Презентация по лабораторной работе №5

Дисциплина “Научное программирование”

Живцова А.А.

11 октября 2024

Кафедра теории вероятностей и кибербезопасности, Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

Информация

Докладчик

- Живцова Анна Александровна
- студент кафедры теории вероятностей и кибербезопасности
- Российский университет дружбы народов имени Патриса Лумумбы
- zhivtsova_aa@pfur.ru
- https://github.com/AnnaZhiv

Вводная часть

Актуальность

Построение полиномиальной регрессии и преобразование плоских изображений – распространенные задачи в науной деятельности. Octave предоставляет удобный инструментарий для быстрой и автоматизированной реализации этих задач и графического представления результатов.

Объект и предмет исследования

- Метод полиномиальной подгонки
- Метод наименьших кадратов
- Плоские фигуры
- Матричные преобразования для операций
  - вращения
  - отражения
  - сжатия

Цели

- Изучить и реализовать в Octave метод построения полиномиальной регрессии
- Изучить и реализовать в Octave методы преобразования изображений

Задачи

- Изучить и реализовать метод построения полиномиальной регрессии второго порядка
- Реализовать построение полиномиальной регрессии второго порядка с помощью встроенной функции Octave
- Изобразить результат регрессии
- Построиь изобрабражение замкнутой линии
- Изучить и реализовать с помощью матричных преобразований операции
  - вращения
  - отражения
  - сжатия
- Изобразить результаты применения данных операций

Материалы и методы

- Язык научного программирования Octave
- Среда программирования GNUoctave

Выполнение работы

Подгонка полиномиальной кривой

Сначала самостоятельно найдем коэффициенты подгоночной параболы, далее сверим их с теми, что дает встроенная функция `= polyfit`.

```
>> D = [ 1 1; 2 2; 3 5; 4 4; 5 2; 6 -3];
>> xdata = D(:,1);
>> ydata = D(:,2);
>> A = ones(6,3);
>> A(:,1) = xdata .^2;
>> A(:,2) = xdata;
>> A
A =

    1    1    1
    4    2    1
    9    3    1
   16    4    1
   25    5    1
   36    6    1

>>  B = A' * A;
>>  B (:,4) = A' * ydata;
>> B_res = rref(B)
B_res =

   1.0000         0         0   -0.8929
        0    1.0000         0    5.6500
        0         0    1.0000   -4.4000

>> P = polyfit (xdata, ydata, 2)
P =
```

Реализуем два вращения на углы 90 и 225 градусов. Отразим изображение относительно прямой $y = x$. Увеличим изображение в два раза.

Повороты

```
theta1 = 90*pi/180;
R1 = [cos(theta1) -sin(theta1); sin(theta1) cos(theta1)];
RD1 = R1*D;
theta2 = 225*pi/180;
R2 = [cos(theta2) -sin(theta2); sin(theta2) cos(theta2)];
 RD2 = R2*D;
plot (x,y, 'bo-', RD1(1,:), RD1(2,:) , 'ro-' , RD2(1,:), RD2(2,:), 'go-')
axis ([-4 4 -4 4] , 'equal' );
grid on ;
 legend ('original' , 'rotated 90 deg' , 'rotated 225 deg' );
```

Отражение

```
R = [0 1; 1 0];
 RD = R*D;
plot (x,y,'o-',RD(1,:),RD(2,:),'o-');
 axis([-1 4 -1 4], 'equal');
axis([-1 5 -1 5], 'equal');
grid on ;
legend ( 'original' , 'reflected' )
```

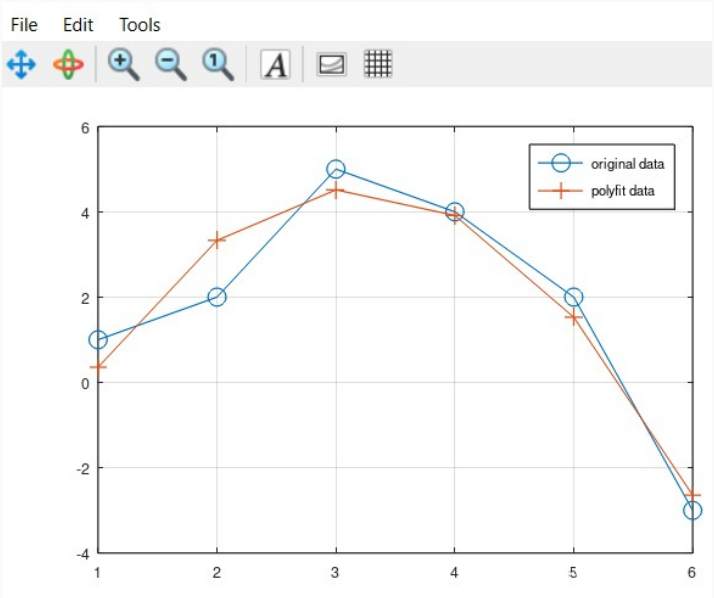
Сжатие

```
> T = [2 0; 0 2];
> TD = T*D;
> x1 = TD(1,:); y1 = TD(2,:);
> plot (x, y, 'o-', x1, y1,'o-')
>  axis ([-1 7 -1 7], 'equal');
> grid on;
>  legend ('original', 'expanded')
```

Рис. 2: Реализация матричных преобразований

Результаты

Полиномиальная регрессия



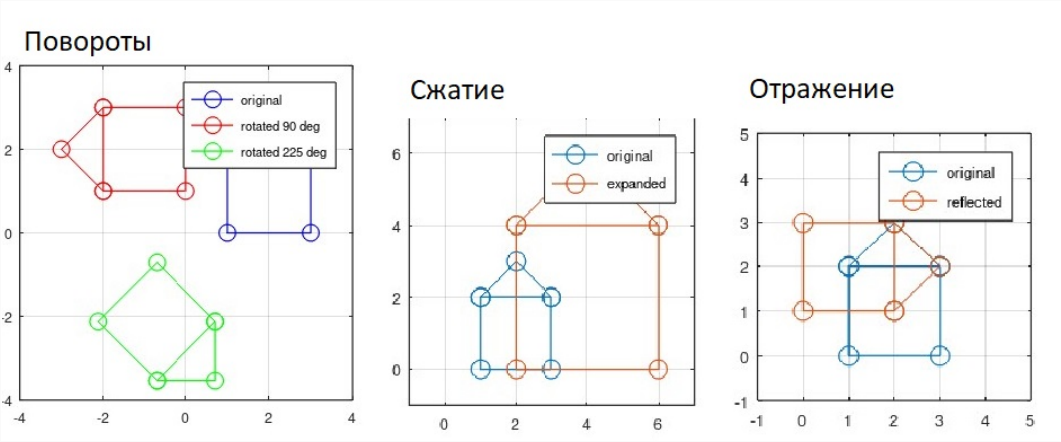


Рис. 4: Результат применения матричных преобразований

Выводы

Выводы

В данной работе я познакомилась с методом построения полиномиальной регресси. Изучила и реализовала метод построения полиномиальной регрессии второго порядка с помощью наименьших квадратов. Сравнила результаты с результатами встроенной функции Octave. Изобразила результат регрессии.

Также я изучила и реализовала в Octave матричные операции для преобразования плоской фигуры. Конкретно, я выполнила операци вращения, отражения и сжатия, а также изобразила результаты применения данных операций.