) + 0,5 (3-3) =7,5 или

F12=P211(R211+f1) +P212 (R212+f2)=0,8(4+6) +0,2( 4-3) =8,2 при выборе стратегии

2.Очевидно, что при нахождении завода в состоянии 1 следует для получения максимального ожидаемого дохода выбирать стратегию 2, т. к. f1=8,2.

При нахождении завода в состоянии два, проведя аналогичные расчеты получим

F21=P1 22 (R122+f2) +P121(R121+f1) =-2,4 при выборе первой стратегии и F22=P222(R222+f2)

+P221(R221+f1)= -1,7 при выборе второй стратегии. Очевидно, что в обоих случаях для полу- чения максимального ожидаемого дохода нужно выбирать стратегию 2.

ORIGIN 1

Подобным образом рассчи- тывается оптимальное управление

P1   0.5

0.5 

R1   9 3 

P2   0.8

0.2 

R2   4

4  и на всех последующих шагах.

 0.4

0.6 

 3 7 

 0.7

0.3 

 1 19 

Ниже представлена програм- ма в Маткаде для решения данной

u 

f 

for

 0 

 0 

i  1 n

n  2

задачи.

Запись ORIGIN=1 означает, что счет начинается не с нуля, а с единицы. Далее приводятся матрицы перехода и матрицы доходов для двух стратегий. В самой про- грамме вычисляются доходы Fi,j для всех

F  P1

R1

 f   P1

R1

 f 

четырех возможных случаев и выбирается

1 1

1 1

1 1 1

1 2

1 2 2

управление «u» в зависимости от того, ка-

F  P2

R2

 f   P2

R2

 f 

1 2

1 1

1 1 1

1 2

1 2 2

кие доходы больше. Принято U=1 для пер-

F  P1

R1

 f   P1

R1

 f 

вой и U=2 для второй стратегии. После это-

2 1

2 2

2 2 2

2 1

2 1 1

го формируется вектор «max» для каждого

F  P2

R2

 f   P2

R2

 f 

случая. Программа набрана для случая вы-

2 2

2 2 2 2 2

2 1

2 1 1

u1  1 u1  2 u2  1 u2  2

i

i

i

i

if F  F

1 1 1 2

 

otherwise

if F  F

 

2 1 2 2

otherwise

числения «u» и приведены его значения для двух шагов управления.

**ЗАДАНИЕ:** 1. Набрать программу.

2. Вычислить значения векторов «u»,

«max», двух, пяти и 10 шагов управления.

max  F if F  F

1 i 1 1 1 1 1 2

max  F

otherwise

1 i 1 2

max  F if F  F

2 i 2 1 2 1 2 2

max  F

otherwise

2 i 2 2

f1  f1  max 

1 i

f2  f2  max 

2 i

i

u

u   1 2 

 1 2 

ЛАБОРАТОРНАЯ РАБОТА №8. ОЦЕНКА КАЧЕСТВА ПИЩЕВЫХ

**ПРОДУКТОВ.(Оценка качества пищевых продуктов методами регрессионного анали- за).**

Задача 1. Оценка качества томатов.

В качестве объективного показателя при оценке качества томатов выбрали оптическую плотность при длине волны =360 нм ().

Установлена экспериментальная зависимость 10- балльной органолептической оценки

(y) от оптической плотности (x):

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| х | 0.98 | 1.1 | 1.25 | 1.39 | 0.99 | 1.14 | 1.28 | 1.43 | 1.1 | 1.22 | 1.33 | 1.4  5 |
| y | 5 | 4.5 | 3.5 | 3.0 | 5.0 | 4.5 | 3.5 | 3 | 4.5 | 4 | 3.5 | 3 |

Для решения задачи объективной оценки качества томатов используем линейную регрессию. В Маткаде, как уже говорилось выше, коэффициенты линейной регрессионной зависимости определяются с помощью встроенных функций slope ( наклон) и intercept (от- резок, отсекаемый на координатной оси)

ORIGIN  1

 5 

 0.98 



 4.5 

 1.1 

 1.25 

 3.5   

 

3.0

 1.39 

   0.99 

 5.0   

 

4.5

vx   1.14 

intercept(vx vy)  9.59

vy   

 1.28 

 3.5   

slope(vx vy)  4.644

 

3.0

 1.43 

   1.1 

 4.5   

 

4.0

 1.22 

   1.33 

 3.5 



 3.0

 1.45 

 

Рис.1. Исходные данные и вычисление коэффициентов регрессии.

На рис. 1 приведены вектора исходных данных – вектор vy- органолептической оценки качества томатов и вектор vx- оптической плотности томатов.

Здесь же показано применение встроенных функций вычисления коэффициентов прямой и уравнение линейной регрессии, использующее эти коэффициенты.

На рис.2 точками показана органолептическая оценка и сплошной линией – аппрокси- мирующая ее прямая. Здесь же показаны максимальные ошибки оценки в баллах (она равна 0,204 балла) и в процентах ( 4,083 %).

x  0.9  1.9

j  1  12

6

5

y(x) 4

max  0.204

max  4.083

3

vyj

2

1

0.8 1 1.2 1.4 1.6 1.8 2

x vxj

Рис.2 График оценки качества томатов.

Задача 2. Инструментальная оценка качества шампанского

Сегодня существенная часть шампанских вин в России и в мире производится непре- рывным способом. Этот способ был разработан в СССР еще в 60-х годах, и авторы разра- ботки Г.Г. Агабальянц, А.А. Мержаниан и С.А. Брусиловский были награждены Ленинской премией. При таком способе производства обычный выборочный органолептический кон- троль недостаточен. Поэтому задача инструментальной оценки качества шампанских вин является вполне своевременной.

Описанная ниже инструментальная оценка качества шампанского разрабатывалась по статистическим данным, приведенным в (С.П. Авакянц. Биохимические основы технологии шампанского. М., «Пищевая промышленность»1980).

В этой книге приведены экспертные балльные оценки и состав, а также физико- химические, окислительно-восстановительные, биохимические показатели и показатели бу- кетистых веществ (всего 35 показателей) для 59 образцов Советского шампанского.

На основе этих показателей была построена теоретическая балльная оценка, которая сравнивалась с экспертной.

Задача решалась методом наименьших квадратов в пакете Маткад .Решение проводи- лось в решающем блоке GIVEN с использованием встроенной функции MINERR.

Первоначально была проведена проверка экспериментального материала на выбросы.

Оказалось, что весь этот материал статистически достоверен.

Затем были вычислены парные коэффициенты корреляции межу всеми показателями и парные коэффициенты корреляции показателей с органолептической балльной оценкой. В результате для дальнейшего исследования оставлено только следующие восемь показателей с максимальными коэффициентами парной корреляции и максимальными коэффициентами корреляции с балльной оценкой:

Содержание этанола % объема Титруемая кислотность г/л.

Летучие кислоты г/л, Содержание ионов железа мг/л,

Устойчивость двусторонних пленок с., Оптическая плотность при длине волны λ=280нм., Доминирующая длина волны нм.,

Свободные альдегиды мг/л.

На рис.3.показана часть сформированной из этих показателей таблицы М2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 9.14 | 11.8 | 6.8 | 0.7 | 5.3 | 8 | 0.264 | 573.9 | 7.9 |
| 2 | 8.96 | 11.9 | 6.2 | 0.6 | 1.4 | 10.8 | 0.272 | 576.6 | 3.7 |
| 3 | 8.95 | 12 | 7.3 | 0.4 | 1.6 | 12 | 0.296 | 577.4 | 5.3 |
| 4 | 8.86 | 11.5 | 7.1 | 0.72 | 4.4 | 8.5 | 0.279 | 574.3 | 4.4 |
| 5 | 8.84 | 11.4 | 6.8 | 0.53 | 2.6 | 7 | 0.277 | 575 | 9.7 |

M2 

Рис.3. Таблица исходных данных.

Здесь первый столбец – балльная оценка, а остальные столбцы – значения перечислен- ных выше показателей. Таблица имеет 59 строк и девять столбцов.

Весь отобранный экспериментальный материал был разделен на обучающую и кон- трольную части.

Обучающая последовательность использовалась для определения параметров аппрок- симирующей функции, на контрольной части оценивалась точность оценки качества по этой функции.

Для контрольной части были отведены 10 опытов из 59. Номера экспериментов кон- трольной части для большей объективности выбирались случайным образом, т. е. они об-

новлялись при каждом новом решении. Соответственно, рассчитанные параметры и резуль- таты каждого решения несколько отличались один от другого.

На рис.4.показаны операторы и номера строк случайного выбора контрольных экс- периментов.

i  1 10

ui  1  floor(rnd(59))

uT 

Рис 4. Результат случайного выбора десяти контрольных экспериментов.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 59 | 8 | 1 | 32 | 36 | 10 | 27 | 4 | 47 | 31 |

С помощью вспомогательной программы контрольные эксперименты были исключены из исходных данных, в результате чего получена показанная на рис.5 таблица М3 опор- ных значений показателей, по которым и определялась аппроксимирующая гиперпло- скость.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 8.96 | 11.9 | 6.2 | 0.6 | 1.272 |
| 2 | 8.95 | 12 | 7.3 | 0.4 | 1.405 |
| 3 | 8.86 | 11.5 | 7.1 | 0.72 | 1.405 |
| 4 | 8.84 | 11.4 | 6.8 | 0.53 | 1.261 |
| 5 | 9.08 | 11.4 | 6.8 | 0.53 | 2.558 |
| 6 | 9.04 | 12.2 | 7.3 | 0.4 | 2.555 |
| 7 | 8.85 | 11.1 | 7.5 | 0.78 | 4.123 |
| 8 | 9.14 | 11.8 | 6.7 | 0.7 | ... |

M3 

Рис.5. Таблица опорных значений показателей

После разделения экспериментального материала на обучающую и контрольную части, на основании значений этих показателей и приведенных экспертных балльных оценок ме- тодом наименьших квадратов была построена статистическая функциональная зависимость теоретической балльной оценки от этих показателей. Как известно, выбор вида аппрокси- мирующей функции в методе наименьших квадратов сам по себе является сложной задачей. В данной работе аппроксимация производилась наиболее простой -линейной функцией вида

9

*ball = a1 +* *aj*  *X j*

*j=*2

(1)

где ball – оценка в баллах , Xj-значение j-ого выбранного показателя ; aj - коэффици- енты показателей, значения которых и определяются методом наименьших квадратов.На рис.6. приведено решение задачи в Маткаде в решающем блоке **Given**.

j  1 9

aj  1

49  9 Given 2

 a1  

a M3    M3   0

a  Minerr(a)



i  1 

j i j

j  2

i 1





aT  0.25 0.508 0.238 0.494 0.072

0.024

0.251 1.305  3

7.018  4

Рис.6. Определение коэффициентов методом наименьших квадратов.

10

10

Сначала задаем начальные приближения для искомых коэффициентов aj. Их всего де- вять. Затем в блоке **given** набирается сумма сорока девяти ( по количеству строк в таблице М3) квадратов разностей инструментальной оценки (1) и органолептической балльной оцен- ки в первом столбце таблицы М3. Записывается требование равенства нулю этого выраже-

ния. Но так как выдержать это требование невозможно, ниже записывается требование опре- делить вектор коэффициентов «а» с минимальной ошибкой (**minerr**).Задача решается и ниже приведено ее решение – значения элементов матрицы коэффициентов а.

Подобрав коэффициенты «а» мы определили девятимерную линейную функцию, ап- проксимирующую зависимость балльной оценки качества шампанского в зависимости от значений объективных показателей. Теперь следует проверить, насколько точно проводится эта аппроксимация.

Сначала проведем эту проверку для опорных значений показателей, а затем для кон- трольных значений. Результаты проверки относительно опорных значений показателей при- ведены на рис.7,а результаты проверки относительно контрольных показателей - на рис8.

i 

.

1  49

vi  M2 

 9

 a 



1



a M2  j

i 1 



j

j  2

i 



vT 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 0.283 | -0.115 | -5.663·10 -3 | -0.502 | 0.236 | -0.107 |

max(v)  0.514

wT 

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | -1.352 | 2.485 | -0.034 | -2.96 | 0.658 | -2.488 | 3.111 | -1.273 | -0.063 |

min(v)  0.543

w 

i

v 100

i

 1 

max(W)  0.717

max( w)  5.308 min(w)  5.222

Рис.7.проверка точности аппроксимации для опорных значений показателей.

На рис.7 сначала приведена формула для подсчета абсолютной ошибки v в баллах, где М2- матрица опорных значений показателей, ниже — вектор значений абсолютных ошибок(для экономии места в транспонированном виде), максимальные ошибки в баллах.

Ниже приведена формула для расчета относительной ошибки w в процентах, а также ее максимальное значение. Максимальная ошибка в баллах равна 0,543, а максимальная ошибка в процентах -5,308%.

На рис.8 сначала в транспонированном виде приведен вектор номеров контрольных опытов u, затем приведена формула для расчета инструментальной балльной оценки и фор- мула подсчета абсолютной ошибки W. Ниже приведены значения абслютных ошибок, а так- же их максимальная и минимальная величины.

Затем на рисунке приводится выражение для относительной ошибки в процентах и максимальное и минимальное значение этих ошибок.

Мы видим, что максимальная ошибка инструментальной балльной оценки контроль- ных параметров не превышает 0,717 балла, а относительная

ошибка не превышает 7,836%.

max(w)  7.836

Проверка показала, что сформированная математическая модель позволяет достаточно точно определять балльность шампанского по измеренным значениям следующих показате- лей: Содержанию этанола % объема, Титруемой кислотности г/л., Летучим кислотам

г/л, Содержанию ионов железа мг/л, Устойчивости двусторонних пленок с., Оптиче- ской плотности при длине волны λ=280нм., Доминирующей длины волны нм., Свободным альдегидам мг/л.

balli  a2M2ui  2  a3M2ui  3  a4M2ui  4  a5M2ui  5  a6M2ui  6  a7M2ui  7  a1 Wi  balli  M2ui  1

T T





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | -0.388 | -0.436 | -0.184 | 0.43 | ... |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 1 | 12 | 35 | 21 | 49 | 11 | 42 | 18 | 6 | 9 |

W u

min(W)  0.436

Максимальная и минимальная ошибки в процентах

wi  Wi

100

## 

1 

max M2

Рис.8.Проверка точности аппроксимации для контрольных значений показателей.

ЛАБОРАТОРНАЯ РАБОТА №9 ОЦЕНКА КАЧЕСТВА КОНФЕТ «ЗЕФИР».

M 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| 0 | 100 | 0 | 20 | 6.4 | 1 |
| 1 | 100 | 0 | 20 | 6.4 | 1 |
| 2 | 100 | 15 | 20 | 6.4 | 1 |
| 3 | 100 | 30 | 20 | 6.4 | 1 |
| 4 | 100 | 45 | 20 | 6.4 | 1 |
| 5 | 100 | 60 | 20 | 6.4 | 1 |
| 6 | 100 | 75 | 20 | 6.4 | 1 |

Рис. 1. Таблица исходных аргументов

В предыдущей лабораторной работе методом наименьших квадратов было построено уравнение множественной линейной регрессии, причем, точность построенной гиперплоско- сти оказалась удовлетворительной. В приведенной ниже задаче предварительно также была проведена попытка построения линейной модели, однако из-за ее недостаточной точности пришлось составлять нелинейную полиномиальную модель.

Задача состояла в том, чтобы на основании измерений пяти факторов: количества бел- ка, количества сахара, температуры производства, кислотности и времени сбивания оценить пенообразующую способность конфет «Зефир».

Все исходные данные помещены в приведенной на рис1. таблице 1. На рисунке приво- дится часть этой таблицы. Полная таблица содержит 145 строк, что соответствует 145 опы- там. В каждом опыте для определенного образца зефира определялись количество белка (столбец 0 ), количество сахара (столбец 1), температура эксперимента в градусах Цельсия

POS 

(столбец 2), кислотность (столбец 3) и время сбивания (столбец 4). Для просмотра не показанных на рисунке 1 частей таблицы преду- смотрены не показанные на рисунке полосы прокрутки.

На рис.2. приведена соответствующая этим факторам поверх- ность отклика – значения пенообразующей способности конфет

|  |  |  |
| --- | --- | --- |
|  | 0 |  |
| 0 | 63.33 |  |
| 1 | 63.3 |  |
| 2 | 63.33 |  |
| 3 | 65.5 |  |
| 4 | 68 |  |
| 5 | 72 |  |
| 6 | 78 |  |

«зефир».

Рис. 2. Таблица пенообразующей способности конфет «зефир»

Вся таблица также состоит из 145 строк, причем каждая строка этой таблицы соответ- ствует той же строке таблицы 1.

Введенные данные необходимо, прежде всего, исследовать на наличие в них грубых ошибок измерения – выбросов. Как известно, для нормально распределенных случайных величин выбросом считается измерение х , выходящее за пределы M-3 xM+3, где М- математическое ожидание ( среднее арифметическое) измерения, а - его среднеквадратиче- ское отклонение.

В Маткаде среднее арифметическое вычисляется встроенной функцией mean, а средне- квадратическое отклонение – встроенной функцией stdev.

Поэтому для выявления выбросов были вычислены функции

r1= (M+3 - значение показателя) и r2= Значение показателя –(M-3)

для всех учитываемых показателей. По результатам вычислений выбросов в анализи- руемых данных не обнаружено.

Теперь необходимо разделить весь отобранный экспериментальный материал на опор- ную и контрольную части. Используя опорные эксперименты, мы сформируем гиперповерх- ность, проходящую как можно ближе к этим опорным точкам. Но ведь задача состоит в том, чтобы дать объективную оценку качества для любого нового образца зефира. Контроль- ные эксперименты будут играть роль именно этих новых образцов.

Отведем для контрольной части одиннадцать

u 

for

for

i 0  10

ui  floor(rnd(145)) i

i 0  9

опытов из 145.Номера экспериментов контрольной части для большей объективности будем выбирать случайным образом. Этот выбор показан на про- грамме рис.3..

T

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 0 | 28 | 84 | 50 | 119 | 25 | 103 | 44 | 13 | 17 |

for

i

u

j i  1  9

ui  floor(rnd(145)) j

u 

if ui uj

T



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 13 | 17 | 25 | 28 | 44 | 50 | 84 | 103 | 119 | 143 |

U

Рис.3.Программа отбора контрольных экспериментов.

В Маткаде встроенная функция rnd(x) - псевдослучайное равномерно распределенное число в пределах от нуля до х ( в нашем случае – до 145). Так как у нас число экспериментов равно 145, то вместо х в функцию следует вставить это число. Однако псевдослучайное чис- ло может оказаться дробным, а номер опыта должен быть целым. Поэтому из вызванного псевдослучайного числа функцией floor выделяется целая часть.

Результатом выполнения программы является вектор u, приведенный на рис 3. Для удобства анализа этот вектор «сортируется» функцией csort. «Отсортированный» вектор U также приведен на рисунке.

Следует иметь в виду, что при каждой реализации вектор U будет меняться, что позво- ляет легко и грамотно набирать статистику.

После определения номеров контрольных экспериментов необходимо отделить от них опорную часть. Это было сделано с помощью специальной вспомогательной программы

В результате была сформированы новые таблицы для опорных данных с именем М1, для контрольных исходных данных с именем М2, и таблицы для опорных и контрольных функций отклика с именем POS1 для опорной и POSK для контрольной частей функций.

Части полученных матриц опорных точек приведены на рис.4.В нулевых столбцах мат- риц расположены номера опорных экспериментов.

M1 

POS1 

Рис.4. Части матриц опорных точек аргументов и функций.

0

1

0

1

2

3

4

5

6

1

2

3

4

5

6

7

63.3

63.33

65.5

68

72

78

78

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 1 | 100 | 0 | 20 | 6.4 | 1 |
| 1 | 2 | 100 | 15 | 20 | 6.4 | 1 |
| 2 | 3 | 100 | 30 | 20 | 6.4 | 1 |
| 3 | 4 | 100 | 45 | 20 | 6.4 | 1 |
| 4 | 5 | 100 | 60 | 20 | 6.4 | 1 |
| 5 | 6 | 100 | 75 | 20 | 6.4 | 1 |
| 6 | 7 | 100 | 90 | 20 | 6.4 | 1 |

В каждую ячейку обеих матриц М2 и POSK вводится элемент № U , где U – элемент вектора номеров контрольных точек. В нулевой столбец обоих матриц вводится сам № U. В нулевом столбце обеих матриц расположены номера контрольных экспериментов.

M2 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 0 | 100 | 0 | 20 | 6.4 | 1 |
| 1 | 13 | 100 | 45 | 20 | 6.4 | 2 |
| 2 | 17 | 100 | 100 | 20 | 6.4 | 2 |
| 3 | 25 | 100 | 90 | 20 | 6.4 | 3 |
| 4 | 28 | 100 | 150 | 20 | 6.4 | 3 |
| 5 | 44 | 100 | 100 | 20 | 6.4 | 5 |
| 6 | 50 | 100 | 60 | 20 | 6.4 | 6 |
| 7 | 84 | 100 | 30 | 20 | 5.5 | 2 |
| 8 | 103 | 100 | 100 | 20 | 4.4 | 3 |
| 9 | 119 | 100 | 100 | 20 | 3.5 | 1 |
| 10 | 143 | 120 | 0 | 45 | 3.6 | 1 |
| 11 |  |  |  |  |  |  |

POSK 

Рис.5. Части матриц контрольных точек.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 |  |
| 0 | 0 | 63.33 |  |
| 1 | 13 | 67 |
| 2 | 17 | 83.5 |
| 3 | 25 | 80 |
| 4 | 28 | 81.5 |
| 5 | 44 | 85 |
| 6 | 50 | 74.5 |
| 7 | 84 | 72.5 |
| 8 | 103 | 105 |
| 9 | 119 | 125 |
| 10 | 143 | 98 |
| 11 |  |  |

После составления матриц М1, М2, POS1, POSK можно приступать непосредственно к решению задачи оптимальной аппроксимации функций какой-либо гиперповерхностью. Для решения задач методом наименьших квадратов необходимо заранее задаться видом искомой функции. Мы выберем степенную функцию всех рассматриваемых аргументов вида

*y*  *b*  (*b x bi* 5

5

* *b x* )

(1)

16

*i*1

*i i i*10 *i*

где все коэффициенты bi , i=1,2,…,16

подбираются компьютером методом наименьших квадратов.

Мы рассматриваем нашу оценку функций М и POS как функцию пяти переменных – аргументов, от значения которых зависит качество конфет «зефир». Поэтому мы используем для оценки коэффициентов искомой функции встроенную функцию minerr, применяемую в блоке решений **given**.

На рис 6. приведено «обучение», т.е. подбор коэффициентов для поверхности функ- ции «Пенообразующая способность».

Так как задача решается численно, то сначала, вне блока GIVEN задаются начальные значения искомых коэффициентов. Мы задаем их равными единице.

Затем в решающем блоке GIVEN набирается и приравнивается нулю сумма квадратов разностей между искомым аналитическим выражением для функции ПОС и эксперимен- тальными значениями этой функции.

i  1  16 Bi  1

T

B



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |

Give

lengthPOS11 1  5



 

j  

Bj5

 j 

    2

1 

# 

i  0



j  1

Bj M1 i

 Bj10 M1

i  B16 

POS1

i 0

 B  Minerr(B)

T



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 0 | 11.576 | 3.731 | 1.694 | -0.22 | 1.355 | 0.762 | 0.985 | 0.555 | -2.968 | 0.256 | -2.734 | -3.421 | -0.15 | -14.809 | 1.51 |

B

Рис.6. Подбор коэффициентов для аппроксимации функции ПОС методом наименьших

квадратов.

*i*

Выражение

*(M j )*

означает, что выбирается i-ая строка j –го столбца матрицы М.

После этого набирается встроенная функция minerr (minimum error – минимум ошиб-

ки).

Запись

B  Minerr(B)

Говорит о том что вектор коэффициентов В будет подобран так, чтобы сумма квадратов разностей

lengthPOS11 1  5

##### 

 

j  

Bj5

 j 

    2

1 

# 

i  0



j  1

Bj M1 i

 Bj10 M1

i  B16 

POS1 i 0

##### 

Была как можно ближе к нулю.

На рис.6.показаны значения подобранных компьютером коэффициентов.

Проверка адекватности полученного решения.

После решения получаем значения подобранных коэффициентов в векторе В.

Подбор коэффициентов осуществлялся по опорным значениям аргументов (матрица М1).Теперь необходимо проверить, какова разница между экспериментальными значения- ми функции П.О.С. и теоретическими, подсчитанными по созданной формуле. Сначала по- считаем ошибки относительно опорных точек. Ведь даже и через них теоретическая функция точно не проходит. Программа подсчета приведена на рис. 7. Здесь именем POSEX обо- значена выбранная модель. Из рисунка видно, что максимальная абсолютная ошибка со- ставляет около 14 единиц, а максимальная относительная ошибка – около 11 %, что терпимо.

i  0  lengthPOS11   1

POSEXi 

5

# 

j  1

BjM1i j

Bj5

 Bj10M1i j  B 

max1

16

 12.652



i 

POSEXi  POS11 

max

 14.058

i

i

max(delta)

 11.429 %

deltai 

maxPOS1

1 

max(delta1)

 10.122 %

Рис. 7. Программа определения ошибок расчета опорных точек.

Затем посчитаем ошибки определения контрольных точек (см. рис.8). i  0  10

16

POSKEXi 

5

# 



j  1

BjM2i j

Bj5

 Bj10M2i j  B 

1i 

POSKEXi  POSK1 

1i

i

delta1i 

maxPOSK

1 

Рис.8. Расчет ошибок определения контрольных точек.

Видим, что и в этом случае максимальная относительная ошибка составляет около 10%.

Делаем вывод: созданная модель работоспособна и может с достаточной достоверно- стью быть использована для расчетов.

ЛАБОРАТОРНАЯ РАБОТА № 10. НЕЧЕТКИЕ МНОЖЕСТВА.

**Построение функций принадлежности**. Как известно, принадлежность элемента дан- ному нечеткому множеству определяется его функцией принадлежности. Ниже приведены примеры задания некоторых функций принадлежности .

Для задания функции принадлежности используется функция Маткада IF (ЕСЛИ). В этой функции первое выражение – условие, в данном примере – это x1<=1, второе число – значение функции при выполнении условия – в примере это 0, третье число – значение функ- ции при невыполнении условия. В примере – это 0.5\*(x1-1).График функции принадлежно- сти приведен на рисунке для x1, меняющегося от 0 до 5.

**11** **x1**  **if** **1** **x1**  **1**  **1** **x1**  **0**

(**x**)  **if** [**5**  **x**  **0**  **if** [**7**  **x**  **0.5**  (**x**  **5**)  **if** [**12**  **x**  **1**  **if** [**16**  **x**  **1**  **0.25**  (**x**  **12**)  **0**]]]]

**1** **x1 i**

**4.5**

**3.375**

**2.25**

**1.125**

**0 2.5 5 7.5 10**

**x1 i**

**1**

**0.4**

**2** **x1 i**  **0.2 0 2**

 **0.8**

 **1.4**

 **2**

**4 6**

**x1 i**

**8 10**

Ниже на рис.1 приведены другие функции принадлежности и построены их графики. Если теперь присвоить х какое либо конкретное значение, то можно определить веро-

ятность принадлежности этого значения тому или иному множеству. Например, положив х1=5, получим 1(х1)=2, 2(х1)=0.1и т.д.

Напоминаем, как выполняется функция IF: сначала записывается условие, затем через запятую, чему равно , если это условие выполняется, затем, снова через запятую, чему рав- но  в случае невыполнения условия. Следовательно, формула выполняется так:

Если 5>x, то  равно 0. В противном случае, если 7> x=>5,  =0.5(x-5). В случае, если

12>x=>7,  =1. И, наконец, если 16>x=>12, то  =1-0.25(x-12), а если 16<=x, то =0

**i**  **0**  **100**

**x1i**  **0.1**  **i**

**1** **x1**  **if x1**  **1**  **0**  **0.5**  **x1**  **1** **2** **x1**  **if x1**  **2**  **1**  **0.3**  **x1**  **2**  **1**

**11** **x1i**

**1**

**0.8**

**0.6**

**0.4**

**0.2**

**0 2 4 6 8 10**

**x1i**

 **xi**

**1**

**0.75**

**0.5**

**0.25**

**0 4**

**8 12 16 20**

**xi**

Рис.1. Построение различных функций принадлежности в Маткаде.

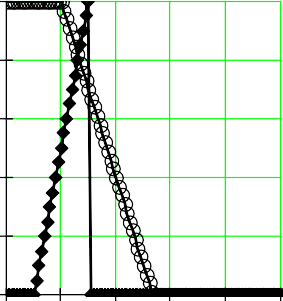
**Задача 1**. Построить в Маткаде приведенные выше функции принадлежности и их гра- фики. Вычислить в Маткаде значения всех функций принадлежности для х1=0,1,2,5,10.

**Задача 2.Операции над нечеткими множествами**. В теории множеств определены операции пересечения и объединения множеств. Пересечением *A B* двух множеств А и В

называется множество, элементы которого принадлежат как множеству А, так и мно- жеству В. Объединением *A B* двух множеств А и В называется множество, элементы ко- торого принадлежат **ИЛИ** множеству А **ИЛИ** множеству В. Для нечетких множеств, конеч- но, также существуют эти операции. Однако при этом меняются их функции принадлежно- сти.

Если новое множество является пересечением двух нечетких множеств, его функция принадлежности является минимумом из функций принадлежности каждого из первона- чальных множеств. Ниже приведены на одном графике функции принадлежности двух не- четких множеств, а рядом приведена функция принадлежности пересечения этих множеств, полученная как минимум каждого из двух исходных множеств.

**3** **x1**  **min****11****x1** **21****x1**

**1**

**0.8**

**11** **x1i** **0.6**



**21** **x1i** **0.4**



**0.2**

**0 2 4 6 8 10**

**x1i**

**0.8**

**0.6**



**3** **x1i** **0.4**

**0.2**

**0 2.5 5 7.5 10**

**x1i**

Рис.3 Функция принадлежности пересечения двух множеств.

Функция принадлежности объединения двух нечетких множеств строится как макси- мальное значение функций принадлежности каждого из объединяемых множеств. Ниже на рис.4 приведены формула и график функции принадлежности объединения двух нечетких множеств, показанных в левой части рис.3.

**Задание** 2. Провести построение всех приведенных выше функций принадлежно сти и графиков

**4** **x1**

 **max**

 **11** **x1** 

 

**1**

**0.8**

**0.6**

 

**4 x1i**

**0.4**

 **21 x1** 

**0.2**

**0**

**2 4 6**

**x1i**

Рис.4. Функция принадлежности объединения двух множеств.

.

ЛАБОРАТОРНАЯ РАБОТА № 11. ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОГО КОЛИЧЕСТВА ЖИРА И БЕЛКОВ В МОЛОЧНОЙ КОЛБАСЕ.

**Использование нечетких множеств.**

Согласно техническим условиям молочная колбаса высшего сорта должна содержать:

Жира: 13,5 -15,5 %, Белков: 14,6 – 15,4%. Нескольким дегустаторам было поручено определить наиболее предпочтительное вкусовое содержание этих составляющих в заданных т.у. пределах. При этом от них требовалась следующая четырех балльная оценка: Вкусно – 1, Скорее вкусно, чем невкусно – 0.8, Скорее невкусно, чем вкусно – 0.3, Невкусно – 0.

После статистической обработки (определялось среднее арифметическое их ответов) были получены матрицы оценок, приведенные на рис.1

**1**   **13.5 13.83**

 **0 0.34**

**14.17**

**0.67**

**14.5 14.83**

**1 0.67**

**15.16 15.5** 

**0.34 0** 

**2**   **14.6 14.73**

 **1 1**

**14.87 15 15.13**

**1 1 0.62**

**15.27 15.4** 

**0.31 0** 

для белков для жира

Рис.1. матрицы оценок содержания жира и белков в молочной колбасе. Здесь в первой строке – процентное содержание жира или белков, а во второй – оцен-

ка вкуса колбасы экспертами.

На основании мнений дегустаторов нужно определить оптимальное соотношение жира и белков в молочной колбасе.

1. Сначала составим аналитические зависимости для обеих матриц.

Каждая матрица является нечетким множеством, поэтому мы будем опрделять для них **функции принадлежности**. Рассмотрение первой матрицы показывает, что она хорошо ап- проксимируется нормальным законом распределения

*(x*  *M[x])2*



*y = e σ 2*

где M[x] – математическое ожидание случайной величины x, 2 –ее дисперсия, а С – константа. Мы обозначим y как g(g,A1,B1), где g- значение случайной величины, А1- вели- чина, обратная дисперсии, В1- математическое ожидание. Имеем:

**2**

  **A1**(**B1****g**)

**g g**  **A1**  **B1**  **e**

. (1)

Численные значение В1 определим как среднее арифметическое нулевой ( счет в Мат- каде начинается с 0) строки матрицы жира, а численное значение А1 – как статистическую дисперсию этой строки. В Маткаде среднее арифметическое вычисляется функцией mean ( v), где v – вектор- столбец, а статистическая дисперсия функцией stdev (v). У нас же имеется строка, а не столбец. Поэтому введем вспомогательный вектор –строку v1, которую потом

транспонируем. В формуле (2) каждому элементу вектора – столбца v1 присваивается зна- чение соответствующего элемента нулевой строки матрицы жиров:

**v1i**  **1 0**  **i** . (2)

Затем вычисляем эти значения, транспонируя вектор – строку v:

.

**B1**  **mean****v1T**

**A1** 

**1**

 **T****2**

**B1**  **14.499**

Здесь т – индекс транспонирования.

**stdev v1**

**A1**  **2.256**

График полученной функции принадлежности на рис.2 показывает, что она хорошо ап- проксимирует экспериментальные точки.

**g**  **12**  **12.1**  **18**

**i**  **0**  **6**

**1**

**0.8**

**g** (**g**  **A1**  **B1**) **0.6**

**1 1**  **i**

**0.4**

**0.2**

**12 13.2 14.4 15.6 16.8 18**

**g**  **1 0**  **i**

Рис.2 . График аналитического выражения для функции принадлежности жира.

Рассмотрим теперь вторую матрицу рис.1. Мы видим, что левая часть ее второй строки является константой, а правая примерно подчиняется нормальному закону. По-видимому, при аппроксимации функции принадлежности аналитическим выражением логично задаться для левой части единицей, а для правой – нормальным законом распределения.

Для определения параметров нормального закона составим симметричную вспомога- тельную матрицу µ20, приведенную на рис.3.

**20**   **14.6**

**14.73**

**14.87**

**15 15.13**

**15.27**

**15.4** 

 **0 0.31**

**0.62**

**1 0.62**

**0.31 0** 

Рис.3. Вспомогательная матрица

Повторяя все действия, проведенные для матрицы µ1, определим параметры нормаль- ного закона распределения для матрицы µ20:

1.Зададимся законом распределения

b0(b  A2  B2)  expA2(B2  b)2

где b – количество белков в колбасе, сформируем вспомогательный вектор v2 и найдем значения коэффициентов А2 и В2:,

**i**  **0**  **6**

**v2i**   **200**  **i**

A2 

1

stdev v1T

2

B2  meanv1T

A2  14.011

B2  15

Теперь создадим аналитическое выражение для функции принадлежности матрицы белка, назвав ее  (b,A2,B2) . Введем после функции принадлежности для белков последова- тельность чисел b и используем условный оператор Маткада IF.

 (**b**  **A2**  **B2**)  **if** **b**  **15**  **1**  **b**(**b**  **A**4**2**2 **B2**)

Если b меньше или равно 15, то функция принадлежности равна 1, в противном случае она равна вспомогательной функции.

График полученной функции принадлежности и аппроксимируемые точки приведены на рис.4.

**1**

 (**b**  **A2**  **B2**)

 **21**  **k**

**0.75**

**0.5**

**0.25**

 **0.998** 

**G**   **14.48**



**14 14.5 15 15.5 16**

**b**   **20**  **k**

 **14.8** 

Рис.4 График аналитического выражения функции принадлежности для белков

Для определения оптимального содержания и жиров и белков в колбасе нам нужно найти теперь пересечение обеих функций принадлежности и определить максимум новой, трехмерной функции принадлежности.

Пересечение ищем в виде функции двух переменных µgb (g,b), используя функцию Маткада min.

 **gb**(**g**  **b**)  **min**  **g**(**g**  **A1**  **B1**) 

  **b**(**b**  **A2**  **B2**) 

Трехмерный график новой функции принадлежности и его построение представлены на рис.6.

Функция  (g,b) показывает, как по мнению дегустаторов, вкус колбасы зависит от со- держания белка и жира. Естественно, мы хотим определить такое количество жира и белков в колбасе, чтобы она была как можно вкуснее, т.е. определить максимум этой функции.

Для нахождения максимума этой функции, т.е. оптимального с точки зрения экспертов

**G** 

**d**  **0**

содержания жира и белков в молочной колбасе, составим

**for**

**i**  **0**  **500**

**for j**  **0**  **500**

небольшую программу, приведенную на рис.5.

Для нахождения максимума мы , исходя из графика функции, рассматриваем содержание жира от 10 до 30%

**gi**  **10** 

**bj**  **5** 

**d1**  **gb G 1**  **gi G 2**  **bj**

**30**  **10**  **i**

**500**

**30**  **5**  **j**

**500**

**gi**  **bj** 

**if d**  **d1 if d**  **d1**

и содержание белка от 5 также до 30%. Каждый из этих участков мы делим на 500 отрезков и вычисляем значе- ние вкуса для каждого сочетания значений жира и белка, т. е. g=10, b=5; g=10,b= 5.05; g=10,b=5.1;….g=10.04,b=5;g=10.04,b=5.05….. и т.д. Вы-

численное значение вкуса – d ( начальное значение d = 0) мы сравниваем с предыдущим и запоминаем в векторе G={G0, G1, G2} все вычисленные величины только в том

**d**  **d1**

**G 0**  **d i**

**j**

**G**

**if d**

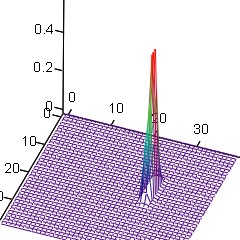
 **d1**

случае, если новое значение d больше предыдущего. В

вычисленном векторе G первый элемент – максимальное значение функции, второй – оптимальное с точки зрения экспертов содержание жира, а третий – оптимальное со- держание белков в молочной колбасе.

Рис.5. Программа вычисления максимума трехмерной функции принадлежности

**i**  **0**  **40**



**M**

**b1**  **5**

**j**  **0**  **40**

**b2**  **20**

**g2**  **20**

**g1**  **10**

**b**  **b1**  ( **b2**  **b1**)  **j**

**j 40**

**M**  **gb****g**  **b** 

**i**  **j i j**

(**g2**  **g1**)  **i**

**gi**  **g1**  **40**

Рис. 6. График трехмерной функции принадлежности.

**ЗАДАНИЕ** 1. Внимательно разобраться в изложенном выше материале. Пользуясь из- ложенным материалом, решить задачу.

ЛАБОРАТОРНАЯ РАБОТА №12. РАСПОЗНАВАНИЕ ОБРАЗОВ ПО МИНИМУМУ РАССТОЯНИЯ

Если заданы два многомерных вектора v и w , то расстояние R между ними будет оп- ределяться в скалярной или матричной форме по формулам:

 3 

 5 

 i i

4



v  w



2

i  1

 1 

 10 

R 

 7.81

w   2  v   6 

 8 

(v  w)T(v  w)

 4 

R  7.81

Условие задачи.

Даны три группы частотных спектров по четыре спектра в каждой группе бензинов: бензина А-76, бензина АИ-95 и бензина АИ-98.

Поступает спектр неизвестного бензина. Нужно вычислить «расстояния» от этого спектра до каждого из заданных, найти минимальное, и отнести поступивший бензин к той марке, расстояние до одного из спектров которой минимально.

Решение.

Перед началом решения командой **файл- настройка страницы** зададим альбомный

формат ( landscape).

Все спектры расположены в папке **Маткад- данные для лаб. работы.** В ней созданы три папки для известных марок бензина и папка « неизвестных» спектров.

1. Для начала счета с единицы введем ORIGIN: =1
2. Сначала необходимо ввести все известные (опорные) спектры в Маткад. Это произ- водится с помощью команд **меню- вставить-данные-ввод файла.** После ввода этих команд открывается окно **file options** (**опции файла**), в котором имеется кнопка

« **browse»** (**искать**). Нажав на эту кнопку, откроем окно **read from file** (**читай из фай- ла**) и укажем путь: **Маткад-данные для лаб. Работы- бензин А-76- спектр1**.Потом нажмем кнопку **открыть**.

Затем нажмем два раза кнопку **готово** в окне **file options**. В Маткаде появится рамка с надписью **А-76 …….txt.** Присвоим ему имя **F11**.

Аналогично введем остальные спектры бензина А76, присвоив им имена F12, F13, F14, и все спектры бензинов АИ-95, присвоив им имена F2i, и все спектры бензина АИ-98, при- своив им имена F3i.Здесь индексы i меняются от 1 до 4.

Ввод данных закончен.

3 . Введенные спектры представляют собой матрицу из 7462 строк и 3 столбцов: пер- вый столбец – порядковый номер, второй – частота и третий – амплитуда спектра

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | "-0.000131" | "0.009584" | "0.012917" | "0.010807" |
| 2 | "0.001253" | "0.006698" | "0.012256" | "0.013565" |
| 3 | "0.001301" | "0.004343" | "0.013362" | "0.016015" |
| 4 | "0.000805" | "0.003434" | "0.015877" | "0.018154" |
| 5 | "0.000681" | "0.004361" | "0.019087" | "0.020238" |
| 6 | "0.001579" | "0.006975" | "0.022081" | ... |

S1 

Рис.1 Вид части одного из введенных спектров.

Нас интересует только амплитуда. Поэтому сформируем из амплитуд всех введенных спектров три матрицы из четырех столбцов каждая: для бензина А-76 матрицу S1, для бен- зина АИ-95 матрицу S2 и для бензина АИ-98 матрицу S3.

Cначала введем заголовок «Формирование матриц». Для этого нажав одновременно кнопки **shift** и Э ( в английском шрифте), откроем окно надписей. Перейдя на русский и вве- дя в меню шрифт **Arial cyr**, запишем этот заголовок.

Формирование матриц

Затем сформируем сами матрицы как показано на рис.2. Цифра или буква в уголках в показателе матрицы обозначает соответствующий столбец данной матрицы.

i  1 4

  F1 3

i

i



S1

   F2 

3

i

   F3 

i

i

S3

3

Рис.2 Формирование матриц.

i

S2

F1 

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 | "1:" | "399.95" | "-0.000131" |
| 2 | "2:" | "400.44" | "0.001253" |
| 3 | "3:" | "400.92" | "0.001301" |
| 4 | "4:" | "401.40" | "0.000805" |
| 5 | "5:" | "401.88" | "0.000681" |
| 6 | "6:" | "402.37" | "0.001579" |

1

Рис.3. Матрица S1.

1. Спектры введены в формате текстов (Это видно из того, что каждое число записано в кавычках) . Их следует перевести в цифровой формат с помощью встроенной функции str2num. (string – строка, num- число). Перевод показан на рис.3

k  1 7462

k i

k i

  S2(k i)  str2numS2 

S1

 str2num S1

k i

S3  str2numS3 

k i k i

Рис.4. Перевод текстовых файлов в цифровые.

1. Подготовим «неизвестные» спектры. На самом деле спектры, которые мы называем

«неизвестными» нам известны. Первый из этих спектров принадлежит бензину А-76, второй

– бензину АИ-95, третий – бензину АИ-98. Мы должны проверить, правильно ли распознает их формируемая нами программа.

X 

X 

X1  2 3

 

Y 

j

X 3

Y  str2numY 

j

j  1 3

k j

k j

Введем «неизвестные» спектры, как мы это делали для известных спектров, дав им имена X1, X2, X3,выделим третьи столбцы и переведем их в цифровую форму ( см.рис.5).

K 

for

j  1 3

1. Составим программу поиска мини-

min

1

min

2

min

3

for







k  1 4

мального расстояния между каждым из

«неизвестных» и всеми известными спек- трами ( см рис.5 ).



S11  Y  S11  Y

1

 

j

 

T

1  

j





S21  Y  S21  Y

1

 

j

 

T

1  

j



1. Так как у нас три «неизвестных» спектра, то мы организуем цикл по j от 1 до 3.



S31  Y  S31  Y

1

 

j

 

T

1  

j



1. Переменными min1,min2,min3 мы обозначим минимальные расстояния до спектров бензинов А-76, АИ-95 и АИ-98,



S11  Y  S11  Y

k

 

j

 

T

k  

j



1k 

соответственно. В начале мы присвоим им значения расстояний от «неизвестных»

min

1

 1k

if 1  min

1

спектров до первого из спектров каждой группы.

2k  9. В каждой марке бензинов мы име-



S21  Y  S21  Y

k

 

j

 

T

k  

j





S31  Y  S31  Y

k

 

j

 

T

k  

j



k

min

2



 2k

if 2  min

2

k

ем четыре известных спектра. Поэтому ор- ганизуем цикл по «к» от 1 до 4-х. В этом цикле мы ищем минимальные расстояния

3k 

от каждого из «неизвестных» спектров до

min

3

k

 3k

if 3  min

3

k

спектров каждой марки.

1. Эти вычисленные минимальные расстояния циклом по «i» мы помещаем в

MIN  min

j

1

вектор MIN.

1. Следующим циклом по «i» мы по-

for

for

i  1 3

MIN  min

j

i

i

i  1 3

if MINj

* min

i

мещаем в вектор «К» условный номер марки бензина. Группа 1 – бензин А-76, группа 2 – бензин АИ-95, группа 3 – бензин АИ-98.

1. Наконец, для большей наглядности

Kj  i i

Kj  "76"

Kj  "95"

if MIN min

i

j

if Kj 1

if Kj 2

 "76" 

K   "95"



 "98" 

присваиваем группе 1 имя «76», а осталь- ным, соответственно, имена «95», «98».

1. ниже приведен вектор ответа. Ви- дим, что распознавание произведено пра- вильно.

Kj  "98"

j

K

if Kj 3

Рис.5 Про- грамма определения марки бензина.

ЛАБОРАТОРНАЯ РАБОТА № 13. РАСПОЗНАВАНИЕ ОБРАЗОВ ПО КОЭФФИЦИЕНТУ КОРРЕЛЯЦИИ.

Как известно, коэффициент корреляции rvw является мерой статистической связи меж- ду случайными величинами и определяется по формулам:

 9 

5

 3 

4

1 w 4



i 

4

1 

vi  Mvw  Mw

  7

i  1

1 vi i

w   2 

v   

Mw 

4

i  1

Kvw  i  1

 8 

 1 

 4 

Mv 

4



i  1

4



v  Mv



2

i

4



i  1

4



w  Mw



2

i

4

v  w 

rvw 

4

Kvw

vw

Рис.1 Коэффициент корреляции.

Здесь Mv , Mw – математические ожидания случайных величин v и w,

v, w – среднеквадратические отклонения этих величин, Kvw- корреляционный момент.

В Маткаде коэффициент корреляции вычисляется встроенной функцией corr(v,w).

Коэффициент корреляции меняется в пределах от –1 до +1.

Чем он, тем сильнее связь между двумя случайными величинами. Таким образом, ко- эффициент корреляции является **мерой сходства** между двумя случайными величинами. Су- ществуют и другие меры сходства.

Условие задачи.

Даны три группы частотных спектров по четыре спектра в каждой группе бензинов: бензина А-76, бензина АИ-95 и бензина АИ-98.

Поступает спектр неизвестного бензина. Нужно вычислить коэффициенты корреляции между неизвестным спектром и каждым из заданных, найти максимальный и отнести посту- пивший бензин к соответствующей марке.

Решение.

Перед началом решения командой **файл- настройка страницы** зададим альбомный

формат ( landscape).

Все спектры расположены в папке **Маткад- данные для лаб. работы.** В ней созданы три папки для известных марок бензина и папка « неизвестных» спектров.

* 1. Для начала счета с единицы введем ORIGIN: =1
  2. Сначала необходимо ввести все известные (опорные) спектры в Маткад. Это произ- водится с помощью команд **меню- вставить- данные-ввод файла.** После ввода этих команд открывается окно **file options** (**опции файла**), в котором имеется кнопка

« **browse»** (**искать**). Нажав на эту кнопку, откроем окно **read from file** (**читай из фай- ла**) и укажем путь: **Маткад-данные для лаб. Работы- бензин А-76- спектр1**.Потом нажмем кнопку **открыть**.

Затем нажмем два раза кнопку **готово** в окне **file options**. В Маткаде появится рамка с надписью **А-76 …….txt.** Присвоим ему имя **F11**.

Аналогично введем остальные спектры бензина А76, присвоив им имена F12, F13, F14, и все спектры бензинов АИ-95, присвоив им имена F2i, и все спектры бензина АИ-98, при- своив им имена F3i.Здесь индексы i меняются от 1 до 4.

Ввод данных закончен.

3 . Введенные спектры представляют собой матрицу из 7462 строк и 3 столбцов: пер- вый столбец – порядковый номер, второй – частота и третий – амплитуда спектра ( см. рис.2)

|  |  |  |
| --- | --- | --- |
| 1: | 399.95 | -0.000131 |
| 2: | 400.44 | 0.001253 |
| 3: | 400.92 | 0.001301 |
| 4: | 401.40 | 0.000805 |
| 5: | 401.88 | 0.000681 |
| 6: | 402.37 | 0.001579 |
| 7: | 402.85 | 0.003724 |
| 8: | 403.33 | 0.006809 |
| 9: | 403.81 | 0.010327 |

Рис.2 Вид введенного спектра.

Затем сформируем сами матрицы как показано на рис.2. Цифра или буква в уголках в показателе матрицы обозначает соответствующий столбец данной матрицы.

Нас интересует только амплитуда. Поэтому сформируем из амплитуд всех введенных спектров три матрицы из четырех столбцов каждая: для бензина А-76 матрицу S1, для бен-

j  1 3зина АИ-95 матрицу S2 и для бензина АИ-98 матрицу S3.

Cначала введем заголовок «Формирование матриц». Для этого нажав одно- временно кнопки **shift** и Э ( в английском шрифте), откроем окно надписей. Перейдя на рус- ский и введя в меню шрифт **Arial cyr**, запишем этот заголовок.

Затем сформируем сами матрицы как показано на рис.3

Формирование матриц

i  1 4

i

3

i

S2

 F2 

3

i

S3

 F3 

3

S1  F1  i i

i

Рис.3 Формирование матриц.

1. Спектры введены в формате текстов. Их следует перевести в цифровой формат с по- мощью встроенной функции str2num. (string – строка, num- число). Перевод показан на рис.4

k  1 7462

S1  str2numS1 

S2(k i)  str2numS2

 S3

 str2numS3 

k i

k i

k i

k i

k i

Рис.4. Перевод текстовых файлов в цифровые.

1. Подготовим «неизвестные» спектры. На самом деле спектры, которые мы называем

«неизвестными» нам известны. Первый из этих спектров принадлежит бензину А-76, второй

– бензину АИ-95, третий – бензину АИ-98. Мы должны проверить, правильно ли распознает их формируемая нами программа.

Введем «неизвестные» спектры, как мы это делали для известных спектров, дав им имена X1, X2, X3,выделим третьи столбцы и переведем их в цифровую форму ( см.рис.4).

X 

1



C:\Ìîè äî

j  3

X 

2

X3 

C:\Ìîè äîê



C:\Ìîè äîêó

Y  Xj

Y  str2numY 

k j k j

Рис.5. Ввод исследуемых спектров.

1. Составим программу поиска максимального коэффициента корреляции каждым из

«неизвестных» и всеми известными спектрами

K 

for

j  1 3

max  0

1

max  0

2

max  0

3

for k  1 4

1k  

k

j 

( см рис.6).

1. Так как у нас три «неизвестных» спек- тра, то мы организуем цикл по j от 1 до 3.
2. Переменными max1,max2,max3 мы обо- значим максимальные коэффициенты корреля- ции между спектрами бензинов А-76, АИ-95 и АИ-98, соответственно. В начале мы присвоим им нулевые значения .

corr S11  Y

1. В каждой марке бензинов мы имеем

max

1

 1k

if 1  max

1

k

S21  Y

четыре известных спектра. Поэтому организу-

2k 

corr k j 

ем цикл по «к» от 1 до 4-х. В этом цикле мы ищем максимальные коэффициенты корреля-

max

2

 2k

if 2  max

2

k

ции между каждым из «неизвестных» спек-

3k  corr k j 

S31  Y

тров и спектрами каждой марки.

1. Эти вычисленные макси-

max

3

k

 3k

if 3  max

3

k

мальные коэффициенты корреляции циклом по «i» мы помещаем в вектор MAX.

MAX j  0

Следующим циклом по «i» мы помещаем в вектор «К» условный номер марки бензина.

for

for

i  1 3

MAX  max

j

i

i

i  1 3

if MAX j

 max

i

Группа 1 – бензин А-76, группа 2 – бензин АИ- 95, группа 3 – бензин АИ-98.

1. Наконец, для большей наглядности присваиваем группе 1 имя «76», а остальным, соответственно, имена «95», «98».

Kj  i

i

Kj  "76"

Kj  "95"

Kj  "98"

j

K

if MAX max

i

j

if Kj 1

if Kj 2

if Kj 3

1. ниже приведен вектор ответа. Видим, что распознавание произведено правильно.

 "76" 

K   "95"



 "98" 

Рис.6. Маткад - программа определения марки бензина по максимальному коэффици- енту корреляции.

ЛАБОРАТОРНАЯ РАБОТА № 14. РАСПОЗНАВАНИЕ ОБРАЗОВ ПО НЕЧЕТКОЙ МЕРЕ СХОДСТВА.

В работах А.Е. Краснова разработано положение об **оптимальных мерах сходства**. Та- кой мерой является, например, выражение, приведенное на рис.1.



  

j k T  j k 

Y  S31  Y  S31

Рис.1 Оптимальная мера сходства.

 j

Здесь  и  константы, Y

k

неизвестный спектр, S31

* один из известных спектров. Т –

индекс транспонирования вектора.

Чем больше мера сходства, тем больше связь между двумя векторами.

Условие задачи.

Даны три группы частотных спектров по четыре спектра в каждой группе бензинов: бензина А-76, бензина АИ-95 и бензина АИ-98.

Поступает спектр неизвестного бензина. Нужно вычислить меру сходства между неиз- вестным спектром и каждым из заданных, найти максимальную и отнести поступивший

бензин к соответствующей марке.

Решение.

Перед началом решения командой **файл- настройка страницы** зададим альбомный

формат ( landscape).

Все спектры расположены в папке **Маткад- данные для лаб. работы.** В ней созданы три папки для известных марок бензина и папка « неизвестных» спектров.

1. Для начала счета с единицы введем ORIGIN: =1
2. Сначала необходимо ввести все известные (опорные) спектры в Маткад. Это произ- водится с помощью команд **меню- вставить-данные-ввод файла.** После ввода этих команд открывается окно **file options** (**опции файла**), в котором имеется кнопка

« **browse»** (**искать**). Нажав на эту кнопку, откроем окно **read from file** (**читай из фай- ла**) и укажем путь: **Маткад-данные для лаб. Работы- бензин А-76- спектр1**.Потом нажмем кнопку **открыть**.

Затем нажмем два раза кнопку **готово** в окне **file options**. В Маткаде появится рамка с надписью **А-76 …….txt.** Присвоим ему имя **F11**.

Аналогично введем остальные спектры бензина А76, присвоив им имена F12, F13, F14, и все спектры бензинов АИ-95, присвоив им имена F2i, и все спектры бензина АИ-98, при- своив им имена F3i.Здесь индексы i меняются от 1 до 4.

Ввод данных закончен.

3 . Введенные спектры представляют собой матрицу из 7462 строк и 3 столбцов: пер- вый столбец – порядковый номер, второй – частота и третий – амплитуда спектра ( см. рис.2)

|  |  |  |  |
| --- | --- | --- | --- |
| 1: | 399.95 | -0.000131 |  |
| 2: |  | 400.44 | 0.001253 |
| 3: |  | 400.92 | 0.001301 |
| 4: |  | 401.40 | 0.000805 |
| 5: |  | 401.88 | 0.000681 |
| 6: |  | 402.37 | 0.001579 |
| 7: |  | 402.85 | 0.003724 |
| 8: |  | 403.33 | 0.006809 |
| 9: |  | 403.81 | 0.010327 |

Рис.2 Вид введенного спектра.

Нас интересует только амплитуда. Поэтому сформируем из амплитуд всех введенных спектров три матрицы из четырех столбцов каждая: для бензина А-76 матрицу S1, для бен- зина АИ-95 матрицу S2 и для бензина АИ-98 матрицу S3.

Cначала введем заголовок «Формирование матриц». Для этого нажав одновременно SHIFT и Э ( в английском шрифте), откроем окно надписей. Перейдя на русский и введя в меню ARIAL CYR, запишем этот заголовок.

j  1 3

Затем сформируем сами матрицы как показано на рис.3.

K 

i

i

i

for

i  1 4 j  1 3

  10

i

S1

 F1 

3

i

S2

 F2 

3

i

S3

 F3 

3

Рис.3 Формиро-

  0.01

max  0

1

max  0

2

max  0

3

вание матриц.

* 1. Спектры введены в формате тек- стов. Их следует перевести в цифровой формат с помощью встроенной функ- ции str2num. (string – строка, num- чис- ло). Перевод показан на рис.4.

for k  1 4

1k  

k  1 7462

  

j

k T 

j

k 

S1 

 str2numS1  

S2 ( k  i)  str2num

S2 k  i

max

1

Y

 1k if

 S11  Y

1  max

k

1

 S11

k i k i

S3  str2numS3 

2k  

k i

k i

   j

k T  j

k 

Рис.4. Перевод текстовых файлов в

max

2

Y

 2k if

 S21  Y

2  max

k

2



 S21

цифровые.

* 1. Подготовим «неизвестные»

спектры. На самом деле спектры, кото-

3k  ~~X ~~

31 k TY j  S31 k

   

 max



1

j

Y

рые мы называем «неизвестными» нам

max

3

k

      S

 3k if 3k

C:\Ìîè äî

3

  известны. Первый из этих спектров при- надлежит бензину А-76, второй – бензи- ну АИ-95, третий – бензину АИ-98. Мы должны проверить, правильно ли распо-

MAX j  0

знает их формируемая нами программа. Введем «неизвестные» спектры,

for

for

i  1 3

MAX  max

j

i

i

i  1 3

if MAX j

 max

i

как мы это делали для известных спек- тров, дав им имена X1, X2, X3,выделим третьи столбцы и переведем их в циф- ровую форму ( см. рис.5).

Kj  i

if MAX max

i

j

X  X 

2 3



i

Kj  "76"

if Kj 1

C:\Ìîè äîê

C:\Ìîè äîê

K  "95"

if K 2

X 

j  X 3

j

j

K  "98"

j

if K 3

1 Y

Y  str2numY 

j j  "76" 

j K   "95"

C:\Ìîè äî

k j

k j



K  "98" 

тров

Рис.5. ввод «неизвестных» спек-

* 1. Составим программу поиска максимального значения меры сходства каждого «неиз-

вестного» со всеми известными спектрами .

* 1. Так как у нас три «неизвестных» спектра, то мы организуем цикл по j от 1 до 3.

Переменными max1,max2,max3 мы обозначим максимальные значения мер сходства со спектрами бензинов А-76, АИ-95 и АИ-98, соответственно. В начале мы присвоим им нуле- вые значения .

1. В каждой марке бензинов мы имеем четыре известных спектра. Поэтому организу- ем цикл по «к» от 1 до 4-х. В этом цикле мы ищем максимальные значения меры сходства каждого из «неизвестных» спектров и спектрами каждой марки.
2. Эти вычисленные максимальные значения циклом по «i» мы помещаем в вектор

MAX.

1. Следующим циклом по «i» мы помещаем в вектор «К» условный номер марки бен- зина. Группа 1 – бензин А-76, группа 2 – бензин АИ-95, группа 3 – бензин АИ-98.
2. Наконец, для большей наглядности присваиваем группе 1 имя «76», а остальным, соответственно, имена «95», «98».
3. Слева приведен вектор ответа. Видим, что распознавание произведено правильно.

ЛАБОРАТОРНАЯ РАБОТА № 15. НЕЙРОННЫЕ СЕТИ. ПЕРСЕПТРОН.

**Искусственный нейрон.**

Искусственный нейрон (рис.1) имитирует в первом приближении свойства биологиче- ского нейрона.

На вход искусственного нейрона поступает некоторое множество сигналов, каждый из

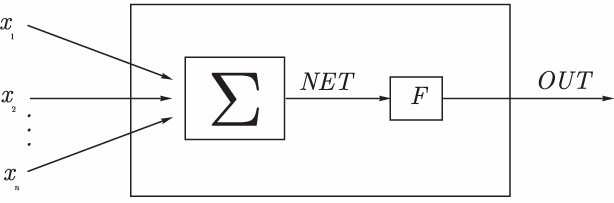
которых является выходом другого нейро- на. Каждый вход умножается на соответ- ствующий вес, и все произведения сумми- руются, определяя уровень активации нейрона.

Рис.1 Модель искусственного нейрона

Множество входных сигналов, обозначенных x1,x2,…xn, поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаются вектором X, Каждый сигнал умножается на соответствующий вес w1,w2,…wn, и поступает на суммирующий блок, обо- значенный Σ. Множество весов в совокупности обозначается вектором W. Суммирующий блок, складывает взвешенные входы алгебраически, создавая выход, который мы будем на- зывать . В векторных обозначениях это может быть компактно записано следующим образом:



*NET=XW*

Сигнал NET далее, как правило, преобра-

зуется функцией активации F и дает выходной нейронный сигнал OUT. Возможны совер- шенно различные функции активации. Но чаще всего используются (см. рис.2):

1. обычная линейная функция OUT=k(NET), где k— константа, величина которой ме- няет наклон функции.
2. пороговая функция, где k- величина порога, изменение которой сдвигает порог,
3. логистическая функция

*OUT =*

1

1*+* exp*(* *k*  *NET)*

На рис.2 приведены основные активизационные функции нейронов и способы их фор- мирования в Маткаде.

ORIGIN  100

k1  10

i  30  30

 i  1.5i

k2  0.9

k3  0.1

a3i  1

a1i  if i  k1  0  1

a2i  (k2  )i

1  expk3  i

a1 i

 60  40  20

1

0.8

0.6

0.4

0.2

0 20 40 60

 i

a2 i



60 a3i

0 20 40

|  |  |
| --- | --- |
| 60 |  |
| 40 |
| 20 |
| 60  40  20 |  |
|  20 |
|  40 |
|  |

 i

 60  40  20

0 20 40 60

1

0.8

0.6

0.4

0.2

 i

Рис.2. Основные функции активации и их реализация в Мат- каде.

Так как наши графики захватывают третий и четвертый квадранты, то счет мы начина- ем от -100 (ORIGIN:= -100).

Логистическая функция используется наиболее часто, т. к. она обладает свойством дифференцируемости, что нужно при обучении нейрона, и имеет переменный коэффициент усиления: при малых значениях аргумента он большой, а при увеличении аргумента – уменьшается. Изменение коэффициента k изменяет крутизну логистической функции.

**Задание1.** Студент сначала должен разобраться в приведенных выражениях, а затем набрать каждую функцию в Маткаде по очереди

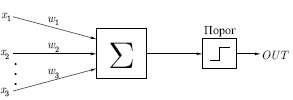
А). Для пороговой функции следует рассмотреть ее работу при значениях порога к1= 0

и +15.

Б). Работу линейной функции следует проверить при значениях коэффициента к2= 5 и -

1;

В). Вид логистической активизационной функции нужно рассмотреть при значениях

коэффициента к3 = 0.1, 1.5 и 3.

Персептрон.

Нейрон или нейронная сеть, с пороговыми функциями активации называется персептроном. ( Рис.3). Это одна из самых простых нейронных моделей. Она разработана в 60-х гг. прошлого ве- ка Розенблаттом.

Рис.3 Персептрон на три входа.

На входы персептрона подаются входные сигналы. Элемент 

умножает каждый вход на вес и суммирует взвешенные вхо-



ды. Если полученная сумма больше заданного порогового значения, выход равен единице, в противном случае - нулю.

**А). Обучение персептрона**. Персептрон способен к обучению. Сеть обучается, чтобы для некоторого множества входов давать желаемое множество выходов. Каждое такое вход- ное (или выходное) множество рассматривается как вектор. Обучение осуществляется путем последовательного предъявления входных векторов с одновременной подстройкой весов в соответствии с определенной процедурой. В процессе обучения веса сети постепенно стано- вятся такими, чтобы каждый входной вектор вырабатывал выходной вектор.

Различают алгоритмы обучения с учителем и без учителя. Обучение с учителем пред- полагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с по- мощью обратной связи подается в сеть, и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются по- следовательно, ошибки вычисляются и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

Персептрон может быть обучен не всему. Некоторым функциям, например, исклю- чающему ИЛИ, его обучить нельзя. Функции, которым можно обучить персептрон, называ- ются линейно разделимыми. Их определение составляет нетривиальную задачу. В этом огра- ниченность персептрона.

Предложенный Ф. Розенблаттом метод обучения состоит в итерационной подстройке матрицы весов, последовательно уменьшающей ошибку в выходных векторах. Алгоритм включает несколько шагов:

Шаг0: Для заданных входов задается желаемый выход. Начальные значения весов всех нейронов представляются случайными.

Шаг1: Вычисляется вектор ошибки δ между заданным и полученным значением выхо-

да.

*T*

Шаг2: Вектор весов модифицируется по формуле *W i+* 1 *=W +ηXδ* . Здесь W и Wi+1 –

i

*i*

Векторы весов до и после очередной итерации, Х- вектор входных сигналов, δТ- транспони- рованный вектор ошибки, 1<η<1 – темп обучения.

Шаг3: Шаги 1 -2 повторяются до тех пор, пока вектор ошибки не станет достаточно ма-

лым.

На рис.4 приведена Маткад- программа обучения персептрона из одного нейрона с

двумя входами.

1. Задаются порог к, случайные значения входов х01 и х02, желаемое зна- чение выхода у0=1 и нулевые начальные значения весов w.
2. Составляется программа обучения. Задается начальное значение ошибки D=2 и в цикле с оператором while D>0 подсчитываются сумма Σ, фактическое зна- чение и выхода у.
3. Подсчитывается ошибка D = y0 – y и новое значение весов с коэффици- ентом обучения η =1.
4. Цикл продолжается пока D>0. В результате получаем значение вектора

W 

D  2 while D  0

  w1x01  w2x02

W, удовлетворяющее требуемой величине ошиб- ки.

ORIGIN  1

y  0

if   k

  1

y0  1

i 

kwi100 1  2

y  1

otherwise

D  y0  y

w1  w1  Dx01

 0 

w 

 0 

x01  rnd (5)

w2  w2  Dx02 D

W  w

 6.292 

W 

 3.094 

x02  rnd (1)

x02  0.773

x01  1.573

Рис.4 Программа обучения персептрона.

Убедимся, что при полученных значениях весов, выход персептрона равен 1.Для этого составим небольшую программу, приведенную на рис.5.

y 

  W1x01  W2x02

y  0 if   k

y  1 y

otherwise

y  1

Рис.5.Проверка вычисленных весов. Из программы видно, что при подаче заданных входных сигналов выход равен 1.

Б).Обученный персептрон разделяет входные сигналы на два класса.

Если на вход персептрона подать входные сигналы, то , в зависимости от знака Σ выход будет равен 0 или 1. Тем самым входные сигналы делятся на два класса. Так как в векторной форме *Σ=XW T* , то выражение Σ=0 является границей этих двух классов.

На рис. 6 приведена программа классификации десяти точек. Случайным образом сформированы десять входных сигналов. Для подсчитанных выше весов из условия Σ=0 граница. Все входные сигналы располагаются выше или ниже этой границы в зависисти от того, равен соответствующий им выход един единице или нулю.

j  1  10

X1  10  10

W1

X2(X1)  X1

W2

30

20

X2(X1) 10

x1j  rnd (9)

x2j  rnd (9)

x2j

 10

 5  10 0 5 10

 20

 30

X1 x1j

Рис.7. Разделение входных сигналов персептрона на два класса

ЗАДАНИЕ 2. Студент должен разобраться в программах, набрать их в Маткаде и про- играть их несколько раз.

ЛАБОРАТОРНАЯ РАБОТА № 16. ОБУЧЕНИЕ ОДНОСЛОЙНЫХ НЕЙРОННЫХ СЕТЕЙ.

Обучение с учителем заключается в том, что на вход сети подаются определенные сиг- налы, а веса подстраиваются так, чтобы выход соответствовал заданному значению.

Обучение однослойных сетей.

Однослойной называется сеть, состоящая из одного слоя нейронов (см. рис.1).Здесь каждый входной сигнал р i подается с весом w i на входы сумматора данного разряда Σ i.С выходов сумматоров сигналы поступают на не показанные на рисунке активизационные функции Fi..Обучение таких сетей можно проводить, методом наименьших квадратов, ми- нимизируя квадрат ошибки – разности между реальным и заданным результатом. На рис. 2 показано такое обучение в Маткаде .

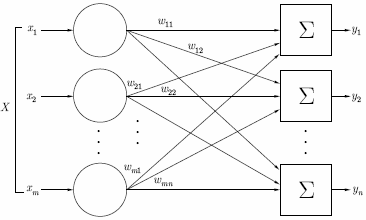


Рис. 1 Однослойная нейронная сеть.

Рассматривается однослойная нейронная сеть, состоящая из «m» нейронов с логистиче- ской функцией активации.

Конкретно в данной задаче принято m=10.

ORIGIN 1

m  10

i  1 m

j  1 m

pi  rnd (1)

ti  rnd (1)

1

|  |  |
| --- | --- |
|  | 1 |
| 1 | 0.989 |
| 2 | 0.119 |
| 3 | 8.923·10-3 |
| 4 | 0.532 |
| 5 | 0.602 |
| 6 | 0.166 |
| 7 | 0.451 |
| 8 | 0.057 |
| 9 | 0.783 |
| 10 | 0.52 |

1m 1.268·10-3

Given

2

 ti



2

3 

0.193 0.585

1  0

 iT 

p  t 

i4 1 

5

6

7

8

9

10

|  |  |
| --- | --- |
| 1 | 2.217·10-3 |
| 2 | 0.193 |
| 3 | 0.585 |
| 4 | 0.35 |
| 5 | 0.823 |
| 6 | 0.174 |
| 7 | 0.711 |
| 8 | 0.304 |
| 9 | 0.092 |
| 10 | 0.147 |

* 1. 5 expw

0.823

w0.17M4 inerr(w)

0.71

0.304

0.091

1

0.147

p 

i 

i

max(t)T 

4



10

i  ti  Ti

max  1.222

Рис.2. Обучение однослойной сети.

Здесь:

* + 1. Командой ORIGIN:=1 объявляется начало счета ;
    2. Вводится число нейронов в сети (m:=10).
    3. Начальные значения весов задаются равными 0 (wi,j :=0);
    4. Значения входных сигналов pi и выходных сигналов ti задаются случайным образом функцией rnd(1). Как известно, эта функция реализует равномерно распределенное число в пределах 0- число в скобках. Так как в скобках стоит единица, то все значения этих функций будут расположены в пределах 0 – 1. Значения выбранных входных и выходных сигналов показаны в виде векторов p и t.
    5. Далее набирается заголовок решающего блока Given и выдвигается требование, ра- венства нулю суммы квадратов разностей между желаемым выходным сигналом ti и реаль-

ным входным сигналом. Реальный входной сигнал формируется логистической активизаци- онной функцией

1

1     i T 

exp w p 

Здесь i-ый столбец матрицы весов.

i

w

Произведение этого транспонированного

столбца на вектор входных сигналов дает сумму произведений соответствующего веса на входной сигнал.

* + 1. совместное написание команд

m  1 2

1  

**А)** 

i  1

ti 



 

exp

i T

 0

p

w



и Б)

w  Minerr( w)

требует подобрать веса

i

##### w

таким

образом, чтобы команда А) выполнялась с наименьшей ошибкой.

В результате мы получаем матрицу весов (ниже помещена только часть этой матрицы)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | -2.151 | -0.46 | 0.082 | -0.165 | 0.5 | -0.508 |
| 2 | -0.259 | -0.056 | 9.938·10-3 | -0.02 | 0.06 | -0.061 |
| 3 | -0.019 | -4.16·10-3 | 7.447·10-4 | -1.497·10-3 | 4.523·10-3 | -4.593·10-3 |
| 4 | -1.164 | -0.257 | 0.053 | -0.101 | 0.279 | -0.283 |
| 5 | -1.33 | -0.307 | 0.073 | -0.131 | 0.33 | -0.335 |
| 6 | -0.368 | -0.085 | 0.02 | -0.037 | 0.092 | -0.093 |
| 7 | -1.003 | -0.238 | 0.061 | -0.107 | 0.255 | -0.259 |
| 8 | -0.127 | -0.03 | 7.702·10-3 | -0.014 | 0.032 | -0.033 |
| 9 | -1.778 | -0.457 | 0.136 | -0.229 | 0.486 | -0.492 |
| 10 | -1.192 | -0.318 | 0.099 | -0.165 | 0.337 | -0.34 |

w 

* + 1. Программа

T 

for

i  1 m

  

i T

i w p

a  1

i 1  expi

i

T  a T

высчитывает значения выходных величин Т, подсчитанных при полученным выше значениях матрицы весов.

* + 1. Далее приведены значения вычисленных Т и заданных выходных величин . Здесь же приведены максимальные абсолютная и относительная ошибки вычисления.

Задание.

* + - 1. Изучить программу решения задачи.
      2. Набрать задачу, получить решение.
      3. Проиграть задачу для трех вариантов входных и выходных величин. Убедиться, что максимальная относительная ошибка во всех случаях составляет ме- нее одного процента.

ЛАБОРАТОРНАЯ РАБОТА №17. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТОВ.

**Экспериментом** называется совокупность опытов, объединенных единой целью, еди- ной системой ограничений в пространстве и времени.

**Опытом** можно считать реализацию на каком либо объекте некоторых условий, пра- вил. В результате опыта появляется то или иное **событие.** Появление события регистрирует- ся при помощи какого либо параметра, имеющего, как правило, численное выражение и наи- более полно характеризующее результат.

Любой технологический параметр, характеризующий результат, процесса, называется

**выходом** процесса.

Результат процесса зависит от условий его протекания, характеризуемых значением параметров, влияющих на процесс. Независимые параметры процесса называются **факто- рами.** Численное значение любого фактора должно устанавливаться независимо от других факторов.

В зависимости от числа учитываемых факторов в эксперименте различают однофак- торные и многофакторные эксперименты.

Из за погрешностей измерения выходы одного процесса при разных опытах отлича- ются один от другого. Все ошибки опыта делятся на три группы: грубые, систематические и случайные.

Планирование однофакторных экспериментов.

При планировании экспериментов для упрощения нормальных уравнений метода наи-

меньших квадратов часто принимают условие

*N*

∑ *xi* = 0

*i=* 1

.План однофакторного эксперимента , составленный с учетом выполнения этого усло- вия, будет симметричным относительно центра эксперимента. Этот план дает возможность независимым образом определить коэффициенты линейного уравнения, т.е. будет ортого- нальным относительно коэффициентов линейного уравнения. Симметричный план преду- сматривает равномерное изменение исследуемого фактора от опыта к опыту:

*Ci+* 1− *Ci =λ=const*

.где Ci – значение фактора в i-ом опыте в натуральной размерности,

Ci+1 – то же для последующего опыта4 λ – интервал варьирования фактора.

Если представить значение фактора в безразмерном выражении x I и за точку отсчета принять С0, то переход к новым координатам имеет вид::

*x* = *Ci*− *C* 0

1 *N*

*i λ* , где С0- центр

*C* 0= ∑ *Ci*

*i=* 1

*N*

эксперимента.

В результате преобразования факторы хi становятся безразмерными.

Таким преобразованием переменной достигается симметричность плана и его ортого- нальность в отношении коэффициентов линейного уравнения.

Назначение координат центра эксперимента т интервалов варьирования факторов во многом определяют эффективность эксперимента.

Двухуровневые планы многофакторных экспериментов.

В многофакторном эксперименте можно учитывать только зависимость выхода от каж- дого из факторов, а

можно учитывать также зависимость выхода от взаимодействия нескольких факторов.

Если учитывается взаимодействие всех факторов, то многофакторный эксперимент

.называется **полным**.

Самым простым планом многофакторного полного эксперимент является план, в кото- ром исследуемые факторы изменяются лишь на двух уровнях: верхнем C + и нижнем C - .

i i

Такой план называется **двухуровневым** и обозначается ПФЭ2n **,** т.е**.** полный факторный

эксперимент двухуровневый , n- факторный.

Центр эксперимента ,

*Ci0*

= *Ci +C i*−

2

интервал варьирования

*λ* = *Ci* − *Ci*− *=C* − *C*

*i* 2 *i*

*i0 =C i0*− *Ci*−

В безразмерном выражении верхний уровень фактора будет выражаться +1, нижний \_1:

*x* = *Ci* − *Ci0* = *Ci* − *Ci0* =+1

*λi Ci* − *Ci0*

*x* = *Ci*− − *Ci0* = *Ci*− − *Ci0* = − 1

−

*i i Ci0* − *Ci*−

*λ*

План ПФЭ2n можно представить таблицей.

Пример. Построить план ПФЭ22 для исследования влияния температуры в диапазоне 30

– 42 градуса Цельсия и величины pH в диапазоне от 5 до 7.

Найдем центр эксперимента

*λ* = *C*1 − *С*1− = 42− 30 = 6

1 2 2

*λ* = *С*2 − *С* 2− = 7− 5 = 1

2 2 2

И интервал варьирования

*C* = *C* 1 *+C 1*− = 42 30 = 36

10 2 2

*C* = *C*2 *+C 2*− = 7 5 = 6,0 *pH*

20 2 2

|  |  |  |  |
| --- | --- | --- | --- |
| i | X1i | X2i | X1iX2i |
| 1 | - | - | + |
| 2 | - | + | - |
| 3 | + | - | - |
| 4 | + | + | + |
| Ci0 | 36 | 6 |  |
| Λi | 6 | 1 |  |

Здесь столбцы 2 – 3 отражают влияние отдельных факторов. Столбец 4 отражает меж- факторное взаимодействие.

Заполнение второго и третьего столбца в пояснениях не нуждается. Четвертый столбец заполняется по правилу перемножения содержимого второго и третьего столбцов: Если вто- рой и третий столбец имеют одинаковый знак, то в четвертом столбце ставится +, в против- ном случае -.

План полного двухфакторного эксперимента ПФЭ22 дает возможность вычислить че- тыре коэффициента уравнения регрессии:

*y*  *a*1  *a*2 *x*1  *a*3 *x*2  *a*4 *x*1 *x*2

План полного трехфакторного эксперимента дает возможность вычислить восемь коэффициентов уравнения регрессии

*y=a 1 +a 2 x*1 *+a 3 x*2 *+a 4 x*3 *+a 5 x*1 *x*2 *+a6 x*1 *x*3 *+a 7 x*2 *x*3 *+a 8 x*1 *x*2 *x*3

и т.д.

ЗАДАЧА. Методом наименьших квадратов рассчитать в Маткаде коэффициенты урав- нения по результатам выполнения представленного в таблице полного двухуровневого трехфакторного плана ПФЭ 23. .

Рассчитывается восемь коэффициентов для уравнения:

*y=a 1 +a 2 x*1 *+a 3 x*2 *+a 4 x*3 *+a 5 x*1 *x*2 *+a6 x* 2 *x*3 *+a7 x*1 *x*3 *+a 8 x*1 *x*2 *x*3

Таблица 1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I | Основные столбцы | | | Вспомогательные столбцы | | | |  | | |
| X1i | X2i | X3i | X1iX2i | X2iX3i | X1iX3i | X1iX2iX3i | Y1i | Y2i | Y3i |
| 1 | - | - | - | + | + | + | - | 73 | 69 | 68 |
| 2 | - | + | - | - | - | + | + | 58 | 58 | 64 |
| 3 | + | - | - | - | + | - | + | 54 | 59 | 52 |
| 4 | + | + | - | + | - | - | - | 84 | 94 | 92 |
| 5 | - | - | + | + | - | - | + | 100 | 106 | 109 |
| 6 | - | + | + | - | + | - | - | 98 | 90 | 97 |
| 7 | + | - | + | - | - | + | \_ | 77 | 85 | 78 |
| 8 | + | + | + | + | + | + | + | 105 | 95 | 100 |

Здесь У – выход процесса. Произведено три повторности эксперимента.

1. В начале программы набираем ORIGIN 1, что означает, что счет всех элементов дол- жен начинаться с единицы.
2. По таблице создаем матрицу выходов У и матрицу входов Х:

 73

 58



 54

 84

Y  

 100

 98

 77



 105

69

58

59

94

106

90

85

95

68 

64 

52 

92 



109 

97 

78 

100 

 1 1

 1 1



 1 1

 1 1

X   1 1



 1 1

 1 1



 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

1 

1 

1 

1 



1 

1 

1 

1 

1. Вычисляем среднее

MYi  meanY  Y

 Y 

выходов: i  1 8

MYT  ( 70 60 55 90 105 95 80 100)

i 1

60

i 2

i 3

1. Задаем начальные приближения для коэффициентов:

ai  1

aT  ( 1 1 1 1 1 1 1 1 )

1. Введя оператор **given** , формируем вычислительный блок. 6.Набираем выражение для метода наименьших квадратов:

8

 MY  a  a  X  a  X  a  X  a  X  a  X  a  X  a  X 2 0

i  1

i 1 2

i 1

3 i 2

4 i 3

5 i 4

6 i 5

7 i 6

8 i 7

7. Решение ищем с помощью встроенной функции MINERR ( Минимальная ошибка): и получаем ответ

a  Minerr(a)

aT  ( 81.875 0.625 4.375 13.125 9.375 1.875 4.375 1.875)

Записываем уравнение и сравниваем сначала визуально модель с объектом моделиро- вания.

Q  a  a  X

 a  X  a  X

 a  X  a  X

 a  X  a  X

i 1 2

i 1 3 i 2

4 i 3 5 i 4

6 i 5 7 i 6

8 i 7

1. Вычисляем абсолютные W и относительные в процентах w ошибки модели

MYT  ( 70 60 55 90 105 95 80 100)

w 

i

QT  ( 70 60 55 90 105 95 80 100)

W 100

i

Wi  MYi  Q

i max(MY)

max( w)  1.002

max(W)  1.052  6

 6

10

  7  10

 4.616  7 



10

 4.397 10



 7 

 1.854

 7 

1.765 10

 10    

   6   1.052 10 6 

 1.104 10    

  7   7.272 10 7 

7.636 10

W   

w    

   8 

 4.913 10 8 

5.159 10

  

  7 



 6.653 10 7 

 6.986 10 

  

  7 



 6.448 10 7 

 6.771 10   

  6

 1.002 10 6 

 1.052 10 

Проводим проверку значимости коэффициентов по Стьюденту:

Для этого:

1. вычисляем выборочную и дисперсию и с.к.о.

8 3

  Y

* + MY 2

D 

i  1

j  1

i j i

D  0.75

S  D

S  0.866

8238

1. Вычисляем обратную функцию распределения Стьюдента для уровня значимости 0,95 и числа степеней свободы =16 Если в прямой функции распределения, задавшись значе- нием случайной величины и числом степеней свободы, мы вычисляем вероятность, то в об-

ратной функции, задавшись вероятностью и числом степеней свободы, мы получаем значе- ние случайной величины.

Для Стьюдента, как известно, число степеней свободы равно удвоенному числу экспе- риментов.

qt(0.95 16)  1.746

1. Вычисляем доверительную ошибку ε и сравниваем ее с модулями значений коэф- фициентов.

qt(0.95 16)  1.746

  qt(0.95 16)S

  1.512

aT  ( 81.875 0.625

4.375 13.125 9.375

1.875

4.375

1.875)

Мы видим, что второй

коэффициент меньше доверительной ошибки, следовательно, он не значим.

1. Записываем уравнение регрессии без учета второго коэффициента:

Q1  a  a  X

 a  X  a  X

 a  X  a  X

 a  X

i 1 3

i 2 4 i 3

5 i 4 6 i 5

7 i 6 8 i 7

Проверка адекватности по Фишеру

1. Вычисляем функцию F.

 8  Q  MY

2

SAD

SAD  

i i 

F 

 12

 1 

D SAD  4.105 10

i  1 

F  5.473

 12

10

1. Вычисляем обратную функцию распределения Фишера:

qF(0.95 16 1)  246.464

1. Так как Вычисленное F < qF, то с вероятностью 0,95 полученное уравнение адек- ватно экспериментальным данным.

ЛАБОРАТОРНАЯ РАБОТА №18. АДАПТИВНЫЙ ФИЛЬТР КАЛМАНА

Основной проблемой, возникающей при любых измерениях, является их недоста- точная точность. Имеется два пути решения этой проблемы: повышение точности изме- рительных приборов и повышение точности путем статистической обработки избыточно- го числа измерений, в результате которой получают оценку измеряемой величины. Повы- шение точности измерительных приборов требует существенных затрат, в то время как статистическая обработка измерений при наличии компьютера стоит дешево и проводит- ся достаточно быстро.

Сегодня существует множество методов статистической обработки измерений. Их можно разделить на два больших класса: просто статистической обработки и опти- мальной обработки, в результате которой получают в каком- то смысле оптимальную оценку измеряемой величины. Кроме того, обычно разделяют методы статистической обработки для статических и динамических систем. Статистическая обработка статиче- ских систем изучалась на 4-м курсе. Это метод наименьших квадратов. Мы ниже рас- смотрим несколько методов оптимальной статистической обработки информации дина- мических систем на конкретных примерах.

Сложность обработки информации в динамической системе заключается в том, что ее состояние ме- няется во времени и отдельные измерения переводить из одного состояния в другое.

Существует также несколько методов оптимальной статистической обработки для динамических систем. Это фильтры Винера, Винера- Хопфа и др.

В 1961 году американский математик КАЛМАН разработал оптимальный фильтр для линейных нестационарных систем. Преимуществом фильтра Калмана является то, что он решает задачу во временной , а не частотной области, как , например, фильтр Винера.

Калман разработал свой фильтр для многомерных задач Формулировка задачи , как правило, задается в матричной форме, однако мы рассмотрим ниже простейшие скалярные задачи. в матричной форме

Фильтр Калмана существует в непрерывной и в дискретных формах. Разберем дис- кретный фильтр Калмана.

Рассмотрим следующую задачу

Блок- схема исследуемого процесса имеет вид:

f(t)



w(t)

Ф(t)



v(t

z(t)

Н(t)

y(t)

x(t)

Рис.1. Блок – схема объекта.

. Здесь Ф – линейный объект( т.е. объект описываемый линейными соотношениями), на выходе которого вырабатывается неслучайный сигнал y(t). На выходную величину воз- действует аддитивный (в виде слагаемого ) шум w(t). Шум w(t) – это какие –то случайности внутри нашей системы. После шума мы имеем выходную величину x(t). X(t) – выходной сигнал нашей системы есть сумма неслучайного процесса y(t) и шума. Поэтому это уже случайный процесс. Поэтому в целом наша система – вероятностная ( стохастическая). Мы хотим определить неизвестный нам случайный выходной процесс y(t) измерить неслучайную выходную величину y(t), но измеряем мы не ее, и не случайный выходной сигнал системы x(t), а преобразованный линейным блоком Н случайный процесс z(t). Это так называемые косвенные измерения, когда меряется какая – то функция определяемой величины. В общем случае Ф и Н могут изменяться во времени.

Шумы W(t) и v(t) – нормально распределенные случайные процессы с нулевыми математическими ожиданиями,

*M(w)=* 0

M( v) =0 и с постоянными дисперсиями Q и R, соответственно.

Процесс x(t) также распределен нормально , т.к. любое линейное преобразование нор- мально распределенного процесса есть нормально распределенный случайный процесс, а шум w(t) подчиняется нормальному закону распределения. Измерения производятся с ошибкой v (t) .

Таким образом, в действительности мы измеряем величину

*z(t)= H(x(t)+W(t))+ v(t)*

, (1)

причем случайный процесс z ( t) также распределен нормально.

оценку

Требуется построить фильтр, после которого мы получим наилучшую

*y*ˆ величины y.

Каждое последующее состояние системы определяется предыдущим: *xk+* 1 *=x k e*

– *∆t*

*T* .

Нижеприведенная программа составлена для Q=R=0.1. Для оценки качества фильтра- ции в программе решается также уравнение незашумленного, якобы «неизвестного» нам процесса.

Программа решения данной задачи в Маткаде приведена на рис.2 Здесь: y-«незашумленная» переменная,

X1- оцениваемая переменная:

Z 

y  0

x1  0

Q  0.1

R  0.1

z  dnormrnd2 P  QR

Q  R

x1  Qz  Rx1

Q  R

T  5

Q y

Q  dnormrnd2

R 0 R

Z- измеряемая переменная.

В доцикловой части первым опе- ратором присваивается начальное зна- чение «незашумленной» и оценивае- мой переменной, причем начальное значение оцениваемой переменной бе- рется «с потолка», вводятся постоян- ные значения дисперсий обоих шумов и постоянной времени системы, про- изводится первое измерение z и вы- числяется первая оценка мат. ожида-

for



k  1 100

 1 

 1 

 ния и дисперсии.

 Измерение рассматривается как

 x1  1  exp

  T

 2

* x1exp

  T 

 сумма двух псевдослучайных величин,

 каждая из которых образуется с по-



 P  P 



P

P  Q  R

 мощью двух функций:

 Функция r n d (x) возвращает

 y  1  exp 1   yexp 1 

 равномерно распределенное случайное

  T 

 z  dnormrnd2



 x1  (Pz  Rx1)



 P  R

Q y

 T 

Q  dnormrnd2

R 0



R 









число в диапазоне 0 –x.

Функция d n o r m ( x,y,w) воз- вращает нормально распределенное псевдослучайное число для аргумента x, с математическим ожиданием y и

 P  P



  z

Z



k

R

P  R

 с.к.о z.

 В первой функции dnorm за мат.

 ожидание принято значение y – «не-

 зашумленной» переменной, во второй

 Y

 k

 X1

 y

 x1

 – мат. ожидание равно 0.

 В цикловой части программы



 k 

 k 

Z

производится перевод оцениваемых переменной и дисперсии в новое со- стояние системы, затем переводится в новое состояние «незашумленная» пе

Рис.2. Программа фильтрации фильтром Калмана

X1 j

Yj Z j



2.5

2.22

1.94

1.67

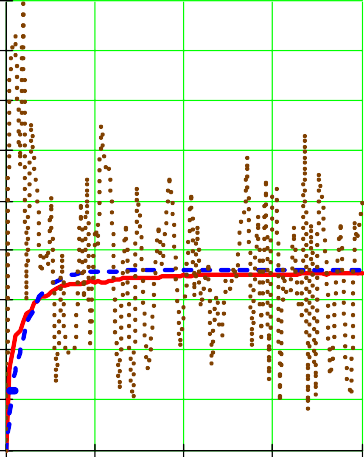
1.39

1.11

0.83

j  0 100

ременная , производится новое измерение z и вы- числяется новая оценка оцениваемой переменной и дисперсии. В последних операторах цикловой части производится накапливание измерений, оцениваемой и

«незашумленной» переменной в массивы, с це- лью вывода их на печать после окончания решения

На рис.3. приведены решение задачи. Кривая, изо- браженная штрихом – «незашумленная» переменная, непрерывная кривая – оценка, точечная кривая – изме- рения без фильтрации.

0.56

0.28

0 25 50 75 100

j

Рис.3. Графики кривых

око

ЛАБОРАТОРНАЯ РАБОТА № 19. ПОДБОР ПИД РЕГУЛЯТОРА

Система автоматического управления задана своей структурной схемой

(рис.1) и данными, приведенными в таблицах.

Необходимо: Задавшись данными из таблиц 1,2, стабилизировать движение системы

+



X

-

3

1

2

+

Y

4

5

+

σ

ло выбранной траектории, подобрав коэффициенты ПИД - регулятора.

Данные для выбора ПИД – регулятора:

Здесь звенья 1 - 4 описывают объект управления, а звено 5 - регулятор.

По исходным данным отдельных звеньев, сведенным в таблицу, следует составить уравнение системы и провести подготовку ее имитационного моделирования на ЭВМ в па- кете МАТКАД. Само моделирование будет проведено на лабораторных занятиях.

Цель моделирования - подбор закона управления в звене 5, подбор ПИД- регулятора. 1.Звено 1 - линейное динамическое и описывается дифференциальным уравнением 2-го

порядка:

T12 dy1 2/ dt2 +1T1dy1/dt +y1 = k1

1. Звено 2 - линейное статическое, зависимость выхода от входа **y 2 (y1)** задана табли- цей. Для ввода в ЭВМ необходимо найти аналитическую функциональную зависимость **y2(y1)** , проведя аппроксимацию.**у2=а2у1**
2. Звено - также статическое, зависимость выхода от входа **y 3 (y 1)** также задана табли- цей. Однако, из-за неточных измерений аппроксимацию следует проводить линейной регрес- сией, вычислив коэффициенты уравнения **y 3= a3 y 1** .
3. Сигналы **y 2** и **y3** складываются в сигнал  **=y2 +y3.**

5 .Звено 4 - линейное динамическое и описывается передаточной функцией

T2dy/ dt +y = k 2  .

6.Структура закона управления задана

y5 = 1 y +  2 dy /dt + 3 y dt

Составим уравнение всей системы. Для этого: Выпишем все уравнения

*T* 2 *y* ``*T y* ` *y*  *k* 

1 1 1 1 1 1

*y*2  *a*2 *y*1 *y*3  *a*3 *y*1

  *y*2  *y*3 *T*2 *y*` *y*  *k*2

*t*

 *m*1 *y*  *m*2 *y*`*m*3  *ydt*

0

Так как мы будем решать задачу при нулевом входном сигнале, то ∆= -y5, что позволя- ет объединить уравнения 1 и 6:

*t*

1 1 1 1 1 1 1 2 3 

*T* 2 *y* ``*T y* ` *y*  *k* (*m y*  *m y*`*m ydt*

0

Объединим уравнения 2-4. Получим

  (*a*2  *a*3 ) *y*1

Запишем уравнения 1 и 5 в операторной форме.

*T* 2 *p*2  *T p* 1) *y*  *k* (*m*  *m p*  *m* 1 ) *y*

1 1 1 1 1 2 3 *p*

(*T*2 *p* 1) *y*  *k*2 (*a*2  *a*3 ) *y*1

Чтобы освободиться от интеграла продифференцируем первое уравнение

(*T* 2 *p*3  *T p*2  *p*) *y*  *k* (*m p*  *m p*2  *m* ) *y*

1 1 1 1 1 2 3

Разрешим второе уравнение относительно у1:

*y*  (*T*2 *p* 1) *y*

1 *k* (*a*  *a* )

ли:

2 2 3

Подставим теперь это выражение в уравнение, которое мы продифференцирова-

(*T* 2 *p*3  *T p*2  *p*) 2

1 1 (*T*2 *p* 1) *y*  *k* 1(*m*1 *p*  *m*2 *p k*2 (*a*2  *a*3 )

* *m*3 ) *y*

Освободимся от знаменателя и сделаем приведение подобных:

(*T* 2 *p*3  *T p*2  *p*)(*T p* 1) *y*  *k k* (*m p*  *m p*2  *m* )(*a*  *a* ) *y*

1 1 2 1 2 1 2 3 2 3

[(*T* 2 *p*3  *T p*2  *p*)(*T p* 1)  *k k* (*m p*  *m p*2  *m* )(*a*  *a* )]*y*  0

1 1 2 1 2 1 2 3 2 3

{*T* 2*T p*4  (*TT*  *T* 2) ) *p*3 [*T*  *T*  *k k m* (*a*  *a* )] *p*2  [1 *k k m* (*a*  *a* )] *p*  *k k m* (*a*  *a* )}*y*  0

1 2 1 2 1 2 1 1 2 2 2 3 1 2 1 2 3 1 2 3 2 3

Перейдем во временную область

2 *d* 4 *y* 2) *d* 3 *y d* 2 *y dy*

*T*1 *T*2

*dt* 4

 (*T*1*T*2  *T*1

) *dt*3

[*T*2  *T*1  *k*1*k*2*m*2 (*a*2  *a*3 )] *dt* 2

[1 *k*1*k*2*m*1 (*a*2  *a*3 )] *dt*  *k*1*k*2*m*3 (*a*2  *a*3 ) *y*  0

Мы получили однородное дифференциальное уравнение четвертого порядка, ко- торое и будем моделировать

Решение задачи в Маткаде.

1. Пусть по таблице 1 выбрано **k 1= 2 , k 2=1,5, T1 = 5, T2= 4,**  **1= 0,4 ,** а из таблицы

2 -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **y1** | **0** | **1** | **2** | **3** |
| **y2** | **25** | **47** |  | **-** |
| **y3** | 1**1**. **4** | **46** | **45** | **70** |

1. По данным таблицы составим векторы vx2,vy2,vx3,vy3.Используя встроенную функцию line определим коэффициенты a2,b2,a3,b3. (см.рис.1)

vy2 

 28 

 17 

yvx2 

2

 0 

 1 

a2 

slope(vx2 vy2)

a2  11

 0 

 1

 14 

a3 

slope(vx3 vy3)

a3  16.7

vx3   

 2

vy3 

 46

#####  

 3 

 45

 70 

После вводим значения остальных констант, полагая пока все коэффициенты регулято- ра m1,m2,m3, равными нулю.

T1 

T2  4

s 

0.4

k1 

2k2 

1.5

m1  0

m2  0

m3  0

Затем решаем дифференциальное уравнение, используя функ-

цию odesolve и строим график.

Give

T1T2

d4 dt4

y(t)  (sT1T2  T1)

d3 dt3

y(t) 

d2 dt2

y(t) [T2  sT1  k1 k2 m2(a2  a3)]  [1  k1 k2 m1(a2  a3)] d

dt

y(t)  k1 k2 m3(a2  a3) y(t) 0

tk 

50y'''(0) 0

y(0) 0

y''(0) 0

y 

Odesolve(t  tk)

y' ( 0 ) 1

t 

0  tk

8

6

y( t) 4

2

0 10 20 30 40 50

t

Из графика видно, что система без регулятора устойчива , но имеет большую статиче- скую ошибку. Была проведена работа по подбору коэффициентов регулятора. В результате получено решение, показанное ниже.

m1  0.5

3

m2 

0.9

m3 

0.03

y( t)

2

1

0 10 20 30 40 50

t

Показаны значения подобранных коэффициентов и график переходного процесса. Порядок подбора рекомен- дуется следующий:

Сначала подбирается коэффициент пропорциональ- ного члена , в нашем случае –m1 , затем управление по скорости - демпфер, в нашем случае m2 , и в последнюю очередь -коэффициент интеграла –m3.

Следует иметь в виду , что большие значения коэффициентов m1 **, m3** могут сделать систему неустойчивой.

2007

ЛИТЕРАТУРА

1. Очков В.Ф. Mathcad 14 для студентов, инженеров и конструкторов БХВ - Петербург
2. Очков В.Ф. Мультимедийный обучающий курс по Mathcad 13. Курс создан на фирме

Мультимедиа Технологии – (495) 673-76-92, [www.mmt-dl.ru](http://www.mmt-dl.ru/)

1. Интернет- форум exponenta.ru//Mathcad
2. Демидович Б.П., Марон И.А. Основы вычислительной математики.

М.,»Наука»,Главная редакция физико-математической литературы.,1970

Турчак Л.И. Основы численных методов. М.,»Наука»,Главная редакция физико- математической литературы.,1987

1. Беллман Р. Динамическое программирование. Издательство иностранной литерату- ры,М.1960
2. Краснов А.Е. и др. Информационные технологии пищевых производств в условиях неопределенности. М.,2001
3. Краснов А.Е. и др. Основы математического моделирования рецептурных смесей пищевой биотехнологии. М., Пищепромиздат ,2006
4. Рутковский Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алго- ритмы и нечеткие системы. М., Горячая линия – Телеком,2007
5. Грачев Ю.П. Математические методы планирования эксперимента. М., «Пищевая промышленность», 1979.
6. Дэвис М.Х.А. Линейное оценивание и стохастическое управление. М.,»Наука», Главная редакция физико-математической литературы.,1984
7. Ли Роберт. Оптимальные оценки, определение характеристик и управление.

М.,»Наука», Главная редакция физико-математической литературы.,1966