Predicting housing price

Shu Niu

Introduction

The goal of this project is to produce a model for predicting housing prices given detailed information. The model can be useful for many purpose. From estimating the worth for a house that's not on market, to figuring out which component factors the most into a house, which can help decide if a remodel makes sense financially.

Data collection

For the purpose of this project, I have written a script to crawl recently sold listings on Zillow.com to obtain details about houses sold in the last 3 years. I am focusing my data set on the South Bay area, with the expectation that they adhere to some consistent pattern and trend.

Here is a sample listing I collected from Zillow:
Address: 2042 Calle Mesa Alta, Milpitas CA 95035
Sold Date: 2013/11/08
Sold Price: 685,000
Built Year: 1990
Interior Size: 1785 sqft
Lot Size: 1785 sqft
Bedroom: 3
Bathroom: 2.5

These information needs to be translated into numbers so they can be fed into a model. Most of the components are straight forward:
Sold Date: represented as number of months since 2010/01
Sold Price: represented as number of thousand dollars
Interior Size, Exterior Size: number of square foot
Bedroom: number of bedroom
Bathroom: i tried to deal with only integers in the input, so this component is converted to an integer that equals to double the number of bathrooms

Address is hard to quantify directly. I have considered using the latitude and longitude as data points, but there's no simple trend like when latitude increases, we expect the house value to rise. I do expect there to be some trending pattern for the intrinsic value of the location, but the contour of that would be too complex to model using the limited data set I have. I do however expect the contour to form around city boundaries and school districts. So I am using the following two values to represent the location:

- Median price of the city: for each city I have collected at least 400 recent sales, and the

median price will be used as a data point. This can be used to capture the base value for each city. I could have also used {city, zipcode} as the unit here. However, I won't have hundreds of data point for each zipcode.

- School API score: I have also crawled schoolandhousing.com for each recently sold listing to collect a school score that ranges from 0 to 100. The score is computed based on the API scores for the elementary school, middle school, and high school the house is assigned to. So it should be a fair estimate for the school value portion of the house.

Overall I have collected 5000 data points, each one can be represented as $<x^i, y^i>$:

- $x^i = <x^i_1, x^i_2, x^i_3, x^i_4, x^i_5, x^i_6, x_7, x^i_8>$
  - x^i_1 = sale months since 2010/01
  - x^i_2 = interior size
  - x^i_3 = lot size
  - x^i_4 = built year
  - x^i_5 = number of bedrooms
  - x^i_6 = number of bathrooms times 2
  - x^i_7 = school api score
  - x^i_8 = median price of the city
- y^i = price (in thousand dollars)

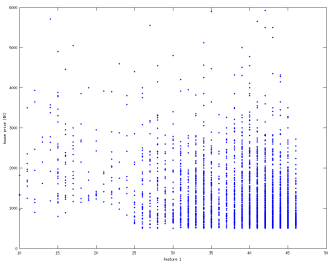
For example, we would represent the sample listing above as

- x = <46, 1990, 1785, 1990, 3, 5, 88, 675>
- y = 685

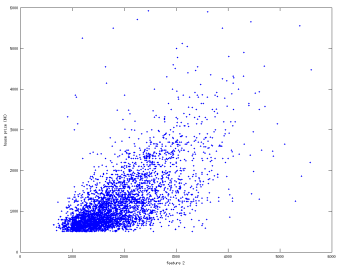
Data Analysis

Next I wanted to look at the distribution of the data in each dimension, comparing it to the y value to see if there's any immediate trend jumping out from the graph. Intruitively I would expect for each x_j dimension, the higher the x_j value, the higher the y value.

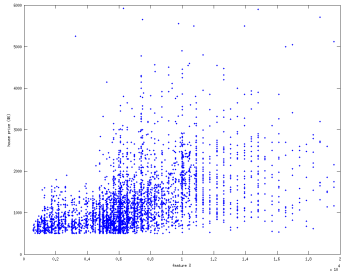
Here is a visual representation of each x_j vs y.



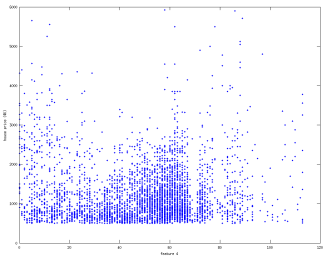
sale month vs price



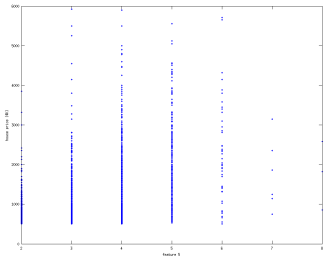
interior size vs price



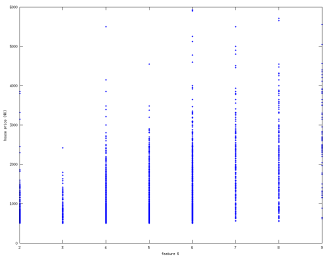
lot size vs price



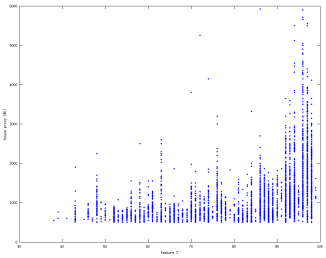
built year vs price



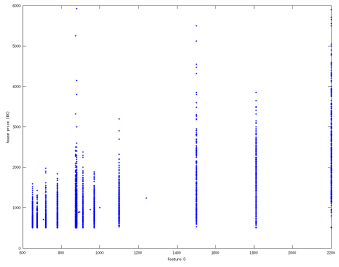
bedroom vs price



bathroom vs price



school score vs price



city median price vs price

From the graph, interior size and school score are the two that shows the strongest correlation with price. The other factors are not as obvious, as there are also expensive houses with low x_j values.

Linear Regression

My first hypothesis is a linear model, where it would predict $y = a^T x$, and $x = <1, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8>$

To test the errors of this hypothesis, I divided my dataset into 10 groups to perform a 10-fold cross validation. Each time using 9 groups to train the linear regression model, and then use the 10th group for testing. The error is calculated to be the average error in the test. The set that produced the smallest error is used as my final model. The resulting model had an error of 322 in the testing group.

The surprising finding is that there are two coefficients that turn out to be negative: a_4 and a_5, corresponding to built year and number of bedrooms. And they are consistently negative in all 10 cross validation groups.

For built year, the intuitive expectation is for the same specification of a house, the more recent ones would cost more. However the least square fitting shows a negative slope between built year and price. It could be that among the data I collected, most of the expensive houses are the older ones, and their value is captured by another dimension that's not part of our data set. Another conjecture is that the linear model is a bad-performing model at least for the year dimension. There's a significant difference in value between a brand new house and a 10 year

old house, but almost no difference in value between a 50 year house and a 60 year old house. With the linear model, it cannot differentiate between those scenarios.

Number of bedrooms is also trending negatively with price. This is surprising at first, but it can be explained. Because interior size is already a dimension in the data. Given the same interior size, more bedrooms means smaller footage per room, thus most likely to be a budget house than a luxury house.

Normalization

Next, I applied a normalization to all the $x_i^{<j>}$, where
$x_{i,normalized}^{<j>} = (x_i^{<j>} - \text{mean of } x_i) / \text{standard deviation of } x_i$

Then applying linear regression again on the normalized data set gives the following values for the coefficients a: <1209, 107, 282, 100, -34, -33, 50, 55, 371>
Since all the x_i values are normalized, we can compare their coefficients to get a sense of how much each dimension contributes to the price. Based that, we rank the dimensions from the most important to least important as:

- x_8: median city price (371)
- x_2: interior size (282)
- x_1: sold month since 2010/1 (107)
- x_3: lot size (100)
- x_7: school score (55)
- x_6: # bathroom (50)
- x_4: built year (-34)
- x_5: # bedroom (-33)

We all know that location is a very important factor in housing price, and the median city price being the most prominent factor also reflects that. There is a caveat that the median city price is generated from the price value of the same data set where we evaluate our model. This could mean the median price is already adjusted to fit the data set well. A more unbiased indicator would be getting the median city price from a census data rather than the same data set.

Component Analysis

A different way of doing the component analysis is to do a backward search: remove one dimension each time and see what's the error of the new model, and then pick the dimension that leaves the smallest error. By doing so, these would be the order of dimensions we take away (with the resulting error in model after taking away that dimension):

- x_5: # bedroom (323)
- x_6: # bathroom (324)
- x_4: built year (325)
- x_7: school score (327)

- x_1: sold month since 2010/1 (335)
- x_3: lot size (346)
- x_2: interior size (448)
- x_8: median city price is the last dimension remaining

I also did a forward search for the most prominent dimensions: starting from 0 dimension, each time adding a dimension that leaves the smallest error in the model. It resulted in the same order of components as the backward search.

Comparing these with the order found in normalization, the ordering is almost identical. The only ones that are switched around are the ones that have very close coefficients in normalization and very close additional errors in backward search. This also solidifies the conjecture that location is the most important factor, followed by size of house.

Quadratic Model

The final model I used is a quadratic normalized model.
I converted each data point from <1, x_1, x_2, x_3… x_8> to
<1, x_{1n}, x_{2n}, x_{3n}… x_{8n}, x_{1n}*x_{1n}, x_{1n}*x_{2n}, x_{1n}*x_{3n}… x_{8n}*x_{8n}> where x_{in} is x_i normalized to number of standard deviations from mean. This expands the x dimensions from 9 to 45, which should still be model-able using our 5000 data points.

Using the same 10 fold cross validation as linear model, I find the error for the quadratic model is 271, better than the error of 321 produced in linear model. So this indicates that quadratic function is a more fitting model in this case.

By comparing the coefficients of all the quadratic terms, the following are the top 3 combinations that had the highest coefficients (in brackets):

- x1 * x7: sold month * school score (42)
- x7 * x8: school score * median city price (39)
- x2 * x8: interior size * median city price (39)

Most of these are also the most valuable dimensions seen in the linear model, noting that school score is having an increased effect here.