MATH 151A PROJECT 3

Shiqi Liang, Yuchen Yao

February 2021

1 Introduction: Cubic Spline Interpolation and Rent In Westwood

Cubic spline interpolation is a powerful tool for interpolating polynomial and is often much smoother than other polynomial like Lagrange and Newton polynomial, and also has smaller errors as well. This powerful tool could be used to interpolate data that could be of many practical use.

To apply this, we would apply cubic spline interpolation in projecting rent in Westwood. We know that rent is partially related to the area of the room/apartment, so it would be interesting to see such a relationship. This is important because it would tell us exactly how accurate cubic spline interpolation on this scenario and how rent depends on area.

2 Rent Data

We will look at cubic spline's performance on interpolating rent in westwood. All the data is taken from online housing information website Zillow. For privacy issues, we will omit the address of those actual data. Visit https://www.zillow.com/westwood-los-angeles-ca/rentals/ for more information. For this project, we will explore interpolating rent prices with square feet of the room/apartment for rent.

For the nodes used for interpolation, we have

sqft	rent per month
530	1899
650	2145
750	2582
1000	2700
1200	2575
1360	2900
1425	3600

To test the results of our cubic spline interpolation, we also have data to be tested:

sqft	rent per month
620	2145
710	2930
800	1950
1050	2495
1300	2600
1409	3995

Note that for every interval, we have a node ready for testing.

3 Cubic Spline Method

Our method of cubic spline interpolation is adopted form the textbook and from the Cubic Spline page on Science Direct. The cubic spline method is based on the following polynomial:

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

and we assume that for each interval we have a different set of coefficients.

We start by finding the appropriate a_i,b_i,c_i and d_i for each interval. Since we know

$$S_j(x_j) = a_j = f(x_j)$$

by the conditions of cubic spline, we have that

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3$$

and we define

$$h_j = x_{j+1} - x_j$$

By solving the system of equations and conditions, we can find that

$$a_i = f_i$$

$$b_i = \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}$$

By plugging this system of equations into real nodes points, we could solve for c_i

4 Coding in Matlab

The exact matlab code would be provided at the end of the report. Here, we would include the pseudo code that guide us through writing this program. This program is adopted from the textbook and Professor Hundley from Whitman College:

```
input x
input f

calculate each interval as hi

calculate ai for each interval

set l0=1
mu0=0
z0=0

for each interval i
    set li=2(x_i+1 - x_i-1) - h_i-1mui-1
    mui=hi/li
    zi=(ai-h_(i-1)z_(i-1)/li
set ln=1
    zn=0
    cn=0

calculate coefficients for interval
    cj=zj-mu_j c_(j+1)
    bj=(a_(j+1)-a_j)/h_j(c_(j+1)+2c_j)/3

    dj=(c_(j+1)-c_j)/3hj

for each node o to be interpolated
    find the appropriate interval k
    s_i(o)=a_i + b_i(o-x_i) + c_i(o-x_i)^2+d_i(o-x_i)^3
```

5 Results and Findings

Here, real values are plotted as yellow and the interpolation curve is plotted as blue. We can see that the interpolation curve is very smooth and could account for some fluctuation and changes in rent, but we can also see that some of the interpolated values are very off from the real value.

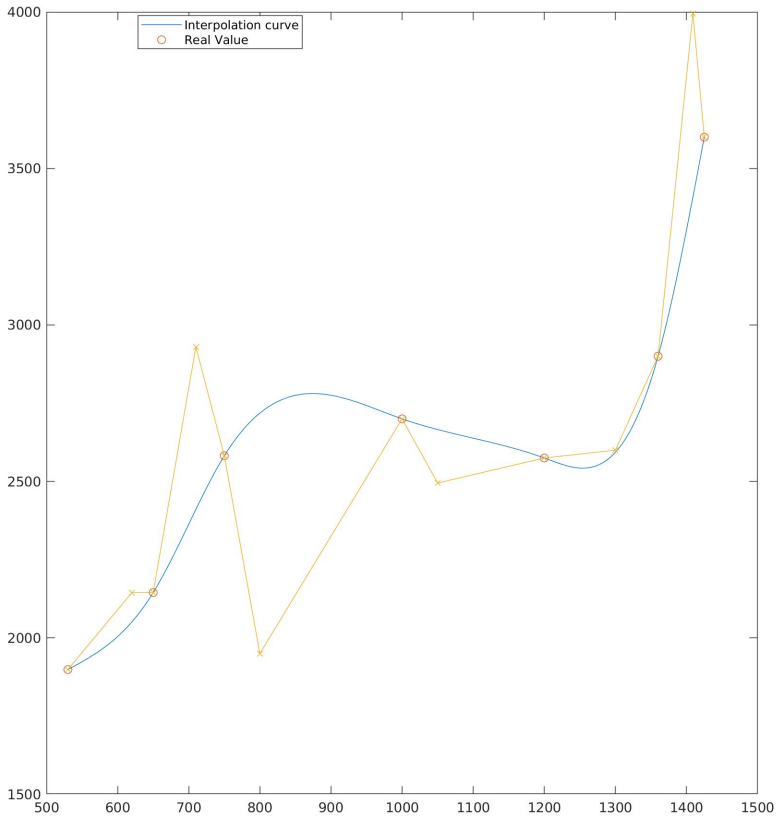


Figure 1: interpolation curve and real values

To further examine our interpolated values, we calculated the relative error and plotted them against area in square feet. we can see that for some apartments, the relative error could be as small as less than 5 percent, but for some it could be as large as 40 percent. This is probably due to the fact that rent is not only dependent on the area, and thus the interpolation could not work as well on some apartments.

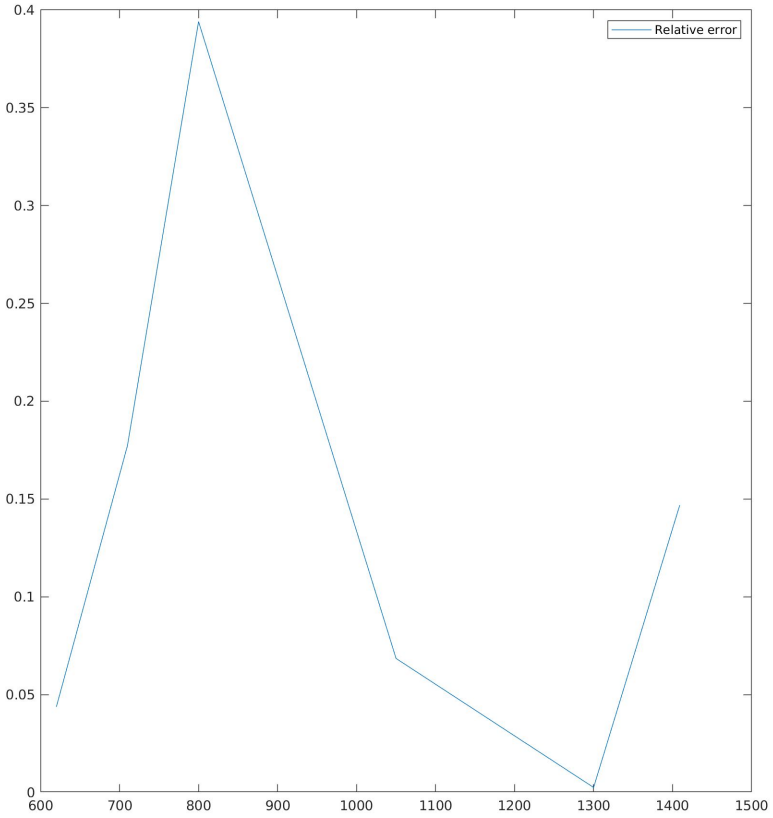
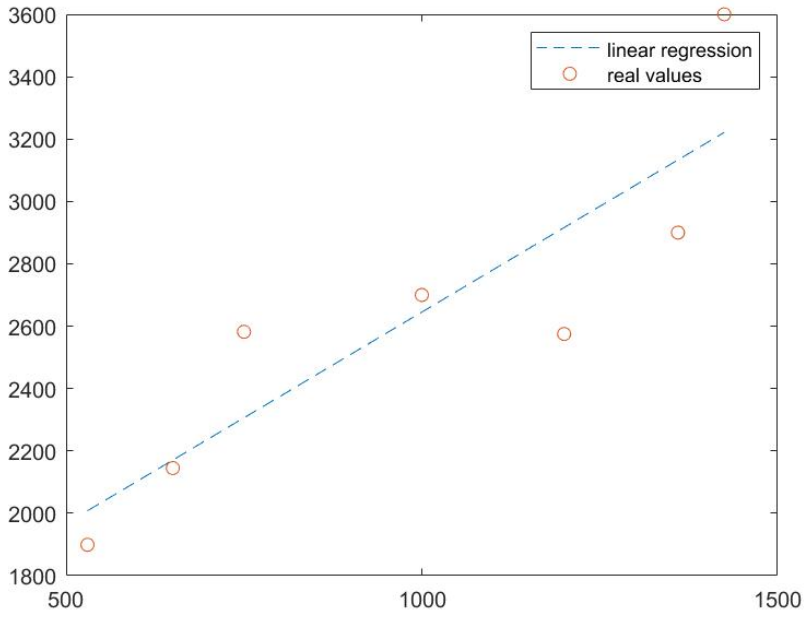


Figure 2: sqft and relative error

confirming with Linear Regression

To improve our accuracy, we also fitted a single-variable linear regression model on the training and testing data set. Since this is a cubic spline project not a linear regression project, we will skip the introduction to linear regression. Here we have the results of our linear regression model with one variable:

From these results, we can see that the linear regression method doesn't improve our accuracy. Due to the relative lack of data on Zillow, we were unable to further improve our model accuracy.



linear regression on training data set

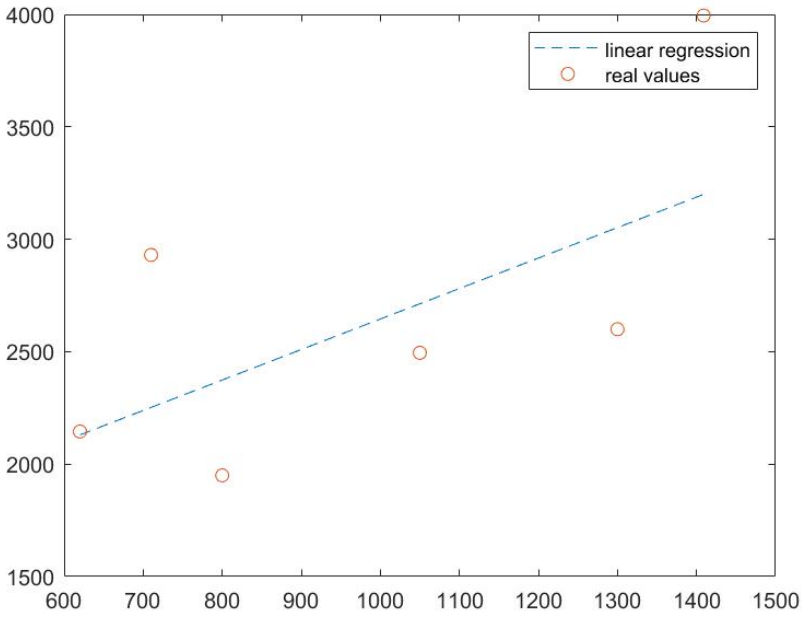


Figure 3: linear regression on testing data set

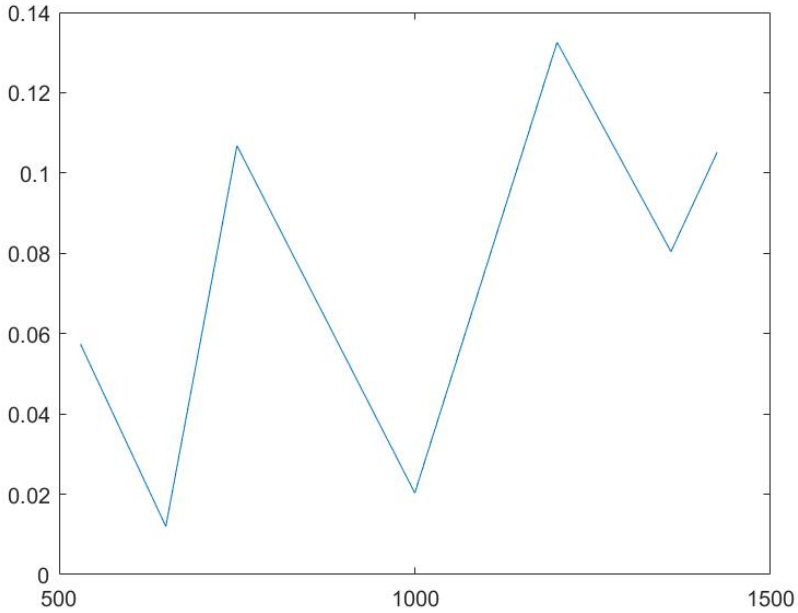


Figure 4: relative error of linear regression on training data set

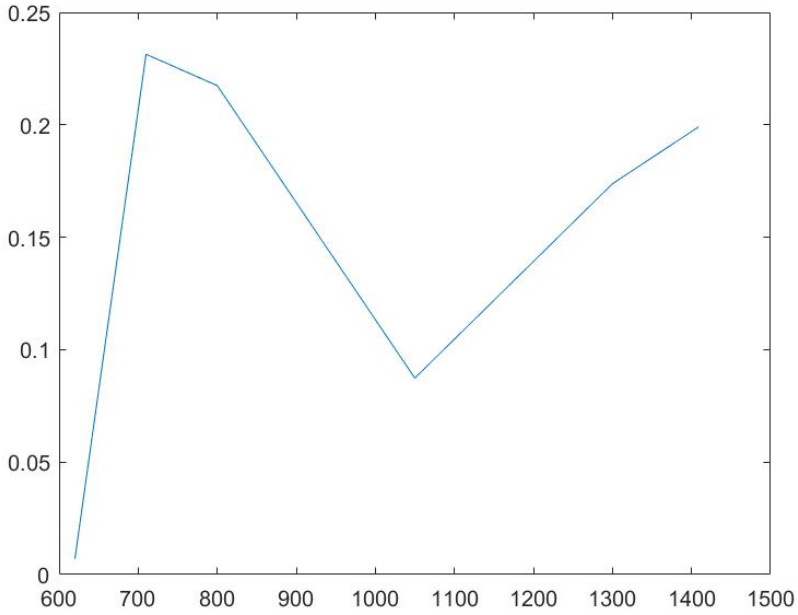


Figure 5: relative error of linear regression on testing data set

6 Discussion

From the results, we can see that the cubic spline interpolation method works well on some data but not so well on others. We have two possible explanation for this. First, we know that the price of an apartment is not only dependent on the area; it is also dependent on the location of the apartment, how new the apartment is, the landlord of that apartment and so on. This is confirmed by the low accuracy rate for our linear regression model, as the price of the apartment does not depend completely on the sqft area of the of said apartment.

Second, we can say that there is a lack of data to adequately model and interpolate. This is rather unfortunate because a lot of the apartment available on Zillow do not have area listed. Without the area listed, we can not incorporate that into our model. If we were to have more data, our model might be more accurate. In the end, we can say cubic spline interpolation is a very powerful tool but has to be used cautiously. It does not work too well on data that is dependent on too many other variables, and it does not work too well on data in low quantity.

7 Conclusion

It is fair to say that cubic spline interpolation is a very strong and powerful tool that offers decently accurate results. However, we can also see that cubic spline interpolation also has a major short coming, and that is poor performance in small data set and in data with too many dependent variable. Nevertheless, we would say cubic spline interpolation is a powerful tool to use for interpolation but we should not blindly use it.

8 References

Zillow.com. www.zillow.com/westwood-los-angeles-ca/rentals/.

Burden, Richard L., et al. Numerical Analysis. Cengage Learning, 2016.

Hundley, Douglas. Cubic Spline and Matlab, Whitman College, 13 Apr. 2009, people.whitman.edu/ hundledr/courses/M467/CubicSplines.pdf.

“Cubic Spline.” Cubic Spline - an Overview — ScienceDirect Topics, Science Direct, www.sciencedirect.com/topics/engineering/cubic-spline.

Appendix: MatLab code

cubic spline methods

```
1
2  x=[530,650,750,1000,1200,1360,1425];
3  a=[1899,2145,2582,2700,2575,2900,3600];
4  n=length(x)-1;
5
6   m = n - 1;
7   h = zeros(1,m+1);
8   for i = 0:m
9     h(i+1) = x(i+2) - x(i+1);
10  end
11
12  aa = zeros(1,m+1);
13  for i = 1:m
14    aa(i+1) = ...
          3.0*(a(i+2)*h(i)-a(i+1)*(x(i+2)-x(i))+a(i)*h(i+1))/(h(i+1)*h(i));
15  end
16
17  xl = zeros(1,n+1);
18  xu = zeros(1,n+1);
19  xz = zeros(1,n+1);
20  xl(1) = 1;
21  xu(1) = 0;
22  xz(1) = 0;
23
24  for i = 1:m
25    xl(i+1) = 2*(x(i+2)-x(i))-h(i)*xu(i);
26    xu(i+1) = h(i+1)/xl(i+1);
27    xz(i+1) = (aa(i+1)-h(i)*xz(i))/xl(i+1);
28  end
29
30  xl(n+1) = 1;
31  xz(n+1) = 0;
32  b = zeros(1,n+1);
33  c = zeros(1,n+1);
34  d = zeros(1,n+1);
35  c(n+1) = xz(n+1);
36
37  for i = 0:m
38    j = m-i;
39    c(j+1) = xz(j+1)-xu(j+1)*c(j+2);
40    b(j+1) = (a(j+2)-a(j+1))/h(j+1) - h(j+1) * (c(j+2) + 2.0 * ...
          c(j+1)) / 3.0;
41    d(j+1) = (c(j+2) - c(j+1)) / (3.0 * h(j+1));
42  end
43
44 %creat subintervals to draw the interpolation plots
45 t=zeros(1,1);
46 cubict=zeros(1,1);
47 j=1;
48 for i=1:n
49     for k=x(i):x(i+1)
```

```
        t(j)=k;
        diff=t(j)-x(i);
        cubict(j)=a(i)+b(i)*diff+c(i)*diff^2+d(i)*diff^3;
        j=j+1;
    end
end

l=[530,620,650,710,750,800,1000,1050,1200,1300,1360,1409,1425];
f=[1899,2145,2145,2930,2582,1950,2700,2495,2575,2600,2900,3995,3600];


li=zeros(1,1);
j=1;

for i=1:n
    for k=1:2
        diff=l(j)-x(i);
        li(j)=a(i)+b(i)*diff+c(i)*diff^2+d(i)*diff^3;
        j=j+1;
    end
end

li(length(l))=a(length(a));

for i=1:length(li)
    rel_error(i)=abs(li(i)-f(i))/f(i);
end
figure(1)
title("relative error and sqft")
rel_error=nonzeros(rel_error);
lr=[620,710,800,1050,1300,1409];
plot(lr,rel_error);
figure(2)
title("cubic interpolation and true values")
plot(t,cubict)
hold on
plot(x,a,'o')
hold on
plot(l,f,'-x')
hold off
```

linear regression

```
 x=[530,650,750,1000,1200,1360,1425];
a=[1899,2145,2582,2700,2575,2900,3600];
x=transpose(x);
a=transpose(a);

l=[620,710,800,1050,1300,1409,];
f=[2145,2930,1950,2495,2600,3995];
l=transpose(l);
f=transpose(f);

X = [ones(length(x),1) x];
```

```
b = X\a;

figure(1)
train = X*b;
title("Linear Regression on Training Data")
plot(x,train,'--')
hold on
plot(x,a,'o')
legend('linear regression','real values')
hold off

figure(2)
L = [ones(length(l),1) l];
test=L*b;
title("Linear Regression on Testing Data")
plot(l,test,'--')
hold on
plot(l,f,'o')
hold off
legend('linear regression','real values')

for i=1:length(train)
    rel_errortrain(i)=abs(train(i)-a(i))/a(i);
end
figure(3)
title("relative error and sqft")
plot(x,rel_errortrain);

for i=1:length(test)
    rel_error(i)=abs(test(i)-f(i))/f(i);
end
figure(4)
title("relative error and sqft")
plot(l,rel_error);
```