МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-ХОРАЗМИЙ

Факультет "Компьютерный инжиниринг"
Кафедра "Компьютерные системы"

Методические указания

по выполнению лабораторных работ по курсу

«Интеллектуальный анализ данных»

для бакалавров обучающих по направлению

5330500 – Компьютерный инжиниринг

Ташкент 2018

Авторы: д.т.н., профессор, Усманов Р.Н., Кутлымуратов А.Ж, Хабирова Д.Н. /ТУИТ, 56 с., Ташкент, 2018

В условиях развития и реформирования экономики страны стратегической целью государства является подготовка свободно мыслящих, высококвалифицированных молодых кадров. Для достижения поставленной цели требуется разработка и издание учебно-методической литературы на основе поставленных задач «Национальной программы по подготовке кадров» .

Цель данного методического указания является определение основных требований по выполнению лабораторных работ по курсу «Интеллектуальный анализ данных».

Напечатано по решению науно-методического совета ТУИТ. (**протокол №__ от_____201__ г.)**

Рецензенты:

заведуючий лаборатории
«Геоинформационных технологий»,
ГП «Институт ГИДРОИНГЕО»
к.т.н. Хушвактов С.Х.

заведующая кафедрой
«Мультимедийные технологии»
ТУИТ имени Мухаммада ал-Хоразмий,
к.т.н., доцент Назирова Э.Ш.

Ташкентский университет информационных технологий имени Мухаммада ал-Хоразмий. 2018

Введение

Методические указания представляют собой указания для получения базовых знаний по анализов данных на основе математического моделирования в среде Matlab. Здесь приведены основные характеристики для вычисления и анализа матричных данных. Рассматриваются фунции аппроксимации и интерполяции данных, линейной и нелинейной регрессионный анализ данных, корреляционной анализ и их построение в среде Matlab. А также, описываются понятие кластеризация данных на основе классификации и алгоритмы кластеризации. Рассматриваются теория нечётких множеств и свойства построения на основе пакета FLT в среде Matlab.

Лабораторная работа №1

Тема: Изучение основных характеристик матриц в среде Matlab

Цель работы: Изучить основных вычислительных характеристик матриц в Matlab

Теоретические сведения

Основные характеристики матрицы:

- детерминант;
- ранг матрицы;
- норма;
- базис ортонормала;

Определитель матрицы, детерминант — число, соответствующее *квадратной матрице* и полученное путем ее преобразования по определенному правилу. Обычное обозначение (для матрицы A) — detA. Например, определитель (второго порядка) матрицы обозначается и вычисляется следующим образом: $detA = a_{11}a_{22} — a_{12}a_{21}$.

В общем случае (квадратной матрице порядка n) из элементов матрицы A сначала составляют все возможные произведения из n сомножителей каждое, содержащие по одному элементу из каждой строки и по одному элементу из каждого столбца, затем эти произведения складываются по специальному правилу.

Определитель, в котором вычеркнуты произвольная строка, например i-я, и произвольный столбец, например j-й, называется минором. Он имеет (n — 1) -й порядок, т.е. порядок на 1 меньше, нежели исходный определитель.

Рангом системы строк (столбцов) матрицы с строк и столбцов называется максимальное число линейно независимых строк (столбцов). Несколько строк (столбцов) называются линейно независимыми, если ни одна из них не выражается линейно через другие. Ранг системы строк всегда равен рангу системы столбцов, и это число называется рангом матрицы.

Ранг матрицы — наивысший из порядков всевозможных ненулевых

миноров этой матрицы. Ранг нулевой матрицы любого размера ноль. Если все миноры второго порядка равны нулю, то ранг равен единице, и т.д.

Обычно ранг матрицы A обозначается Rank (A)

Полный список характеристики по любому из них можно получить в matlab\matfun, используя команды help matfun

Таблица 1.1Функции матричных действий

Функция и ее синтаксис	Описание
zeros(m, n)	Возвращает нулевую матрицу (состоящую из одних нулей) размерности mxn.
ones(m, n)	Возвращает матрицу состоящую из одних единиц размерности mxn.
eye(m, n)	Создает единичную матрицу размерности mxn.
rand(m, n)	Возвращает матрицу случайных чисел равномерно распределенных в диапазоне от 0 до 1, матрица имеет размерность mxn.
randn(m, n)	Возвращает матрицу размерности mxn, состоящих из случайных чисел в диапазоне от 0 до 1 и имеющих гаусовское распределение (имеющих нормальный закон распределения).
hadamard(n)	Возвращает матрицу Адамара размерности nxn.
hild(n)	Возвращает матрицу Гильберта размерности nxn.
invhild(n)	Возвращает обратную матрицу Гильберта размерности nxn.
reshape(A, n, m)	Функция образует матрицу размерности nxm путем выборки элементов заданной матрицы A по столбцам и последующему распределению этих элементов по m столбцам каждый из которых содержит n элементов, при этом матрица A должна иметь размерность nxm.
tril(A)	Выбирает нижнюю треугольную матрицу из матрицы A
triu(A)	Выбирает верхнюю треугольную матрицу из матрицы A
diag(A, n)	Функция создает диагональную матрицу из вектора A, при этом второй параметр необязателен. Если второй параметр присутствует, то создается матрица в которой вектор A помещается в другую диагональ, при этом если n>0, то вектор помещается выше главной диагонали, если n <0 то ниже, если n=0 то на главную диагональ. Если A матрица, то функция выдает вектор

	сформированный из элементов главной ее диагонали, если второй параметр отсутствует или равен нулю. Если второй параметр присутствует, то функция выдает другую диагональ, в соответствие с вторым параметром (см. предыдущий абзац).
' (апостроф)	Оператор производит транспонирование матрицы
+ - * / \ ^	Математические действия над матрицами. Применимы как к выражению вида «матрица-скаляр», так и «матрица-матрица» (за исключением возведения в степень, он применима только к выражению «матрица-скаляр»). Во всех операциях необходимо следить за размерностями матриц.
inv(A)	Возвращает обратную матрицу по отношению к матрице A
det(A)	Подсчет определителя (детерминанта) матрицы
cross(A, B)	Векторное произведение векторов
lu(A)	Производит LU-разложение матрицы A и выдает матрицы в следующем порядке [L U P] (подобное запись обязательна), при этом выполняется следующее соотношение P*A=L*U.
size(A)	Возвращает массив состоящий из числа строк (первый элемент) и числа столбцов (второй элемент).
sum(A)	Возвращает сумму всех элементов по столбцу
std(A)	Возвращает среднеквадратическое отклонение столбца матрицы
min(A) max(A)	Возвращает минимум и максимум соответственно, по столбцу матрицы
sort(A)	Сортирует столбец матрицы по возрастанию
prod(A)	Вычисляет произведение всех элементов столбцов
rank(A)	Ранг матрицы Функция r = rank(A) возвращает ранг матрицы, который определяется как количество сингулярных чисел, превышающих порог max(size(A))*norm(A)*eps.
sum(A)	Мы подсчитали вектор-строку, содержащую сумму элементов столбцов матрицы A.
fliplr	зеркально отображает матрицу слева направо.
magic.(A)	MATLAB на самом деле обладает встроенной функцией, которая создает магический квадрат почти любого размера.
Det(A)	Считает детерминант матрицы

```
>> A = [1 2 3; -3 2 5; 2 4 8]

A =

     1     2     3
    -3     2     5
     2     4     8

>> det(A)

ans =

    16
```

Рис.1.1. Детерминант матрицы

```
>> A = [1 2 -1 3 2; 2 -1 3 0 1; 3 1 2 3 3; 1 2 3 1 1]

A =

     1     2    -1     3     2
     2    -1     3     0     1
     3     1     2     3     3
     1     2     3     1     1

>> rank(A)

ans =

     3
```

Рис 1.2. Ранг матрицы

```
>> A = [1 3 7; 2 5 8; -1 4 3]

A =

     1     3     7
     2     5     8
    -1     4     3
```

```
>> A1 = norm(A, 1)

A1 =

    18
```

```
>> A2 = norm(A, 2)

A2 =

   13.0488
```

```
>> Ainf = norm(A, inf)

Ainf =

    15
```

Рис 1.3. Норма матрицы

Варианты:

№	Матрица
1.	$\begin{bmatrix} 2 & 3 & -4 & 1 \\ 1 & -3 & -1 & 2 \\ 3 & 3 & -7 & 0 \\ 1 & -12 & 1 & 5 \end{bmatrix}$
2.	$\begin{bmatrix} 1 & -3 & 2 & \\ 4 & 1 & & 3 \\ 2 & 7 & & 1 \\ 5 & -2 & 0 & 4 \end{bmatrix}$
3.	$\begin{bmatrix} 3 & -1 & 1 & -2 \\ 2 & 1 & -2 & 1 \\ 1 & -2 & 3 & -3 \\ 0 & -5 & 8 & -7 \end{bmatrix}$
4.	$\begin{bmatrix} 3 & 4 & -2 & 3 \\ 2 & 7 & -5 & -1 \\ 1 & -3 & 3 & 4 \\ 4 & 1 & 1 & 7 \end{bmatrix}$
5.	$\begin{bmatrix} 1 & 3 & 2 & -1 \\ 2 & -1 & 3 & 2 \\ 3 & -5 & 4 & 5 \\ 1 & 10 & 6 & -6 \end{bmatrix}$
6.	$\begin{bmatrix} 5 & -1 & 2 & -3 \\ 1 & -2 & 1 & -1 \\ 3 & 3 & 0 & -1 \\ 2 & 5 & -1 & 0 \end{bmatrix}$
7.	$\begin{bmatrix} 3 & 4 & 2 & -5 \\ 4 & 1 & -4 & 1 \\ 2 & 7 & 8 & -11 \\ -1 & 3 & 6 & -6 \end{bmatrix}$
8.	$\begin{bmatrix} 1 & 3 & 2 & 1 \\ 4 & -1 & 3 & -1 \\ 2 & -7 & -1 & -3 \\ 6 & 5 & 7 & 1 \end{bmatrix}$
9.	$\begin{bmatrix} 1 & 3 & 5 & 1 \\ 3 & 5 & 3 & 5 \\ 1 & -1 & -4 & 3 \\ 2 & 4 & 4 & 3 \end{bmatrix}$
10.	$\begin{bmatrix} 1 & -4 & 1 & -2 \\ 2 & 3 & -1 & 1 \\ 4 & -5 & 1 & -3 \\ 1 & -15 & 4 & -7 \end{bmatrix}$

11.	$\begin{bmatrix} 3 & -1 & 2 & -5 \\ 2 & -3 & 3 & -2 \\ 5 & -4 & 5 & -7 \\ 1 & 2 & -1 & -2 \end{bmatrix}$
12.	$\begin{bmatrix} 2 & -3 & 4 & 2 \\ 1 & -1 & 3 & -1 \\ 4 & -7 & 11 & 3 \\ 3 & -5 & 5 & 5 \end{bmatrix}$
13.	$\begin{bmatrix} 2 & 5 & 3 & -4 \\ 2 & -1 & -1 & 3 \\ 6 & -7 & -5 & -1 \\ 0 & 5 & -1 & -5 \end{bmatrix}$
14.	$\begin{bmatrix} 1 & -3 & -5 & 7 \\ 2 & -1 & -3 & 4 \\ 1 & 2 & 2 & -3 \\ 1 & -8 & -12 & -7 \end{bmatrix}$
15.	$\begin{bmatrix} 3 & -5 & -3 & 1 \\ 2 & 1 & 3 & 2 \\ 4 & -11 & -9 & 0 \\ 1 & -6 & 0 & -1 \end{bmatrix}$

Порядок выполнения работы

1. Ознакомиться с теоретической частью по приведенным выше материалам;
2. Установить прикладный программный продукт Matlab по характеристику компьютера
3. Выполните задания по каждому варианту в среде Matlab и определить детерминант, ранг, норма матрицы

Контрольные вопросы

1. Основные характеристики матрицы
2. Как найти детерминант матрицы
3. Как найти сумму всех элементов по столбцу
4. Как найти обратную матрицу
5. Как найти норму матрицы

Лабораторная работа №2

Тема: Функции аппроксимации и интерполяции данных

Цель работы: Построить аппроксимации и интерполяции данных

Теоретические сведения

1. Сущность понятия аппроксимации.
2. Интерполяция способ нахождения промежуточных значений функции.

Аппроксимация, или **приближение** — научный метод, состоящий в замене одних объектов другими, в том или ином смысле близкими к исходным, но более простыми.

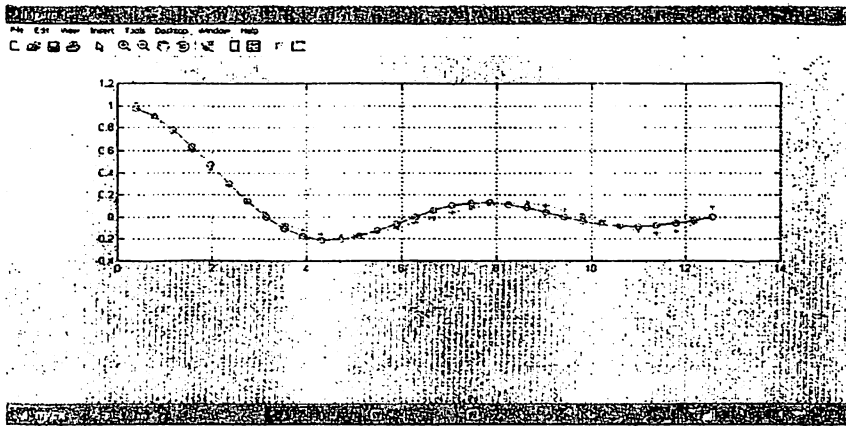
Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов (например, таких, характеристики которых легко вычисляются, или свойства которых уже известны).

Интерполяция, интерполирование - способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений.

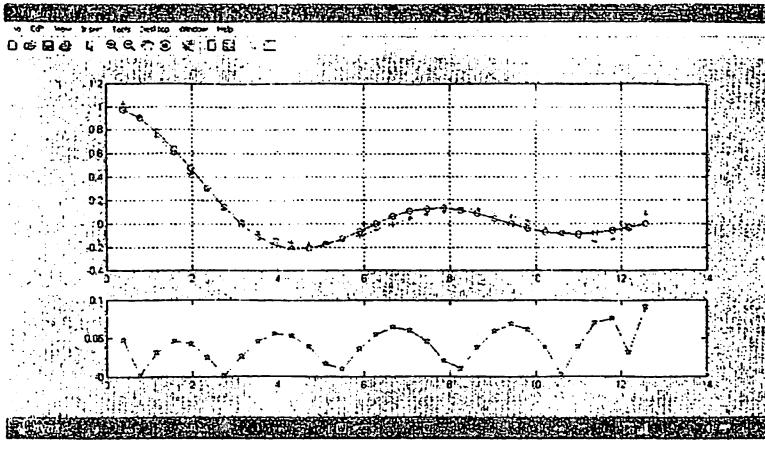
Решение задач аппроксимации

Функция $y = \frac{\sin(x)}{x}$ аппроксимировать, узловые точки распределены равномерно, использовать 5 порядок многочлена.

```
>> x=pi/8:pi/8:4*pi;  (равномерное распределение узловых точек)
>> y=sin(x)./x;
>> p=polyfit(x,y,5);
>> fa=polyval(p,x);
>> subplot(3,1,1:2), plot(x,y,'-o',x,fa,':*'), grid, hold on;
```



>> error=abs(fa-y); subplot(3,1,3), plot(x,error,'--p')



Аппроксимация на основе неравномерных узловых точек

Пример: аппроксимировать функцию $y=\frac{\sin(x)}{x}$ на промежутке [0.1;4.5] на основе 3-порядка многочлена.

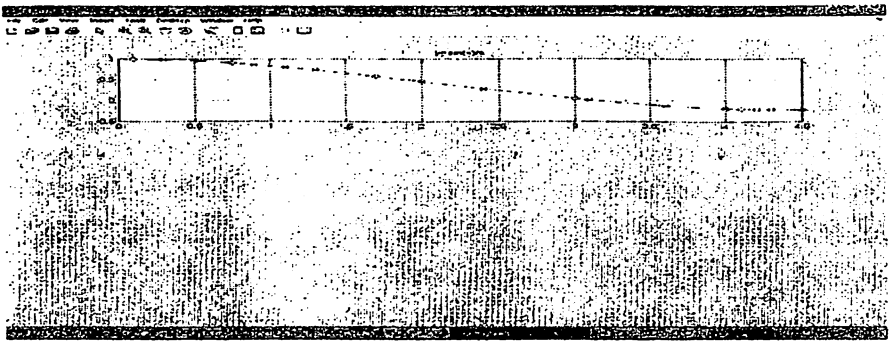
>> x=[0.1 0.3 0.5 0.75 0.9 1.1 1.3 1.7 2 2.4 3 3.1 3.6 4 4.1 4.2 4.3 4.5];

>> y=sin(x)./x;

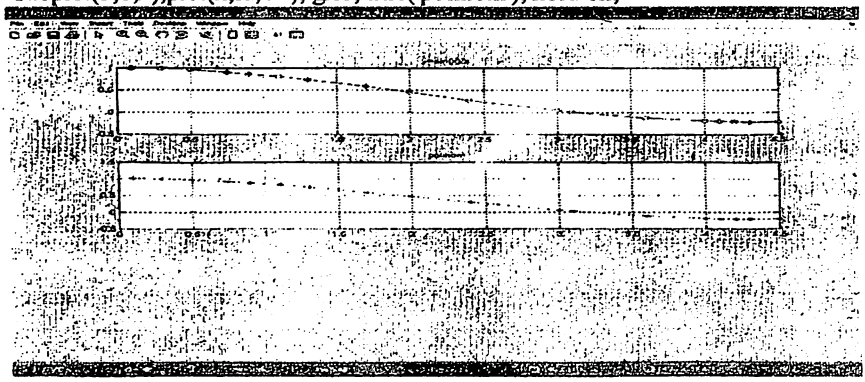
>> p=polyfit(x,y,3);

>> fa=polyval(p,x);

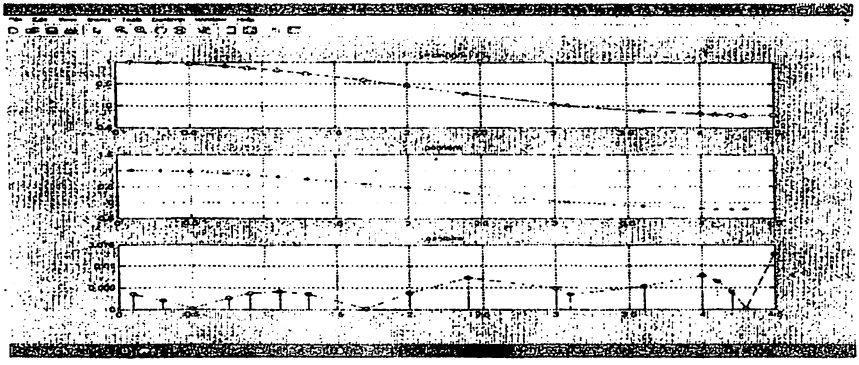
>>subplot(3,1,1),plot(x,y,'-o'), grid, title('y=sin(x)/x'), hold on;



>>subplot(3,1,2),plot(x,fa,':*'), grid, title('polinom'), hold on;



>> error=abs(fa-y);
>>subplot(3,1,3),plot(x,error,'--p'), grid, title('oshibka'), hold on;
>>stem(x,error)



Решение задач интерполяции

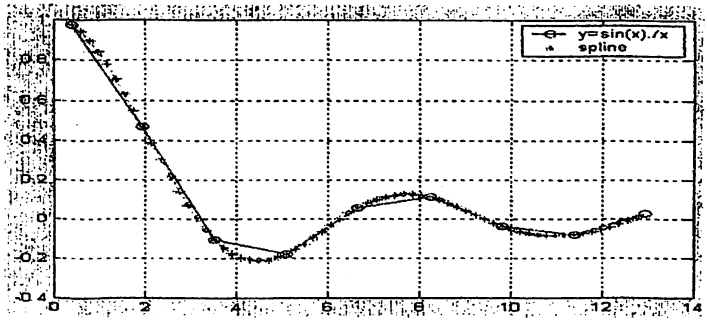
Интерполяция —способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений.

На основании этих наборов требуется построить функцию, на которую могли бы с высокой точностью попадать другие получаемые значения. Такая задача называется аппроксимацией кривой. Интерполяцией называют такую разновидность аппроксимации, при которой кривая построенной функции проходит точно через имеющиеся точки данных.

Задача интерполяции — найти данные в окрестности узловых точек. Для этого используются подходящие функции, значения которых в узловых точках совпадают с координатами этих точек.

Выполнить интерполяцию функции $y=\frac{\sin(x)}{x}$ на основе равномерного распределения кубического многочлена и кубического сплайна.

```
X=pi/8:pi/2:(4*pi+pi/2);
Y=sin(x)./x;
Xi=pi/8:pi/16:(4*pi+pi/16);
Fi1=interp1(x,y,xi,'cubic');
Plot(x,y,'-o',xi,fi1,':*'), grid, hold on
Legend('y=sin(x)./x','cubic')
Figure
Fi2=interp1(x,y,xi,'spline');
Plot(x,y,'-o',xi,fi2,':*'),grid, hold on
Legend('y=sin(x)./x','spline')
```



Варианты

№	1	2	3	4	5	6	7
x	*y*	*y*	*y*	*y*	*y*	*y*	*y*
0.25	0.778	2.284	0 247	0.552	1.031	0.444	0.255
0.31	0.758	2.363	0.285	0.615	1.048	0.530	0.320
0.36	0.717	2.433	0.362	0.667	1.066	0.645	0.376
0.39	0.677	2.477	0.390	0.740	1.107	0.771	0.411
0.43	0.650	2.537	0.416	0.642	1.194	0.640	0.458
0.47	0.625	2.100	0.352	0.587	1.233	0.538	0.508
0.52	0.644	1.982	0.339	0.543	1.138	0.477	0.572
0.56	0.661	1.851	0.331	0.589	1.061	0.508	0.626
0.64	0.717	1.896	0.397	0.684	1.021	0.564	0.544
0.66	0.714	1.935	0.513	0.709	1 122	0.578	0.476
0.71	0.691	2.034	0.651	0.771	1.256	0.610	0.559

№	8	9	10	11	12	13	14
x	*y*	*y*	*y*	*y*	*y*	*y*	*y*
0.24	0.335	1.274	0.586	0.242	1.002	0.544	0.237
0.26	0.254	1.297	0.571	0.262	1.103	0.566	0.257
0.27	0.263	1.310	0.663	0.273	1.203	0.576	0.266
0.29	0.384	1.436	0.648	0.294	1 204	0.598	0.286

0.30	0.491	1.535	0.540	0.304	1.304	0.509	0.295
0.32	0.509	1.437	0.526	0.325	1.255	0.431	0.234
0.37	0.454	1.344	0.590	0.308	1.316	0.387	0.161
0.38	0.363	1.146	0.683	0.289	1.377	0.399	0.170
0.42	0.397	1.252	0.657	0.232	1.409	0.446	0.247
0.49	0.455	1.363	0.612	0.309	1.412	0.533	0.247
0.59	0.533	1.380	0.554	0.324	1.357	0.669	0.206

Порядок выполнения работы

1. Ознакомиться с теоретической частью по приведенным выше материалам;
2. Построить аппрокрсимация данных в среде Матлаб по варианту
3. Выполните задания по каждому варианту и построить интерполяция данных в среде Матлаб

Контрольные вопросы

1. Что такое аппроксимация
2. Что такое интерполяция
3. Команды аппроксимация и интерполяции.
4. Что такое одномерная табличная интерполяция
5. Что такое двумерная табличная интерполяция
6. Что такое трехмерная табличная интерполяция

Лабораторная работа №3

Тема: Создание линейных регрессионных моделей в среде Matlab

Цель работы: Построить линейных регрессионных моделей в Matlab

Теоретические сведения

Алгоритм построения линейной регрессионной модели и её реализация на Matlab.

Пусть $X = x_{ij}$ - матрица исходных данных, $Y = y_{ij}$ - вектор зависимой переменной, $i = 1,2,.....,n$, $j = 1,2,.....,m$. Требуется построить линейную регрессионную модель вида:

$$y^{M} = Xa + e \qquad (1)$$

Здесь y^{M} - вектор зависимого переменного, X – матрица исходных данных, a – вектор коэффициентов, e – вектор переменной.

Для двухмерного случая регрессионная модель (1) представляется так:

$$y_i^{M} = a_0 + a_1 x_{i1} + a_2 x_{i2} + e_i \qquad (2)$$

$$y_i^{M} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} \end{bmatrix} [a_0 a_1 a_2] + e_i \qquad (2')$$

Для определения коэффициентов a_0, a_1, a_2 воспользуемся методом наименьших квадратов (МНК), согласно который решается следующая оптимизационная задача:

$$J(a_0 a_1 a_2) = \sum_{i=1}^{n} a_0 + a_1 x_{i1} + a_2 x_{i2} + y_i^2 \to \min \qquad (3)$$

Решение задачи (3) сводится к решению следующей системы линейных алгебраических уравнений (СЛАУ):

$$\left.\begin{array}{l}\frac{\partial J}{\partial a_0}=0\\ \frac{\partial J}{\partial a_1}=0\\ \frac{\partial J}{\partial a_2}=0\end{array}\right\}(4)\Leftrightarrow\begin{array}{c}2\sum(a_0+a_1x_{i1}+a_2x_{i2}+y_i)=0\\ 2\sum(a_0+a_1x_{i1}+a_2x_{i2}+y_i)x_{i1}=0\\ 2\sum(a_0+a_1x_{i1}+a_2x_{i2}+y_i)x_{i2}=0\end{array}$$

Отсюда:

$$\begin{array}{c}{}^{n}a_0+a_1\sum x_{i1}+a_2\sum x_{i2}=\sum y_i\\ a_0\sum x_{i1}+a_1\sum x_{i1}^{2}+a_1\sum x_{i1}{}^{*}x_{i2}=\sum x_{i1}y_i\\ a_0\sum x_{i2}+a_1\sum x_{i1}{}^{*}x_{i2}+a_2\sum x_{i2}^{2}=\sum x_{i2}y_i\end{array}\qquad(5)$$

Решаемая СЛАУ (5) находим коэффициенты уравнение линейной регрессии (a_0,a_1,a_2). В системах (4) или (5) $\Sigma=\sum_{i=1}^{n}$

Результаты регрессионного анализа целесообразно организовать в виде следующие таблицы (3.1):

Таблица 3.1.

I	x_{i1}	x_{i2}	y_i	y_i^M	$(y_i-y_i^M)^2$
1	x_{11}	x_{12}	y_1	y_1^M	$(y_1-y_1^M)^2$
2	x_{21}	x_{22}	y_2	y_2^M	$(y_2-y_2^M)^2$
...	...	...	...	...	...
N	x_{n1}	x_{n2}	y_n	y_n^M	$(y_n-y_n^M)^2$

Результаты ВЭ целесообразно представляет в графическом виде, рассматривая в одном графическом окне графику функций:

$y_i=f_1(x_{i1},x_{i2})$ - заданные значение

$y_i^M=f_1(x_{i1},x_{i2})$ - результаты ВЭ по регрессионной модели

Более простой способ определение коэффициентов множественной линейной регрессии основывается на следующем алгоритме:

$Y = X * a$ - искомая уравнение множественной регрессии. Умножив данное матричное уравнение с обеих сторон на X^T имеем:

$$X^T Y = X^T * X * a$$

отсюда, вектор коэффициентов линейной многофакторной регрессии определяется так:

$$a = (X^T Y) * (X^T x)^{-1} \qquad (6)$$

Приводим данный алгоритм на конкретном примере. Пусть дана:

$$A = \begin{bmatrix} i & y_i & x_{i1} & x_{i2} \\ 1 & 10 & 2 & 1 \\ 2 & 12 & 2 & 2 \\ 3 & 17 & 8 & 10 \\ 4 & 13 & 2 & 4 \\ 5 & 15 & 6 & 8 \\ 6 & 10 & 3 & 4 \\ 7 & 14 & 5 & 7 \\ 8 & 12 & 3 & 3 \\ 9 & 15 & 9 & 10 \\ 10 & 12 & 10 & 11 \end{bmatrix} \qquad y = A(:,2) \qquad X = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 8 & 10 \\ 1 & 2 & 4 \\ 1 & 6 & 8 \\ 1 & 3 & 4 \\ 1 & 5 & 7 \\ 1 & 3 & 3 \\ 1 & 9 & 10 \\ 1 & 10 & 11 \end{bmatrix}$$

В общем случае:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \end{bmatrix} * \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} \end{bmatrix} = \begin{bmatrix} n & \sum x_{i1} & \sum x_{i2} \\ \sum x_{i1} & \sum x_{i1}^2 & \sum x_{i1} x_{i2} \\ \sum x_{i2} & \sum x_{i1} x_{i2} & \sum x_{i2} \end{bmatrix}$$

$$X^T Y = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_{i1} y_i \\ \sum x_{i2} y_i \end{bmatrix}$$

Результаты расчетов:

$$X^T X = \begin{matrix} 10 & 50 & 60 \\ 50 & 336 & 398 \\ 60 & 398 & 480 \end{matrix}$$

$$(X^T X)^{-1} = \begin{matrix} 0.4018 & -0.01676 & -0.03631 \\ -0.01676 & -0.16760 & -0.13687 \\ -0.03631 & -0.13687 & 0.12011 \end{matrix}$$

$$X^T Y = (137,756,908)^T$$

$$a = (X^T X)^{-1} * (X^T Y) = (0.3872, 0.1285, 0.6117)^T$$

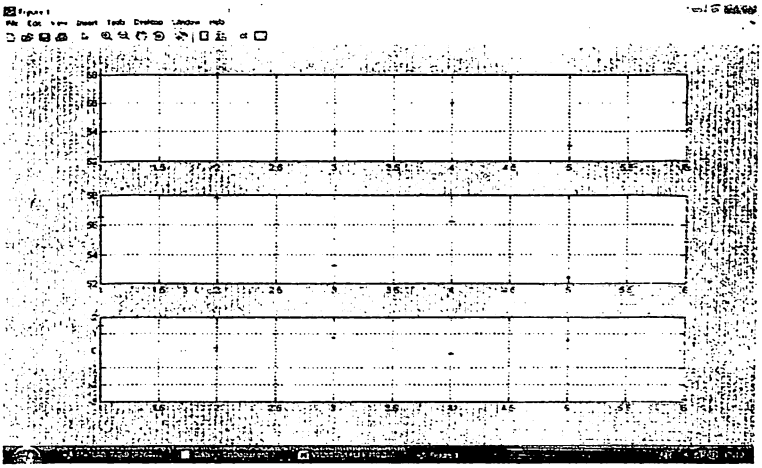
$$y_i^M = 9.3872 + 0.1285 x_{i1} + 0.6117 x_{i2}$$

Построение линейной регрессионной модели в среде Матлаб

```
>> X=[19 17 58 26; 18 16 58 24; 15 17 54 28; 16 16 56 26; 14 17 53 25; 17 17 52 25
]
X =
   19   17   58   26
   18   16   58   24
   15   17   54   28
   16   16   56   26
   14   17   53   25
   17   17   52   25
>>x1=X(:,1);
>>x1=X(:,1)
x1 =
   19
   18
   15
   16
   14
   17
>>x2=X(:,2)
x2 =
   17
   16
   17
   16
   17
   17
>> y=X(:,3)
```

```
y =
   58
   58
   54
   56
   53
   52
>> XX=[1 1 1 1 1 1]'
XX =
    1
    1
    1
    1
    1
    1
>> AX=[XX x1 x2]
AX =
    1   19   17
    1   18   16
    1   15   17
    1   16   16
    1   14   17
    1   17   17
>> a=inv(AX'*AX)*(AX'*y)
a =
   77.1940
    0.8209
   -2.1343
>>Ym=a(1)+a(2)*x1+a(3)*x2
Ym =
   56.5075
   57.8209
   53.2239
   56.1791
   52.4030
   54.8657
>> tab1=[x1 x2 y Ym y-Ym]

tab1 =

   19.0000   17.0000   58.0000   56.5075    1.4925
   18.0000   16.0000   58.0000   57.8209    0.1791
   15.0000   17.0000   54.0000   53.2239    0.7761
   16.0000   16.0000   56.0000   56.1791   -0.1791
   14.0000   17.0000   53.0000   52.4030    0.5970
```

```
  17.0000  17.0000  52.0000  54.8657  -2.8657
>>subplot(3,1,1),plot(y,'*'),grid
>>subplot(3,1,2),plot(Ym,'*'),grid
>>subplot(3,1,3),plot(y-Ym,'*'),grid
```



Порядок выполнения работы

1. Ознакомиться с теоретической частью по приведенным выше материалам;
2. Освоить алгоритмы для линейный регрессионный анализ в среде Матлаб
3. Построить линейной регрессионной модели в среде Матлаб

Контрольные вопросы

1. Что такое регрессионный анализ данных
2. Что такое линейный регрессионный анализ данных
3. Алгоритмы для выполнение линейный регрессионный анализ в среде Матлаб

Лабораторная работа №4

Тема: Создание нелинейных регрессионных моделей в среде Matlab

Цель работы: Построить нелинейных регрессионных моделей в среде Matlab

Теоретические сведения

Связь между аргументом и функцией может иметь нелинейный характер. Аппроксимация кривой выполняется тем же путем с использованием метода наименьших квадратов, что и в случае прямой линии . Линия регрессии должна удовлетворять условию минимума суммы квадратов расстояний до каждой точки корреляционного поля. В данном случае в уравнении (1) y представляет собой расчетное значение функции, определенное при помощи уравнения выбранной криволинейной связи по фактическим значениям x_j.

В общем случае нелинейной зависимости

$$y'(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k .$$

По методу наименьших квадратов находим частные производные по коэффициентам регрессии и приравниваем их к нулю. Получаем систему уравнений:

$$\begin{cases} \dfrac{\partial S^2}{\partial a_0} = -2\sum\limits_{j=1}^{m}(y_j - a_0 - a_1 x_j - a_2 x_j^2 - \ldots - a_k x_j^k) = 0; \\ \dfrac{\partial S^2}{\partial a_1} = -2\sum\limits_{j=1}^{m}(y_j - a_0 - a_1 x_j - a_2 x_j^2 - \ldots - a_k x_j^k) x_j = 0; \\ \ldots \\ \dfrac{\partial S^2}{\partial a_k} = -2\sum\limits_{j=1}^{m}(y_j - a_0 - a_1 x_j - a_2 x_j^2 - \ldots - a_k x_j^k) x_j^k = 0. \end{cases}$$

Оценкой тесноты связи при криволинейной зависимости служит теоретическое корреляционное отношение η_{xy} , представляющее собой корень квадратный из соотношения двух дисперсий: среднего квадрата σ_p^2 отклонений расчетных значений y'_j функции по найденному уравнению регрессии от среднеарифметического значения Y величины y к среднему квадрату отклонений σ_y^2 фактических значений функции y_j от ее среднеарифметического значения :

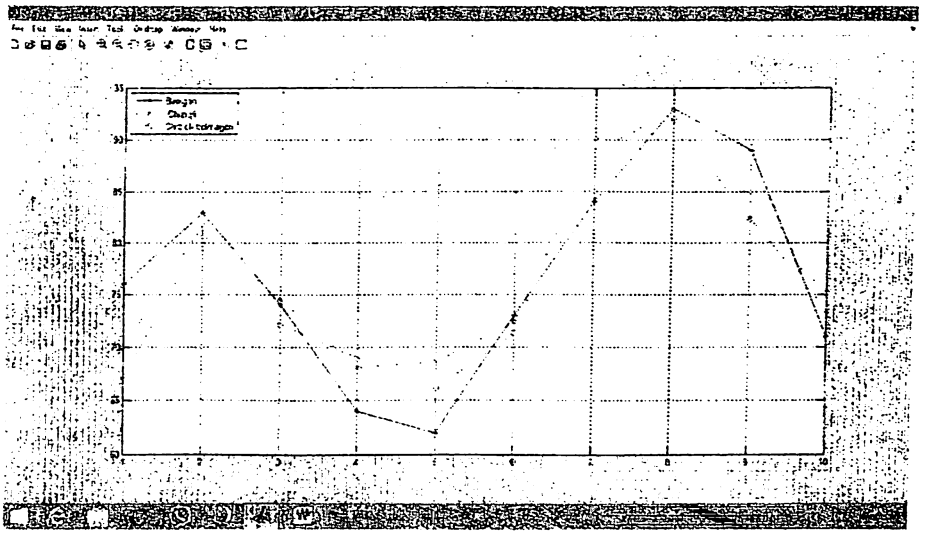
$$\eta_{xy} = \sqrt{\frac{\sigma_p^2}{\sigma_y^2}} = \sqrt{\frac{\sum_{i=1}^{m}(y_i' - \bar{Y})^2}{\sum_{i=1}^{m}(y - \bar{Y})^2}}$$

```
>> x=[1 20 25; 1 22 33; 1 25 28; 1 22 25; 1 17 27; 1 24 27; 1 25 37; 1 29 38; 1 22 34;
1 23 28]
x =
    1   20   25
    1   22   33
    1   25   28
    1   22   25
    1   17   27
    1   24   27
    1   25   37
    1   29   38
    1   22   34
    1   23   28

>> y=[76 83 74 64 62 73 84 93 89 71]'
y =
   76
   83
   74
   64
   62
   73
   84
   93
   89
   71

>> a=inv(x'*x)*(x'*y)
a =
  16.2555
   0.5375
   1.6005

>> x1=x(:,2)
x1 =
   20
   22
   25
```

```
   22
   17
   24
   25
   29
   22
   23

>> x2=x(:,3)
x2 =
   25
   33
   28
   25
   27
   27
   37
   38
   34
   28
>> F=16.2555+0.5375*x1+1.6005*x2
F =
   67.0180
   80.8970
   74.5070
   68.0930
   68.6065
   72.3690
   88.9115
   92.6620
   82.4975
   73.4320

>> z1=x1;

>> z2=x2;

>> z3=x1.*x2;

>> z4=x1.^2;

>> z5=x2.^2;

>> z=[z1 z2 z3 z4 z5]'
```

```
z =
 Columns 1 through 6

    20    22    25    22    17    24
    25    33    28    25    27    27
   500   726   700   550   459   648
   400   484   625   484   289   576
   625  1089   784   625   729   729

 Columns 7 through 10

    25    29    22    23
    37    38    34    28
   925  1102   748   644
   625   841   484   529
  1369  1444  1156   784

>> zz=z'
zz =

    20    25   500   400   625
    22    33   726   484  1089
    25    28   700   625   784
    22    25   550   484   625
    17    27   459   289   729
    24    27   648   576   729
    25    37   925   625  1369
    29    38  1102   841  1444
    22    34   748   484  1156
    23    28   644   529   784

>> p=zz;
>> a=inv(p'*p)*(p'*y)
a =
   5.3786
  -0.6625
   0.3947
  -0.3728
  -0.1110

>> ym=5.3786*z1-0.6625*z2+0.3947*z3-0.3728*z4-0.1110*z5
```

```
ym =

   69.8645
   81.7047
   72.1810
   69.0415
   66.0578
   71.3127
   90.0910
   91.9550
   82.2886
   75.1094

>> tab=[x1 x2 y ym y-ym]
tab =
   20.0000   25.0000   76.0000   69.8645    6.1355
   22.0000   33.0000   83.0000   81.7047    1.2953
   25.0000   28.0000   74.0000   72.1810    1.8190
   22.0000   25.0000   64.0000   69.0415   -5.0415
   17.0000   27.0000   62.0000   66.0578   -4.0578
   24.0000   27.0000   73.0000   71.3127    1.6873
   25.0000   37.0000   84.0000   90.0910   -6.0910
   29.0000   38.0000   93.0000   91.9550    1.0450
   22.0000   34.0000   89.0000   82.2886    6.7114
   23.0000   28.0000   71.0000   75.1094   -4.1094

>> x=[1:1:10];
>> plot(x,y,'-*',x,F,':p',x,ym,':o'),grid on;
>> legend('Berilgan',' Chiziqli','Chiziqli bolmagan')
>> legend('Location','northwest')
```



Порядок выполнения работы

1. Ознакомиться с теоретической частью по приведенным выше материалам;
2. Освоить алгоритмы для линейный регрессионный анализ в среде Матлаб
3. Построить нелинейной регрессионной модели в среде Матлаб

Контрольные вопросы

1. Что такое регрессионный анализ данных
2. Что такое нелинейный регрессионный анализ данных
3. Алгоритмы для выполнение нелинейный регрессионный анализ в среде Матлаб

Лабораторная работа №5

Тема: Корреляционный анализ статических данных

Цель работы: Изучить основных вычислительных характеристик матриц в Matlab

Теоретические сведения

Данные обычно представляется в виде матрицы определенного размера, т.е. N – строк и M – столбцов, при этом строки обозначают объекты (транзакции), а столбцы атрибуты данных. Например, успеваемость студентов определенный группы можно представить в виде таблицы (5.1.):

Таблица 5.1.

Предмет/ № студента	Математика	Информатика	История Узбекистана	Физика	Физическая культура
1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
2	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
3	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
…	…	…	…	…	…
…	…	…	…	…	…
20	$a_{20,1}$	$a_{20,2}$	$a_{20,3}$	$a_{20,4}$	$a_{20,5}$

Наличие взаимосвязи между двумя случайными величинами X и Y определяется с помощью корреляционного момента.

$\mu_{xy} = M\{(x - M(x))^*(y - M(y))\}$, при этом если $\mu_{xy} \neq 0$ то между X и Y существует корреляционная зависимость, при этом, если $\mu_{xy} > 0 \Rightarrow x\uparrow \Rightarrow y\uparrow$ - прямая корреляционная связь; если $\mu_{xy} < 0 \Rightarrow x\uparrow \Rightarrow y\downarrow$ - обратное корреляционная связь; если $\mu_{xy} = 0$ то между X и Y отсутствует корреляционная связь.

μ_{xy} размерная случайная величина, для расчета силы корреляционной связи μ_{xy} преобразуется в безразмерный вид:

$$ч_{xy} = \frac{\mu_{xy}}{\sigma_x \sigma_y}, \text{ где } \sigma_x = \sqrt{D(x)}, \sigma_y = \sqrt{D(y)}$$

$|ч_{xy}| < 1$, при этом если $ч_{xy} \geq 0.6$ - то корреляционная связь называется примой и существенной. Если $-1 < ч_{xy} \leq 0$ - то корреляционная связь называется обратной и существенной.

Организация процессии корреляционного анализа данных.

Процесс корреляционного анализа по матрице походных данных целесообразно организовать так:

1. все столбцы матрицы проверяется на принадлежность основным законам распределении по критерию x^2.
2. если определенный столбцы матрицы не подчиняется к какому то определенному закону, то этот столбце данных анализируется на наличие случайных ошибок, шумов, фильтрируется и после этого заново приводится по критерию x^2.
3. во возможности для всех столбцов строится их гистограммы с дополнительными числовыми характеристиками $M(a_{ij}), \sigma(a_{ij})$
4. далее, с применением функции Matlab corr(A) вычисляется матрица корнях коэффициентов корреляции.
5. Приводится анализ корреляционной матрицы и определяется задачи регрессии с выделение наиболее существенных факторов на результирующий фактор.

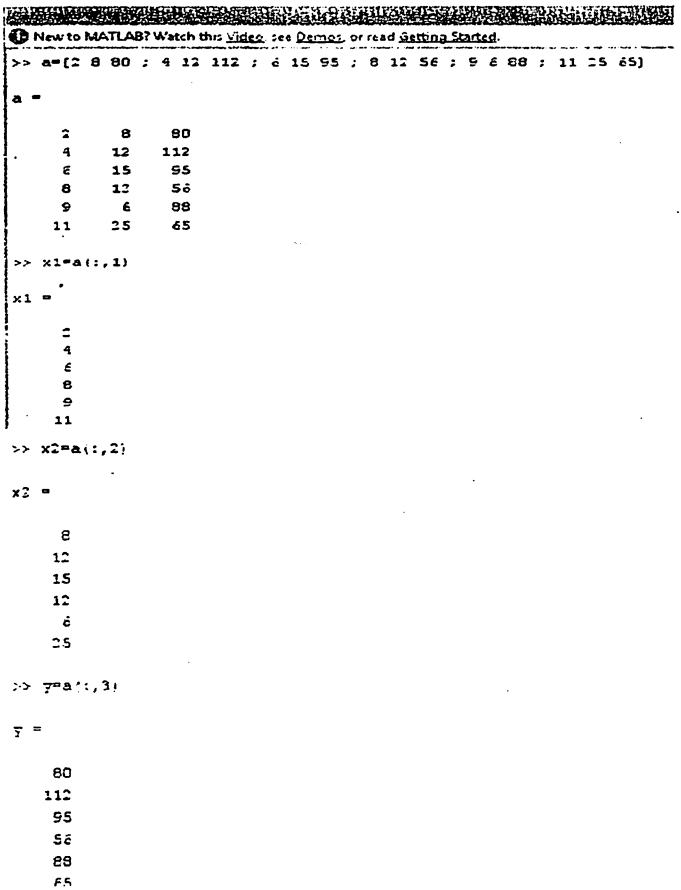
```
a=[ 2 8 80 ; 4 12 112 ; 6 15 95 ; 8 12 56 ; 9 6 88 ; 11 25 65 ]
x1=a(:,1);   x2=a(:,2);   y=a(:,3);
subplot(3,1,1),plot(x1,'*'),grid;
subplot(3,1,2),plot(x2,'*'),grid;
```

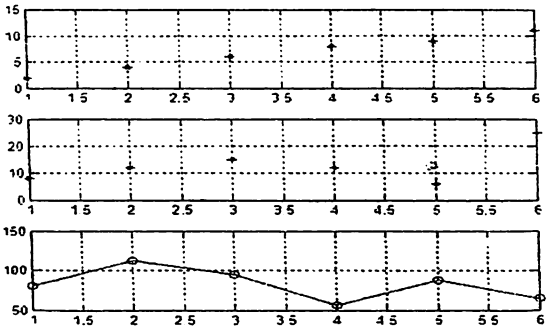
subplot(3,1,3),plot(y,'*'),grid

ar=cov(a) - коверация матрицы

acr=corrcoef(a) - корреляция матрицы

R_{xy}=std(a) - ўртача квадратик четланиш





```
>> ar=cov(a)

ar =

   11.0667   11.8000  -34.7333
   11.8000   44.8000  -42.8000
  -34.7333  -42.8000  414.2667

>> acr=corrcoef(a)

acr =

    1.0000    0.5259   -0.5130
    0.5259    1.0000   -0.3142
   -0.5130   -0.3142    1.0000

>> Pxy=std(a)

Pxy =

    3.3267    6.6933   20.3535
```

Порядок выполнения работы

1. Ознакомиться с теоретической частью по приведенным выше материалам;
2. Освоить операторы для выполнения корреляционный анализ в среде Матлаб
3. Построить корреляционный анализ модели в среде Матлаб

Контрольные вопросы

1. Что такое корреляционный анализ данных
2. Операторы для выполнение корреляционный анализ данных в среде Матлаб
3. Коэффициентные значении корреляционного анализа данных

Лабораторная работа №6

Тема: Кластеризация на основе классификации

Цель работы: Изучить основных вычислительных характеристик матриц в Matlab

Теоретические сведения

Кластерный анализ позволяет рассматривать достаточно большой объем информации и резко сокращать, сжимать большие массивы информации, делать их компактными и наглядными.

Задачи кластеризации состоят в разделении исследуемый множеств объектов на группы «похожих» объектов называемых кластером.

Решением задачи кластеризации является отнесение каждого объектов данных к одному (или нескольким) из заранее определённых классов и построение модели данных определённых разбиение множество объектов данных на классы.

В задачи кластеризации отнесение каждого из объектов данных осуществляет к одному (или нескольким) из заранее неопределённых классов. Определение кластеров и разбиение по ним объектов данных выражается в итоговой модели данных, которая является решением задачи кластеризации.

Постановка задачи кластеризации.

Дано: Набор данных следующими свойствами:

- Каждый экземпляр данных выражается чётким числовым значением;
- Класс для каждого конкретного экземпляра данных неизвестен.

Найти:

- Способ сравнение данных между собой (меру сходство);
- Способ кластеризации;
- Разбиение данных по кластером.

Дано множество объектов данных I, каждый из которых представлен набором атрибутов. Требуется настроить множество кластеров C и отображение F множества I на множество C т.е. $F: I \to C$

Отображение F задаёт модель данных, являющем решением задачи. Множество I определим следующим образом:

$I = \{i_1, i_2, \ldots\ldots, i_j, \ldots\ldots, i_n\}$ где i_j - исследуемый объект.

Задача кластеризации состоит в построение множества:

$C = \{c_1, c_2, \ldots\ldots, c_k, \ldots\ldots, c_g\}$

Здесь c_k - кластер, содержащий похожие друг на друге объекты из множества I:

$c_k = \{i_j, i_p \mid i_j, i_p \in I \cup d(i_j, i_p) < \sigma\}$

где σ - величина определяющие меру близости для включение объектов в один кластер;

$d(i_j, i_p)$ - мера близости между объектами, называемая расстояние.

При этом:

$1^0. d(i_j, i_p) \geq 0$ для $\forall i_j, i_p$

$2^0. d(i_j, i_p) = 0 : i_j = i_p$

$3^0. d(i_j, i_p) = d(i_p, i_j)$

$4^0. d(i_j, i_p) \leq d(i_j, i_r) + d(i_r, i_p)$

Если $d(i_j, i_r) < \sigma \Rightarrow i_j, i_p$ близки и помещается в один кластер.

Алгоритм кластеризации.

Большинство алгоритмов кластеризации используют в качестве входящих данных матрицу D:

$$D = \begin{bmatrix} 0 & d(e_1, e_2) & \ldots\ldots & d(e_1, e_n) \\ d(e_2, e_1) & 0 & \ldots\ldots & d(e_2, e_n) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ d(e_n, e_1) & d(e_n, e_2) & \ldots\ldots & 0 \end{bmatrix}$$

Меры близости кластеров.

Приведем наиболее известные меры близости:

1) Евклидово расстояние:

$$d_2(x_i, x_j) = \sqrt{\sum_{it=1}^{m} (x_{it} - x_{jt})^2}$$

2) Хемминга расстояние:

$$d_H(x_i, x_j) = \sum_{it=1}^{m} |x_{it} - x_{jt}|$$

3) Расстояние Чебишов:

$$d_\infty(x_i, x_j) = \max_{1 \le t \le m} |x_{it} - x_{jt}|$$

Синтаксис

```
Y = pdist(X)
Y = pdist(X,'metric')
Y = pdist(X,distfun,p1,p2,...)
Y = pdist(X,'minkowski',p)
```

МИСОЛ 1.

```
>> X=normrnd(0,1,7,3)
X =
   0.6353   0.4620  -0.1132
  -0.6014  -0.3210   0.3792
   0.5512   1.2366   0.9442
  -1.0998  -0.6313  -2.1204
   0.0860  -2.3252  -0.6447
  -2.0046  -1.2316  -0.7043
  -0.4931   1.0556  -1.0181
>> Y = pdist(X)
Y =
  Columns 1 through 11
    1.5444   1.3134   2.8696   2.8902   3.1917   1.5635   2.0183   2.5677   2.3532
1.9930   1.9646
  Columns 12 through 21
    3.9505   3.9277   3.9168   2.2302   2.5404   1.7845   2.1045   2.3601   3.4504
2.7595
>> S = squareform(Y)
S =
         0   1.5444   1.3134   2.8696   2.8902   3.1917   1.5635
    1.5444        0   2.0183   2.5677   2.3532   1.9930   1.9646
    1.3134   2.0183        0   3.9505   3.9277   3.9168   2.2302
```

2.8696	2.5677	3.9505	0	2.5404	1.7845	2.1045
2.8902	2.3532	3.9277	2.5404	0	2.3601	3.4504
3.1917	1.9930	3.9168	1.7845	2.3601	0	2.7595
1.5635	1.9646	2.2302	2.1045	3.4504	2.7595	0

При помощи функции squareform вектор Y можно конвертировать в квадратную матрицу. Элемент (*i, j)*полученной матрицы, при *i<j*, будет соответствовать расстоянию между *i-м и j-м* объектами исходного множества данных:

Y = pdist(X,'*metric*') входной параметр '*metric*' определяет вид расстояния. Предусмотрены следующие виды расстояний между объектами:

Виды парных расстояний между объектами, используемые в pdist:

1. Евклидово расстояние:

$$d_{rs}^2 = (x_r - x_s)(x_r - x_s)'$$

2. Стандартизованное Евклидово расстояние:

$$d_{rs}^2 = (x_r - x_s)D^{-1}(x_r - x_s)'$$

где D - диагональная матрица. Диагональными элементами D являются выборочные дисперсии признаков многомерной случайной величины X_j ,j=1..m.

3. Расстояние Махаланобиса:

$$d_{rs}^2 = (x_r - x_s)V^{-1}(x_r - x_s)'$$

где V - ковариационная матрица, рассчитанная по выборке X.

4. Расстояние по Манхеттену:

$$d_{rs} = \sum_{j=1}^{n} |x_{rj} - x_{sj}|$$

5. Метрика Минковского:

$$d_{rs} = \left[\sum_{j=1}^{n} |x_{rj} - x_{sj}|^p\right]^{\frac{1}{p}}$$

6. Расстояние Хэмминга:

$$d_{rs} = \left(\#(x_{rj} \neq x_{sj})/n\right)$$

Пример 1

1. Расчет парных расстояний между объектами исходного множества данных. В качестве исходного множества данных используется двумерная случайная величина. Количество объектов в множестве исходных данных равно 7.

```
>> X = [3 1.7; 1 1; 2 3; 2 2.5; 1.2 1; 1.1 1.5; 3 1]
X =
   3.0000   1.7000
   1.0000   1.0000
   2.0000   3.0000
   2.0000   2.5000
   1.2000   1.0000
   1.1000   1.5000
   3.0000   1.0000
>> Y = pdist(X)'
Y =
   2.1190
   1.6401
   1.2806
   1.9313
   1.9105
   0.7000
   2.2361
   1.8028
   0.2000
   0.5099
   2.0000
   0.5000
   2.1541
   1.7493
   2.2361
   1.7000
   1.3454
   1.8028
```

```
  0.5099
  1.8000
  1.9647
```

Пример 2

```
>> X = [3 1.7; 1 1; 2 3; 2 2.5; 1.2 1; 1.1 1.5; 3 1]
X =
   3.0000   1.7000
   1.0000   1.0000
   2.0000   3.0000
   2.0000   2.5000
   1.2000   1.0000
   1.1000   1.5000
   3.0000   1.0000
>> Y1 = pdist(X, 'euclidean')';
>> Y2 = pdist(X, 'seuclidean')';
>> Y3 = pdist(X, 'mahalanobis')';
>> Y4 = pdist(X, 'cityblock')';
>> Y5 = pdist(X, 'minkowski')';
>> Y6 = pdist(X, 'cosine')';
>> Y7 = pdist(X, 'hamming')';
>> Y8 = pdist(X, 'jaccard')';
>> Y = [Y1'; Y2'; Y3'; Y4'; Y5'; Y6'; Y7'; Y8']'
 Y =
   2.1190   2.4992   2.3879   2.7000   2.1190   0.0362   1.0000   1.0000
   1.6401   2.0036   2.1983   2.3000   1.6401   0.1072   1.0000   1.0000
   1.2806   1.5399   1.6946   1.8000   1.2806   0.0715   1.0000   1.0000
   1.9313   2.2815   2.1684   2.5000   1.9313   0.0160   1.0000   1.0000
   1.9105   2.2378   2.2284   2.1000   1.9105   0.0879   1.0000   1.0000
   0.7000   0.8757   0.8895   0.7000   0.7000   0.0187   0.5000   0.5000
   2.2361   2.7621   2.6097   3.0000   2.2361   0.0194   1.0000   1.0000
   1.8028   2.2115   2.0616   2.5000   1.8028   0.0061   1.0000   1.0000
   0.2000   0.2341   0.2378   0.2000   0.2000   0.0041   0.5000   0.5000
   0.5099   0.6363   0.6255   0.6000   0.5099   0.0116   1.0000   1.0000
   2.0000   2.3408   2.3778   2.0000   2.0000   0.1056   0.5000   0.5000
   0.5000   0.6255   0.6353   0.5000   0.5000   0.0038   0.5000   0.5000
   2.1541   2.6713   2.5522   2.8000   2.1541   0.0412   1.0000   1.0000
   1.7493   2.1518   2.0153   2.4000   1.7493   0.0010   1.0000   1.0000
   2.2361   2.7621   2.9890   3.0000   2.2361   0.2106   1.0000   1.0000
   1.7000   2.0970   1.9750   2.3000   1.7000   0.0202   1.0000   1.0000
   1.3454   1.6354   1.5106   1.9000   1.3454   0.0009   1.0000   1.0000
   1.8028   2.2115   2.4172   2.5000   1.8028   0.1604   1.0000   1.0000
   0.5099   0.6363   0.6666   0.6000   0.5099   0.0295   1.0000   1.0000
   1.8000   2.1067   2.1400   1.8000   1.8000   0.0688   0.5000   0.5000
```

1.9647 2.3101 2.4517 2.4000 1.9647 0.1840 1.0000 1.0000

Пример 3

```
>> X = [3 1.7; 1 1; 2 3; 2 2.5; 1.2 1; 1.1 1.5; 3 1]
X =
   3.0000   1.7000
   1.0000   1.0000
   2.0000   3.0000
   2.0000   2.5000
   1.2000   1.0000
   1.1000   1.5000
   3.0000   1.0000
>> Y1 = pdist(X,'minkowski',1)';
>> Y2 = pdist(X,'minkowski',2)';
>> Y3 = pdist(X,'minkowski',3)';
>> Y = [Y1'; Y2'; Y3']'
Y =
   2.7000   2.1190   2.0282
   2.3000   1.6401   1.4732
   1.8000   1.2806   1.1478
   2.5000   1.9313   1.8346
   2.1000   1.9105   1.9007
   0.7000   0.7000   0.7000
   3.0000   2.2361   2.0801
   2.5000   1.8028   1.6355
   0.2000   0.2000   0.2000
   0.6000   0.5099   0.5013
   2.0000   2.0000   2.0000
   0.5000   0.5000   0.5000
   2.8000   2.1541   2.0418
   2.4000   1.7493   1.6010
   3.0000   2.2361   2.0801
   2.3000   1.7000   1.5723
   1.9000   1.3454   1.2002
   2.5000   1.8028   1.6355
   0.6000   0.5099   0.5013
   1.8000   1.8000   1.8000
   2.4000   1.9647   1.9115
```

LINKAGE DENDROGRAM

```
Z = linkage(Y)
Z = linkage(Y,'method')
```

Z = linkage(Y) функция позволяет сформировать иерархическое дерево бинарных кластеров с использованием алгоритма <ближайшего соседа>.

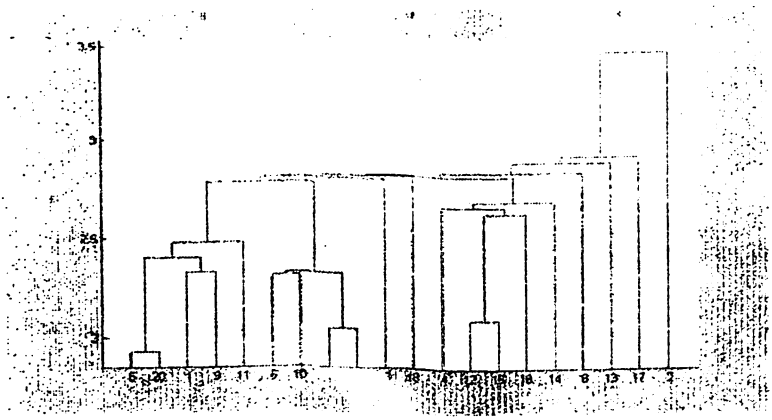
Z = linkage(Y,'*method*') входной аргумент '*method*' позволяет задать алгоритм кластеризации.

Алгоритмы кластеризации

Примеры использования функции формирования иерархического дерева бинарных кластеров

1. Количество объектов в множестве исходных данных равно 20. Графическое представление дерева бинарных кластеров выполняется с помощью функции dendrogram.

```
>> X=normrnd(0,1,20,10);
>> Y = pdist(X);
>> Z = linkage(Y)
Z =
   5.0000   20.0000    1.9330
   7.0000   19.0000    2.0429
  12.0000   15.0000    2.0979
   6.0000   10.0000    2.3100
  24.0000   22.0000    2.3239
   1.0000    9.0000    2.3283
  26.0000   21.0000    2.4018
  27.0000   11.0000    2.4800
  23.0000   16.0000    2.6409
   4.0000   29.0000    2.6727
  30.0000   14.0000    2.7020
  28.0000   25.0000    2.7839
  32.0000    3.0000    2.8100
  33.0000   18.0000    2.8246
  34.0000   31.0000    2.8279
  35.0000    8.0000    2.8495
  36.0000   13.0000    2.9005
  37.0000   17.0000    2.9329
  38.0000    2.0000    3.4541
>> H = dendrogram(Z);
```



2. Формирование иерархического дерева бинарных кластеров для 2-х мерной случайной величины. Количество объектов в множестве исходных данных равно 7. Сравниваются различные алгоритмы кластеризации. Графическое представление результатов кластеризации выполняется с помощью функции dendrogram.

2.1. Исходные данные:

```
>> X = [3 1.7; 1 1; 2 3; 2 2.5; 1.2 1; 1.1 1.5; 3 1]
X =
   3.0000   1.7000
   1.0000   1.0000
   2.0000   3.0000
   2.0000   2.5000
   1.2000   1.0000
   1.1000   1.5000
   3.0000   1.0000
>> Y = pdist(X);
Y =
  Columns 1 through 11
   2.1190   1.6401   1.2806   1.9313   1.9105   0.7000   2.2361   1.8028   0.2000
0.5099   2.0000
  Columns 12 through 21
   0.5000   2.1541   1.7493   2.2361   1.7000   1.3454   1.8028   0.5099   1.8000
1.9647
```

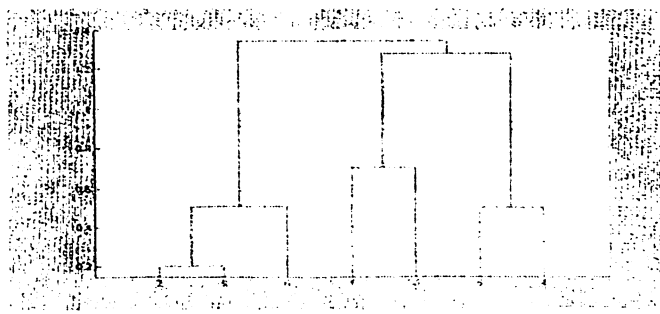
2.2. Кластеризация при помощи алгоритма <ближайшего соседа>

```
>> Z = linkage(Y, 'single')
```

Z =

2.0000	5.0000	0.2000
3.0000	4.0000	0.5000
8.0000	6.0000	0.5099
1.0000	7.0000	0.7000
11.0000	9.0000	1.2806
12.0000	10.0000	1.3454

>> dendrogram(Z);



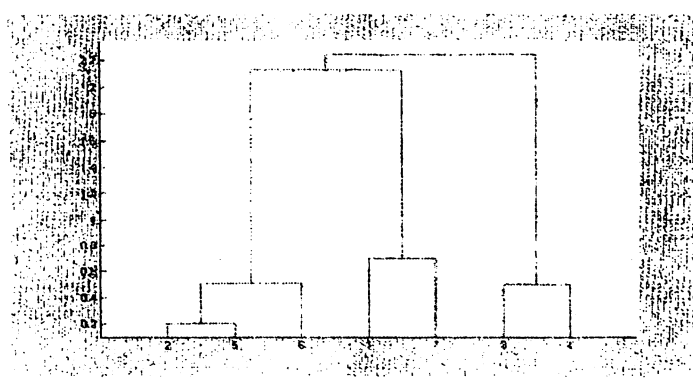
2.3. Кластеризация при помощи алгоритма <дальнего соседа>

>> Z = linkage(Y, 'complete')

Z =

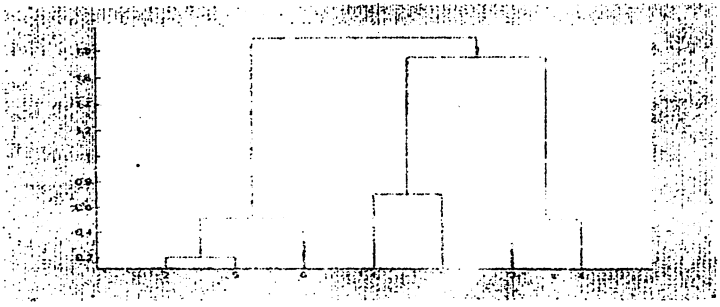
2.0000	5.0000	0.2000
3.0000	4.0000	0.5000
8.0000	6.0000	0.5099
1.0000	7.0000	0.7000
11.0000	10.0000	2.1190
12.0000	9.0000	2.2361

>> dendrogram(Z);



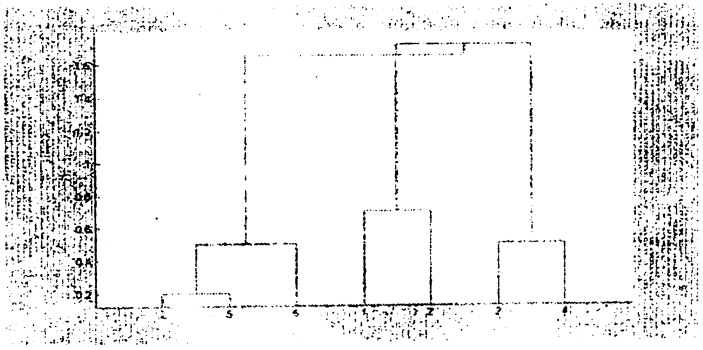
2.4. Кластеризация при помощи алгоритма <средней связи>

```
>> Z = linkage(Y, 'average')
Z =
   2.0000    5.0000    0.2000
   3.0000    4.0000    0.5000
   8.0000    6.0000    0.5099
   1.0000    7.0000    0.7000
  11.0000    9.0000    1.7399
  12.0000   10.0000    1.8928
>> dendrogram(Z);
```



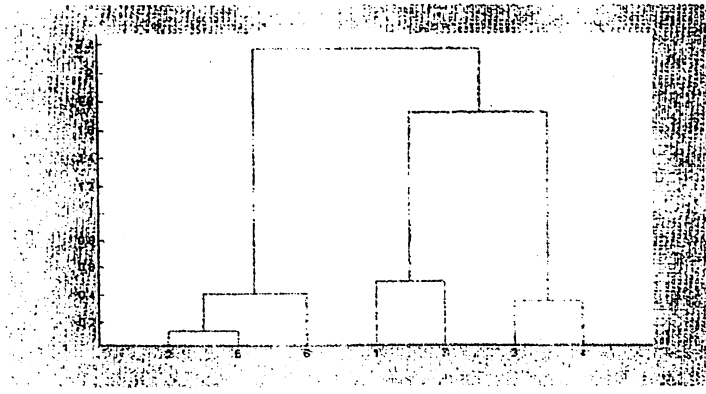
2.5. Кластеризация при помощи алгоритма центроидного алгоритма

```
>> Z = linkage(Y, 'centroid')
Z =
   2.0000    5.0000    0.2000
   8.0000    6.0000    0.5000
   3.0000    4.0000    0.5000
   1.0000    7.0000    0.7000
  11.0000   10.0000    1.7205
  12.0000    9.0000    1.6554
>> dendrogram(Z);
```



2.6. Кластеризация при помощи алгоритма пошагового алгоритма

```
>> Z = linkage(Y, 'ward')
Z =
   2.0000    5.0000    0.1414
   3.0000    4.0000    0.3536
   8.0000    6.0000    0.4082
   1.0000    7.0000    0.4950
  11.0000    9.0000    1.7205
  12.0000   10.0000    2.1674
>> dendrogram(Z);
```



Порядок выполнения работы

1. Ознакомиться с теоретической частью по приведенным выше материалам;
2. Ознакомиться с операторами кластеризации данных Матлаб
3. Освоить алгоритмы кастеризации
4. Построить кластеризация данных модели в среде Матлаб

Контрольные вопросы

1. Что такое кластеризация
2. Что такое классификация
3. Алгоритмы кастеризации
4. Операторы для выполнение кластеризации данных в среде Матлаб

Лабораторная работа №7

Тема: Изучение нечётких множеств на основе пакета FLT в среде Matlab

Цель работы: Изучить основных вычислительных характеристик матриц в Matlab

Теоретические сведения

Понятие нечеткого множества - эта попытка математической формализации нечеткой информации для построения математических моделей. В основе этого понятия лежит представление о том, что составляющие данное множество элементы, обладающие общим свойством, могут обладать этим свойством в различной степени и, следовательно принадлежать к данному множеству с различной степенью. При таком подходе высказывания типа "такой-то элемент принадлежит данному множеству" теряют смысл, поскольку необходимо указать "насколько сильно" или с какой степенью конкретный элемент удовлетворяет свойствам данного множества.

Определение 1. *Нечетким множеством (fuzzy set)* $\tilde{A}$ на универсальном множестве U называется совокупность пар $(\mu_A(u), u)$, где $\mu_A(u)$ - степень принадлежности элемента $u \in U$ к нечеткому множеству $\tilde{A}$. Степень принадлежности - это число из диапазона [0, 1]. Чем выше степень принадлежности, тем в большей мерой элемент универсального множества соответствует свойствам нечеткого множества.

Определение 2. *Функцией принадлежности (membership function)* называется функция, которая позволяет вычислить степень принадлежности произвольного элемента универсального множества к нечеткому множеству.

Если универсальное множество состоит из конечного количества элементов $U = \{u_1, u_2, \ldots u_k\}$, тогда нечеткое множество $\tilde{A}$ записывается в виде $\tilde{A} = \sum_{i=1}^{k} \mu_A(u_i)/u_i$. В случае непрерывного множества U используют такое обозначение $\tilde{A} = \int_U \mu_A(u)/u$.

Примечание: знаки Σ и $\int$ в этих формулах означают совокупность пар $\mu_A(u)$ и u.

Определение 3. *Лингвистической переменной* (linguistic variable) называется переменная, значениями которой могут быть слова или словосочетания некоторого естественного или искусственного языка.

Определение 4. *Терм–множеством (term set)* называется множество всех возможных значений лингвистической переменной.

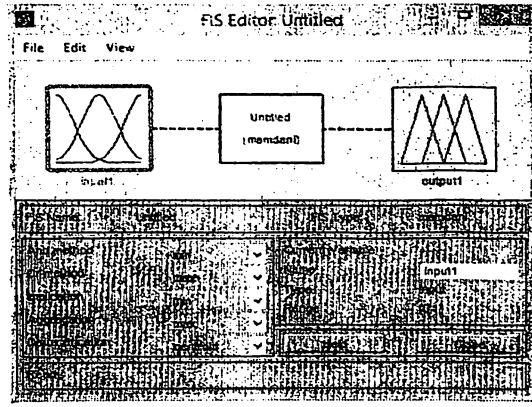
Определение 5. *Термом (term)* называется любой элемент терм–множества. В теории нечетких множеств терм формализуется нечетким множеством с помощью функции принадлежности.

Пример 2. Рассмотрим переменную "*скорость автомобиля*", которая оценивается по шкале "*низкая*", "*средняя*", "*высокая*" и "*очень высокая*".

В этом примере лингвистической переменной является "*скорость автомобиля*", термами - лингвистические оценки "*низкая*", "*средняя*", "*высокая*" и "*очень высокая*", которые и составляют терм–множество.

Вводим команда *fuzzy*

```
>> fuzzy
```



Пример: Урожай хлопко-производство. Для достижения эффективного урожая приводиться несколько взаимосвязанных фоакторов. По естественный эффективности таблица 7.1.

таблица 7.1

Температура / Влажность	холода	Сред.холод	нормал	жара	знойный
низкий	низкий	низкий	удовлетво	удовлетво	низкий
Низко влажно	низкий	низкий	средний	средний	средний
средний	низкий	удовлетво	средний	хороший	хороший
высокый	низкий	удовлетво	средний	хороший	хороший
мокрый	низкий	удовлетво	хороший	хороший	отлично

По искуственного эффективности таблица 7.2

минерализация / Ishlan.lik	плохо	удовлетво	средний	хороший	высокый
неудовлетво	низкий	низкий	удовлетво	удовлетво	низкий
удовлетво	низкий	удовлетво	средний	средний	средний
средний	низкий	удовлетво	средний	хороший	хороший
хороший	низкий	удовлетво	средний	хороший	хороший
отлично	удовлетво	средний	хороший	хороший	отлично

Температура:X(1)={ холода, сред.холод, normal, жара, знойный }

Влажность:X(2)={ низкий, низко влажно, средний, высокый, мокрый }

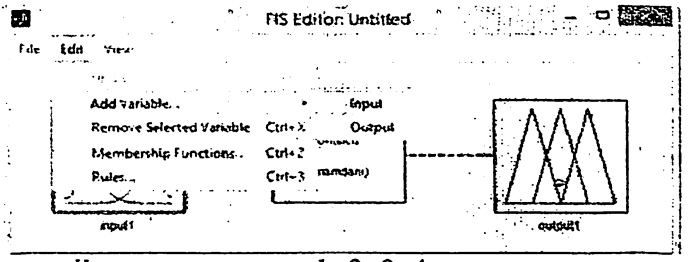
Минерализация:X(3)={ плохо, удовлетво, средний, хороший, высокый }

Ishlanganlik:X(4)={ неудовлетво, удовлетво, средний, хороший, отлично }

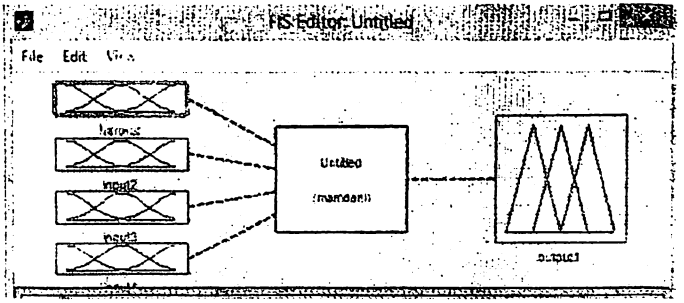
Естественная эффективность:y(1)={ низкий, удовлетво,средний, хороший, отлично }

Искуственная эффективность:y(2)={ низкий, удовлетво, средний, хороший, отлично }

1. Добавления входные переменные

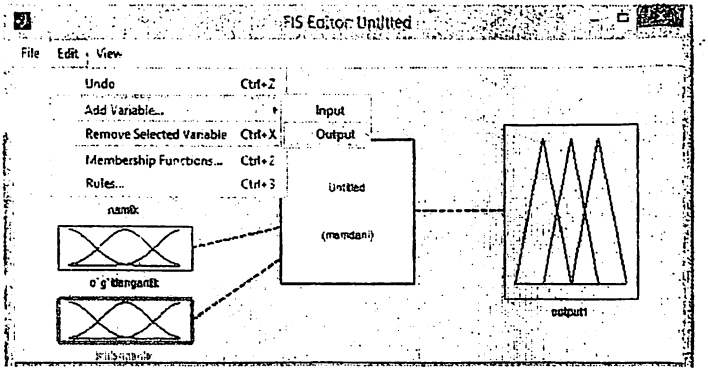


Четыре переменные x1,x2,x3,x4 и наименвоание

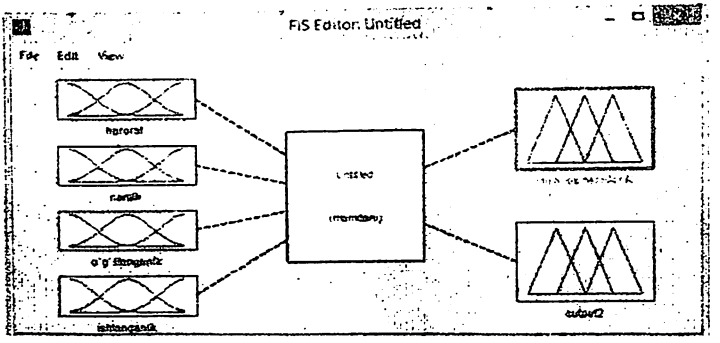


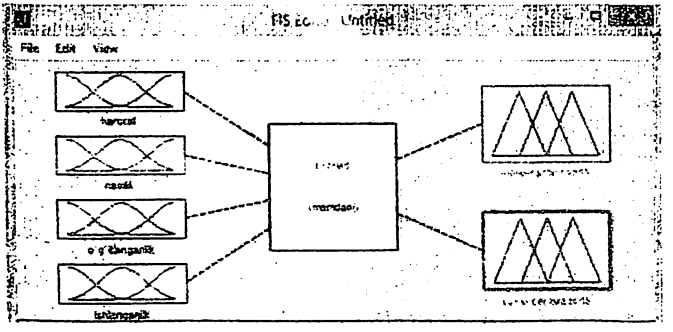
Вводится две выходные параметры

2. Добавится выходные переменные

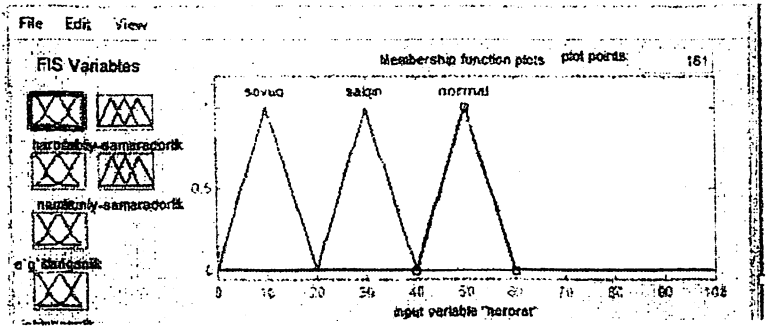


наименование



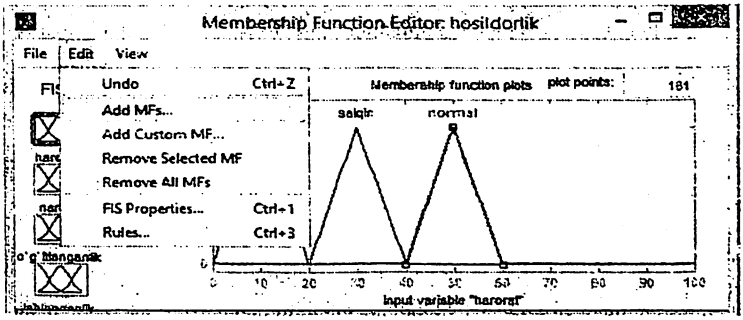


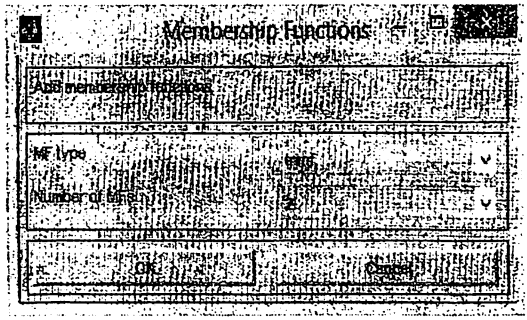
3. В следующем этапе вводиться x1,x2,x3,x4 принимающие параметры

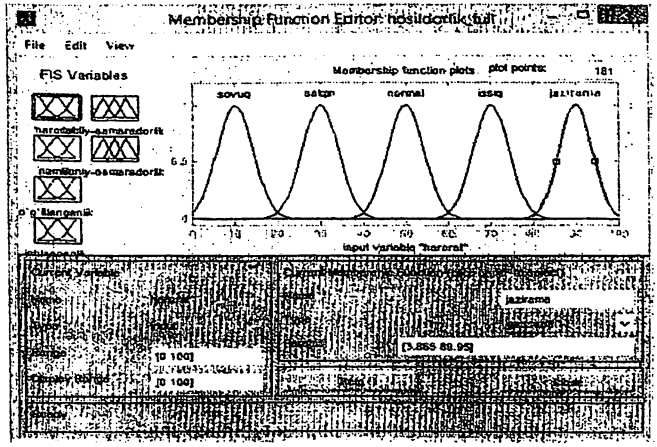


Задается интервал (0 - 100), (gaussmf).

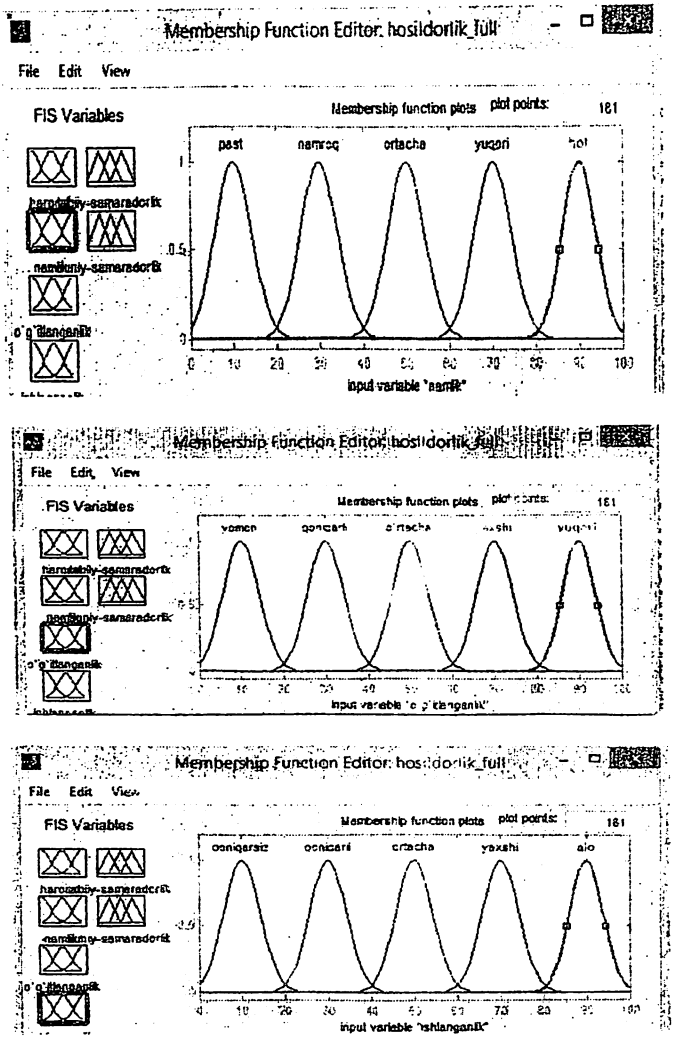
Добавления нового параметра



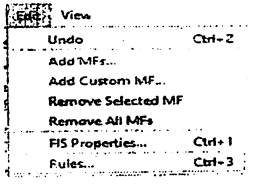




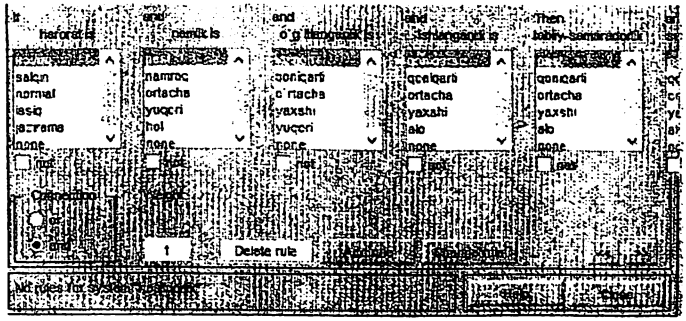
После добавления все параметров

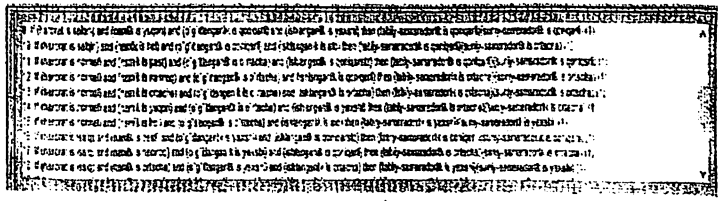


4. После добавления все параметров, создается база знаний

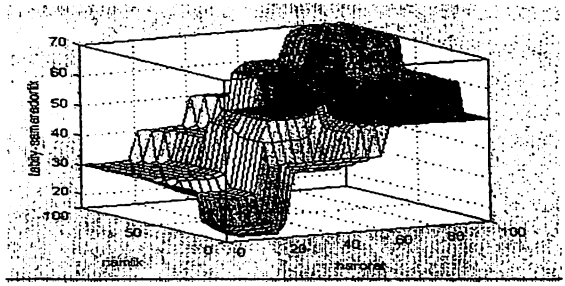


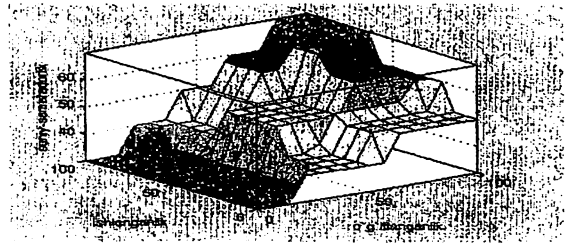
С помощью Add rule создается 25 правил.

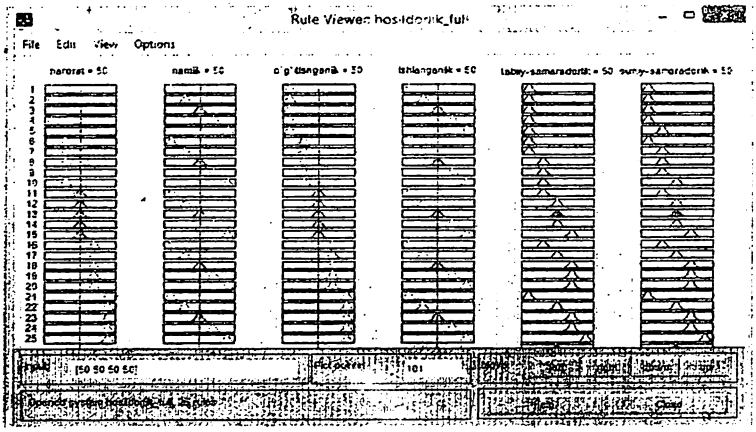




Последовательность визуализация база знаний View->Surface.







Порядок выполнения работы

1. Ознакомиться с теоретической частью по приведенным выше материалам;
2. Ознакомиться с возможностью Fuzzy Logic Toolbox
3. Построить нечёткая модель данных с помощью Fuzzy Logic Toolbox в среде Матлаб

Контрольные вопросы

1. Что такое нечёткая множества
2. Что такое функция принадлежности
3. Что такое *term*
4. Возможности Fuzzy Logic Toolbox

Список использованной литературы:

1. Ўзбекистон Республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. Ўзбекистон республикаси Президентининг ПФ-4947-сон фармони. Тошкент, 2017 йил 7-феврал.
2. Мирзиёев Ш.М. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Тошкент. «Ўзбекистон», НМИУ, 2017. – 48 б.
3. R.N. Usmanov, A.N. Mirzayev, K.K. Seitnazarov. Sonli usullar va dasturlash. – T.: «Fan va texnologiya», 2016, 192 bet.
4. А.А. Барсегян, М.С.Куприянов, В.В. Степаненко, И.И. Холод. Методы и модели анализа данных: OLAP и Data Mining. Москва, 2013
5. Ю.Д. Макленнен, Ч. Танг, Б. Криват. Data Mining — интеллектуальный анализ данных. 2013
6. Программирование, численные методы СП.б. БХВ-Питербург 2005.
7. MATLAB асослари. Тўлан Дадажонов, Муҳсин Муҳитдинов, “Фан” нашриёти – 2008, 640 бет.
8. Лазарев Юрий Федорович, “Начала программирования в среде MatLAB”: Учебное пособие. - К.: НТУУ "КПИ", 2003. - 424 с.

Интернет источники и портал Зиёнет

1. Matlab.exponenta.ru
2. Math.com
3. gov.uz

ЗАКЛЮЧЕНИЕ

В методическом указании рассмотрены базовые знания по анализам данны на основе математического моделирования в среде Matlab. Приводятся основные характеристики для вычисления и анализа матричных данных. Это позволяет рассматривать функции аппроксимации и интерполяции данных, линейный и нелинейный регрессионный анализ данных, корреляционный анализ. Приводятся понятия кластеризации данных на основе классификации, теория нечетких множеств на основе FLT в среде Matlab.

СОДЕРЖАНИЕ

Формат 60x84 1/16. Печ.лист 3,5.
Заказ № 215. Тираж 30.
Отпечатано в «Редакционно издательском»
отделе при ТУИТ.
Ташкент ул. Амир Темур, 108.

Методические указания по выполнению лабораторных работ по курсу «Интеллектуальный анализ данных» для бакалавров обучающих по направлению 5330500 – «Компьютерный инжиниринг»

Рассмотрены на заседании кафедры «Компьютерные системы»
от «15» 05 2018
Протокол № 22

Рассмотрены на заседании факультета «Компьютерный инжиниринг»
от «23» 05 2018
Протокол № 34

Рассмотрены и рекомендованы к изданию на заседании научно-методического Совета ТУИТ
от «23» 05 2018
Протокол № 10 (111)

Составители:
д.т.н., профессор, «Компьютерные системы» ТУИТ Усманов Р.Н.,
ассистент кафедры «Компьютерные системы» ТУИТ Кутлымуратов А.Ж.,
старший преподаватель кафедры «Компьютерные системы» ТУИТ Хабирова Д.Н.

Рецензенты:
Хушвактов С.Х.,
Назирова Э.Ш.

Ответственный редактор: д.т.н. Джуманов Ж.Х.

Корректор: Шукуров К.Э.