International Journal of Smart Home
Vol.8, No.2 (2014), pp.191-200
http://dx.doi.org/10.14257/ijsh.2014.8.2.19

The Research on Price Prediction of Second-hand houses based on KNN and Stimulated Annealing Algorithm

Weikun Zhao, Cao Sun and Ji Wang

School of Computer Science and Technology, Harbin institute of technology, Harbin, Heilongjiang, China
williamzwk@hotmail.com

Abstract

Second-hand housing market is the barometer of the real estate market since the buyers of second-hand houses usually are those who really want to live, whereas financial investment and speculation are not their goals. So the price and determinants of second-hand houses reflect the real demand of housing market. In this paper KNN related algorithms are applied to study the problems associated with price of second-hand house. It includes using KNN and weighted-KNN algorithms to predict the price, using cross validation method to compute average deviation of prediction algorithm and compare KNN's prediction effect with weighted-KNN's, and using stimulated annealing optimization algorithm to compute the weight values of house attributes and evaluate the relative importances of them. Through the analysis of attribute importance it can show the influences of different house attributes on house price and the main concerns of buyers. All these results can give valuable information for manages, decision makers and appraisers of real estate.

Keywords: *Second-hand housing; KNN; Cross validation; Stimulated annealing*

1. Introduction

With the rapid development of real estate market Chinese, first-tier cities housing prices are rising and remain high. But incomplete market information, non standardized market behavior, the scarcity of land, and the imperfect economy system result in property bubble gradually emerged [1]. The purchasing power of citizens remains relatively low, but property developers seek to maximize profits, so the supply and demand imbalance of housing market appears. Now secondary housing market is being paid more and more attention.

On the real estate market dominated by commodity housing the price is the main market index [2]. When supply and demand of first-hand housing have problems, secondary commercial housing prices are more close to reflect the real estate prices and the changes of sales and prices of second-hand housing can indicate the real market demand [3, 4].

From the price point of view, the transaction prices of second-hand housing prices generally come from calm thinking, calculating and economy game of supply and demand sides. No matter what kind of pricing principles, market always plays a decisive role [5, 6]. Because it is close to real situation, second-hand housing price has great reference significance for determining the prices of first-hand and government-subsidized housing. The reasonable pricing strategy will have a fundamental influence on the success of the transaction. So in this paper we will study reasonable second-hand housing pricing strategy. There are four sections in the paper. The first section will give the description of our housing data and basic KNN algorithm. In the second section the weighted-KNN is introduced and the cross validation is used to evaluate and compare the prediction effect of KNN and weighted-

ISSN: 1975-4094 IJSH

KNN. The third section uses stimulated annealing algorithm to weight other attributes except for price and gives the result analysis. In the fourth section we summarize the paper.

2. Data Set and KNN Algorithm

2.1. Data Set Description

Our second-hand housing data come from two districts: "AiJian" and "Highway Bridge" of Harbin, the capital of Heilongjiang province in China. We collect 1086 records of houses data. Each record contains 17 attributes (or fields). Table1 describes details of these attributes. Our main task is research how to predict the last attribute "Price" based on other attributes. In the process of collecting data there many attributes having description of string, but the algorithms we study need numerical values so some conversions are needed. The details of conversions are shown in the table. The basic principle is to assign an integer value to a string value of an attributes according how many records that "string value" classifies in whole data set.

Table 1. The Description and Value of Data Fields

Attribute Name	Description	Value
Bedroom(BR)	The number of bedrooms.	A numerical value
Living room(LR)	The number ofbedrooms	A numerical value
Kitchen(K)	The number of kitchens	A numerical value
Bathroom(T)	The number of Bathroom	A numerical value
GFA	Gross Floor Area	A numerical value
NFA	Net Floor Area	A numerical value
Year(Y)	Year	A numerical value
Layer(L)	Layer	A numerical value
Heigh(H)	Height of the building	A numerical value
Orientation(O)	Orientation of the house	A numerical value. The relation of orientation and the numerical value: {**North:1, East:2, West:3, Northeast:4, Northwest:5, Southwest:6, East-west:7, Southeast:8, South:9, North-south:10**}. *The greater number means the corresponding item has more instances in sampling. (The same below)
Structure(S)	Structure of the house	A numerical value. The relation of orientation and the numerical value:**{Duplex apartment:1, ,Others:2, Split-level:3, Duplex:4, General:5}**.
Renovation (Ren)	Renovation types	A numerical value. The relation of orientation and the numerical value:**{middle:1, luxury:2, Rough:3, simple:4, refined:5}**.
Residential category (Res)	Residential category	A numerical value. The relation of residential category and the numerical value:**{ Affordable housing:1, Commercial and residential building:2, Apartment:3, Ordinary residential:4}**.

Building types (BT)	Building types	A numerical value. The relation of building types and the numerical value:**{Bungalow:1, Brick building:2, Tower:3, Brick and Concrete :4, Steel and concrete:5}**.
Property right (PR)	Property right	A numerical value. The relation of building types and the numerical value:**{Commercial residential building:1, Others:2, Bills in three parts:3, Personal property:4}**.
Facilities description (FD)	The supporting facilities.	The numerical value of facilities of house.
Price(P)	House price	The numerical value of house price.

2.2. KNN Algorithm

Bayesian classifiers, decision tree, and support-vector machines are not the best method when they are used in numerical price prediction. In this section we introduce KNN (k-nearest neighbors) algorithm [7, 8].

As the name suggests, there two important points to understand this algorithm: "K" and "nearest"

1) The measure of the "nearest"

The nature of KNN algorithm is to find out the "nearest" neighbor items to an item that has some attribute to be predicted, *e.g.*, "price" attribute in this research. The "nearest" also means "most similar", "closest" *etc.*, and how to represent this measure is a key problem. Inspired by geometry we can use Euclidean distance as the metric. Although usually it is often used in 2D and 3D geometry, it can also be used in higher dimensions. This extension is easy to understand and operate. For example in 2D there are two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, then the Euclidean distance d between them is,

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The data in Table 2 is a segment of data set. The field names and values are described in Table 1.

Table 2. A Segment of Data Set

BR	LR	K	BaR	GFA	NFA	Y	L	H	O	S	Ren	Res	BT	PR	FD	P
2	1	1	1	85.12	67.89	2005	15	16	8	5	5	4	7	4	18	72
2	1	1	1	69.9	50	1996	2	7	7	5	1	4	7	4	18	51.8
3	1	0	2	140	95	2005	6	7	10	5	4	4	1	4	19	94
2	1	1	1	62	42	1993	2	7	7	5	4	4	7	4	0	46
1	1	1	1	66	48	1987	6	7	6	5	5	4	7	4	0	42
2	1	1	1	81.9	54.6	1993	3	8	10	5	5	4	7	4	18	60
2	1	1	1	74.2	53	1994	3	7	5	5	1	4	7	4	12	51
2	1	1	1	61.09	41	1989	2	6	10	5	1	4	7	4	12	41
2	1	1	1	132	93	2001	7	8	10	5	4	4	7	4	15	85
1	1	1	1	3800	3600	1992	1	3	10	5	5	4	7	4	15	4500
2	1	1	1	99	66	2002	2	7	8	5	2	4	7	4	5	77
2	2	1	1	174	128	2009	5	30	8	5	5	4	7	4	29	190

The record in third line can be represented as P_3 (3, 1, 0, 2, 140, 95, 2005, 6, 7, 10, 5, 4, 4, 1, 4, 19, 94) and the fourth line as P_4 (2, 1, 1, 1, 62, 42, 1993, 2, 7, 7, 5, 4, 4, 7, 4, 0, 46), so the d can be computed as above,

$$d = \sqrt{(BR_4 - BR_3)^2 + (LR_4 - LR_3)^2 + (K_4 - K_3)^2 + (T_4 - T_3)^2 + \cdots + (P_4 - P_3)^2}$$
$$= \sqrt{(2-1)^2 + (1-1)^2 + (1-0)^2 + (1-2)^2 + \cdots + (46-94)^2}$$

2) The meaning of k

Based on the measure of similarity we can choose the k items that have the most nearest d as the k nearest neighbors of current item. When KNN is used in prediction the task usually is to predict the value of some attribute based on the known values of other attributes. In this research our main task is to predict the reasonable housing price 'P' which lies in the last column in table2.

Except the 'P' attribute we need give other attribute values, and use these values to compute d, *i.e.*, in the above computation the last item $(P_4 - P_3)^2$ should not be included. After get the k nearest neighbors we use the average value of the prices of the k neighbors as the price of the item that we want to predict.

The KNN algorithm is described as below. Suppose that the data record format is: {[a_1, a_2, … , a_n], a_p}, 'a_i' represent attribute i. The corresponding values are represented as {[v_1, v_2, … , v_n], v_p}, v_p is the attribute value to be predict, and [v_1, v_2, … , v_n] are the input values other attributes.

KNN algorithm:

input: [v_1, v_2, … , v_n],
[r_1, r_2, … , r_m], data_set （set of all known data），
k, number of neighbors.

output: v_p

Step1. Compute all distances ds between [v_1, v_2, … , v_n] and each item in data_set and get ds = [d_1, d_2, … , d_m], m is the number of items in data_set.

Step2. Sort ds in ascending order [$d_{1'}$, $d_{2'}$, … , $d_{m'}$] and get the corresponding k items based on the indices of first k distance value, i.e. r = [$r_{1'}$, $r_{2'}$, … , $r_{k'}$].

Step3. Compute the average value of v_p.

$$v_p = \frac{1}{k}\sum_{i}^{k} r_i(a_p)$$

When k=1 the $v_p$is the nearest neighbor corresponding attribute value. Either too few or too many neighbors all will have a big effect on the v_p. Usually k is greater than 1 and has different values according to different data sets and problem domains. In this research k is set to 3.

KNN algorithm is easy to understand and implemented. For example now we have a new second-hand house in the near district to sell and the values of 16 attributes (in the first 16 columns in table1) [2.0, 1.0, 1.0, 1.0, 69.9, 50.0, 1996.0, 7.0, 7.0, 3.0, 1.0, 2.0, 0.0, 2.0, 1.0, 18.0] and k=20 are input into the algorithm, after computation we get $v_p = 44.6$.

3. Weighted-KNN¬¬ and Cross Validation

As stated earlier k is an important parameter. In KNN algorithm step3 the formula computes the average just by divided by k, *i.e.*, each nearest neighbor' contribution to prediction is treated as equivalent. But a more reasonable deduction is that the farther distance gives lower contribution. Figure 1 gives an illustration showing this idea. In this example k=3, n1, n2, and n3 are the k nearest neighbors of n0. D1, d2, and d3 are the distances and the corresponding weight values w1, w2, and w3 are used to measure contribution. Since $d_1 \leq d_2 \leq d_3$, based on description above, $w_1 \geq w_2 \geq w_3$。 The relation of distance and weight value can be expressed by Gaussian Function [9].

$$\text{weight} = g(\text{distance}) = ae^{\frac{-(\text{distance}-b)^2}{2c^2}} + d$$

a, b, c, d are real constants, and $e \approx 2.7182$…. Figure 2 shows this function.

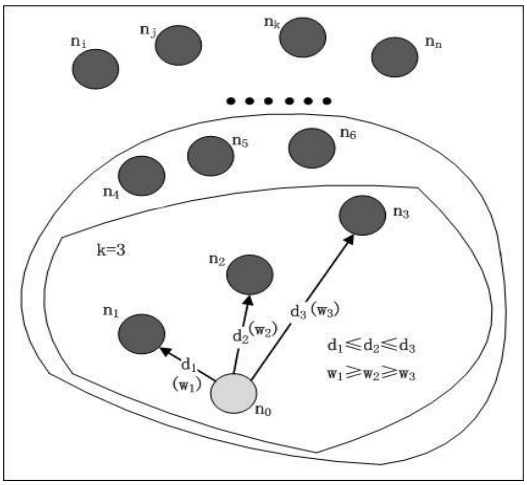


Figure 1. Weighted-KNN Illustration

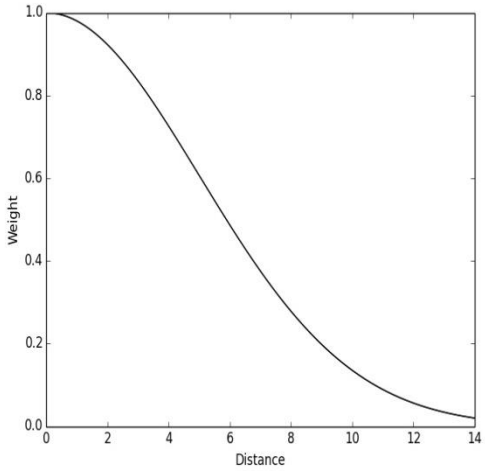


Figure 2. The Relation of Distance and Weight

Weighted–KNN algorithm is similar to KNN. The main difference is in step3. In weighted–KNN the formula should be

$$v_p = \frac{1}{k}\sum_{i=1}^{k} g(d_i) r_i(a_p)$$

In the following test KNN and weighted–KNN algorithms are compared. We randomly choose 4 records, r1, r2, r3, and r4, in the comparison different k's values are chosen: k=1, k=2, k=3, k=4, k=5.

r1 =[2, 1, 0, 1, 85.12, 67.89, 2005, 15, 16, 8, 5, 4, 4, 1, 4, 19, 68]
r2=[1, 1, 1, 1, 62.3, 45, 2005, 13, 30, 8, 5, 1, 4, 7, 4, 41, 63.8]
r3 =[3, 1, 1, 2, 183.38, 123.07, 2012, 20, 32, 10, 5, 3, 4, 7, 3, 12, 315]
r4 =[2, 1, 1, 1, 64, 43, 2014, 1, 30, 10, 5, 3, 4, 7, 4, 21, 75]

For each record the first 16 attributes values are input into KNN and weighted-KNN algorithm respectively and the corresponding predicted values are returned. The comparison features are shown in Figures 3 to 7. In each figure the x-axis represents k's values and y-axis

represents the real and predicted values. The straight line is for the real value. The dot line and dash line are for predicted values of KNN and weight-KNN algorithm respectively.

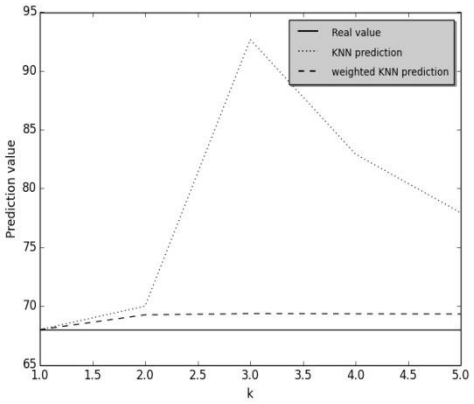


Figure 4. The Comparison Result for r1

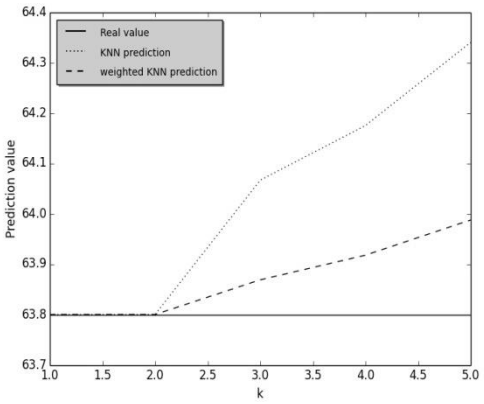


Figure 5. The Comparison Result for r2

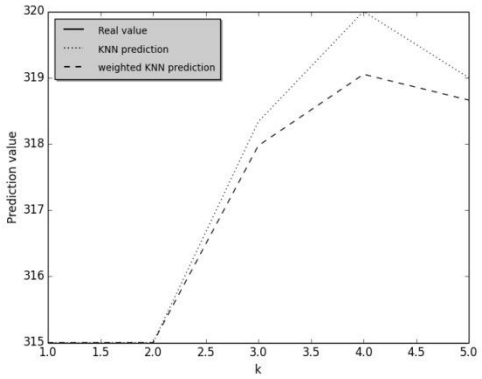


Figure 6. The Comparison Result for r3

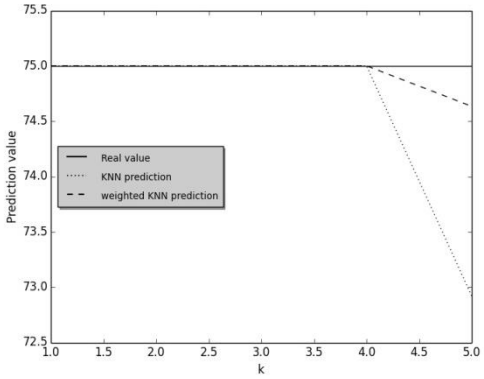


Figure 7. The Comparison Result for r4

Obviously in all four situations the weighted-KNN predictions are more precise than KNN. In this example only 4 records are chosen, next we will further verify the result by cross-validation [10] the data set is divided into two sets: train set and test set. The idea is easy to understand: use data in train set to predict records in test set and compute the deviation of predicted value and real value (in the test set real value of record is known). The average deviation will give the approximate prediction of applied algorithm. The different division will generate different average deviation so it always needs to run many times to approaching a reasonable estimation. Figure 8 gives the cross validation comparison results of KNN and weighted-KNN algorithms. The x-axis also represents different k's values and y-axis represents the corresponding average prediction deviations or errors. The dot line and dash line represent average prediction errors of KNN and weighted-KNN respectively. The results indicate that weight-KNN is more accurate than KNN. We can also find the average error increases with the k'value growth.

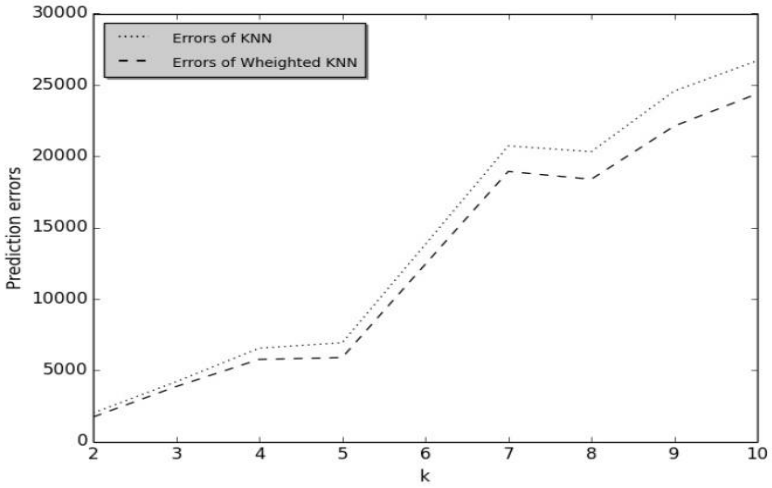


Figure 8. The Cross Validation Results of KNN and Weight-KNN

Although for our housing price data set the prediction effect of weighted-KNN generally is better than KNN's, it can't say weighted-KNN is always better than KNN. For the different data sets the weight values and chosen k are different and usually this can have significant influence on predication effectiveness. For a specific data set the weight values the weight values can be determined by test just as we did in above experiment.

Now we have an effective way to predict the housing price but there remains a problem. In weighted-KNN the neighbor's price weight value is the Gaussian function of distance and the value describes the degree of importance of distance, but what the importances of attributes or their relative importances are is an interesting problem. Through the analysis of relative importances we can answer which attributes are more important than others and know the factors that buyers pay more attention to when they choose second housing to buy.

4. Stimulated Annealing Algorithm

There are still two problems in our price prediction.

(1) In the data fragment shown in Table 2 the values in the 'Y' (year) column are obviously greater than values of other columns. In KNN or weighted-KNN algorithm the measure of difference between two records is Euclidean distance so though the value is greater, the subtraction will eliminate the kinds of effect. For example in the first two rows in Table 2 the difference of 'Y' values is 2005-1996=9 or 1996-2005=-9 whereas the difference of 'L' is 13 or -13, instead it is greater. But when both attribute value and difference value of some attribute are obviously greater than other attributes, only using subtraction can't eliminate the effect.

(2) In weighted-KNN algorithm the distance is associated with a weight value. Now if we want to compare the relative importances of attributes, it is also necessary to give appropriate weight values to corresponding attributes. In Figure 1 the relation of weight and distance is intuitive but there are 16 attributes except for "price" and the relative importances of them are hard to perceptible. So how to weigh these attributes is an interesting topic

Optimization algorithms can be used to solve these problems. Many optimization algorithms can achieve this goal such as genetic algorithm, stimulated annealing [11, 12],

particle swarm algorithm and so on. For case (1) though in our house data set the subtraction can eliminate the effect, on many other cases appropriate weight values determined by optimization algorithms will help reduce influences of some attributes on prediction results.

For case (2) we now use the annealing optimization algorithm[13] to find the relative importances of house attributes on price prediction.

First it needs to set the ranges of weight values of all attributes. In the house data there 16 attributes so we set 16 ranges,

$$\{attribute_1: [0,100],\ attribute_2: [0,100], \ldots,\ attribute_{16}: [0,100]\}.$$

For simplicity all the ranges are same. In each range 0 represents minimum weight value that means the corresponding attribute has no effect on price prediction and 100 represents maximum weight value meaning greatest influence on the prediction result. The value between 0 and 100 can be used to measure the relative importances of other attributes.

Then for each attribute a random initial weight value in range [0,100] is set and all of them together are fed into annealing algorithm as one parameter. The other parameters can be set and chosen based on specific situation. The result of every running is usually different. For our house data one running result is shown in Table 3.

Table 3. The Weight Values of Attributes after Annealing Algorithm

id	Attribute Name	Weight Value	id	Attribute Name	Weight Value
1	Bedroom(BR)	86	9	Height(H)	81
2	Living room(LR)	25	10	Orientation(O)	59
3	Kitchen(K)	29	11	Structure(S)	54
4	Bathroom(BaR)	58	12	Renovation (Ren)	45
5	GFA	90	13	Residential category(Res)	0
6	NFA	52	14	Building types (BT)	62
7	Year(Y)	86	15	Property right (PR)	82
8	Layer(L)	67	16	Facilities description(FD)	52

In this result the higher weight value means that it will have more influence on prediction of price. For **"Bedroom"**, **"Living room"**, **"Kitchen"**, and **"Bathroom"** it is obvious that the weight of bedroom in house is 86 indicating more influence than other three attributes. It is easy to understand, for our second-hand housing data set is mainly consisted of residential buildings and in this kind of house it usually has more than one bedroom and the space of bedroom is often greater.

"GFA" is assigned the weight 90 which is the greatest weight value in all attributes. The reason is to some extend that when computing the house price the direct method it is to multiply GFA and the price of unit area, so the influence is direct.

The weight value 86 of **"Year"** is also high. In China the housing property is valid for 70 years. The houses information we collect in our data set comes from two newer districts "AiJian" and "Highway Bridge" in Harbin. Although in the two newer districts the years of all houses are not too long, the high weight of year comes from consumption psychology and it is obviously when other attributes are equal the newer house is more attractive to buyers.

"Layer" and **"Height"** usually are two related attributes and always are considered together. In most cases the price of top layer is lower than other layers. The difference of

weight value 67 and 81 is small. The difference may come from the price mechanism, since except for the top layer with the increase of layer the price of unit area also increases so that total price increases. For example in a building in the two districts the price of unit area of mid-height is a, when increasing one layer the price adds 20/m^2 and when decreasing one layer the price subtract 20/m^2.

For **"Orientation"**, **"Structure"** and **"Renovation"**, orientation' weight value 59 is greater and this reason why orientation is more important is relative with local environment. In Harbin, the northeast city in China, the winter is cold and summer is hot. So in order to have a warm room in winter the south orientation may be the first choice and in summer the good exposure structure is the first choice, in brief the orientation is a the result of comprehensive consideration and usually is paid more attention than other two attributes. The extent of renovation can give the buyers initial confidence in house quality but they always redecorate houses after buying, so the weight value is lower than others.

"Residential category" has weight value 0 but this doesn't mean it is not important. The reason comes from our house data. In the data set there are four residential categories and the percentages are: Affordable housing: 0.0920%, Commercial and residential building: 0.8280%, Apartment: 5.7958%, and Ordinary residential: 93.1923%. The greater percentage 93.1923% is the main reason that causes annealing to converge at 0. It also can be understood from the limit. If an attribute has only one kind of value, *i.e.*, the percentage of the value is 100%, this attribute will not have any influence on prediction so the weight value should be 0.

"**Property right**" has a higher weight value 86. The value can reflect the buyers' concerns on property. When other attributes are equal the complete property is first choice.

5. Second and Following Pages

Now the housing market in China is booming and many economic analysts point out the existing problems and forecast future bubbles. In this paper we research the price prediction of Harbin second-hand house market in China and analyze the influences of different attributes on price determination. All the time second-hand house market always reflects the actual market demand so the price trend of these houses is the barometer of real estate market. In this paper we use KNN-related algorithms to implement the price prediction and stimulated annealing methods to evaluate the importances of attributes. All of these algorithms and techniques will help to set the more reasonable prices and give valuable advice on key factors that will have greater influences on house prices to analysts decision makers of housing market.

References

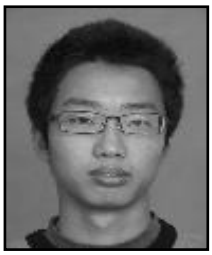
[1] Y. Hou, "Housing price bubbles in Beijing and Shanghai?: A multi-indicator analysis", International Journal of Housing Markets and Analysis, vol. 3, no. 1, **(2010)**, pp. 17-37.
[2] L. Berg, "Prices on the second-hand market for Swedish family houses: correlation, causation and determinants", European Journal of Housing Policy, vol. 2, no. 1, **(2002)**, pp. 1-24.
[3] D. Gray, "House price diffusion: an application of spectral analysis to the prices of Irish second-hand dwellings", Housing Studies, vol. 28, no. 6, **(2013)**, pp. 869-890.
[4] T. C. Leung and K. P. Tsang, "Anchoring and loss aversion in the housing market: implications on price dynamics", China Economic Review, **(2013)**, 24: 42-54.
[5] K. E. Case, R. J. Shiller and A. Thompson, "What have they been thinking? Home buyer behavior in hot and cold markets", National Bureau of Economic Research, **(2012)**.
[6] K. De Bruyne and J. Van Hove, "Explaining the spatial variation in housing prices: an economic geography approach", Applied Economics, vol. 45, no. 13, **(2013)**, pp. 1673-1689.
[7] G. Amato and F. Falchi, "On knn classification and local feature based similarity functions", Agents and Artificial Intelligence. Springer Berlin Heidelberg, **(2013)**, pp. 224-239.

[8] R. J. Samworth, "Optimal weighted nearest neighbour classifiers", The Annals of Statistics, vol. 40, no. 5, **(2012)**, pp. 2733-2763.
[9] T. Segaran, "Programming Collective Intelligence: Building Smart Web 2.0 Applications. O'Reilly Media", **(2007)**.
[10] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection", Statistics surveys, vol. 4, **(2010)**, pp. 40-79.
[11] K. A. Dowsland and J. M. Thompson, "Simulated annealing", Handbook of Natural Computing, Springer Berlin Heidelberg, **(2012)**, pp. 1623-1655.
[12] L. Ingber, A. Petraglia and M. R. Petraglia, "Adaptive simulated annealing", Stochastic global optimization and its applications with fuzzy adaptive simulated annealing", Springer Berlin Heidelberg, **(2012)**, pp. 33-62.
[13] F. Pedregosa, G. Varoquaux and A. Gramfort, "Scikit-learn: Machine learning in Python", The Journal of Machine Learning Research, vol. 12, **(2011)**, pp. 2825-2830.

Authors

Weikun Zhao, Student in School of Computer Science and Technology, Harbin Institute of Technology. His research fields are intelligence information processing.

Chao Sun, Doctor, work in automatic test and control institute, Harbin Institute of Technology, China; Lecturer. The main research covers image compression coding and virtual test technology.



Ji Wang, Student in School of Computer Science and Technology, Harbin Institute of Technology. His research fields are intelligence information processing.

