This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2710804, IEEE Transactions on Multimedia

Image Based Appraisal of Real Estate Properties

Quanzeng You, Ran Pang, Liangliang Cao, and Jiebo Luo, *Fellow, IEEE*

***Abstract*—Real estate appraisal, which is the process of estimating the price for real estate properties, is crucial for both buys and sellers as the basis for negotiation and transaction. Traditionally, the repeat sales model has been widely adopted to estimate real estate price. However, it depends the design and calculation of a complex economic related index, which is challenging to estimate accurately. Today, real estate brokers provide easy access to detailed online information on real estate properties to their clients. We are interested in estimating the real estate price from these large amounts of easily accessed data. In particular, we analyze the prediction power of online house pictures, which is one of the key factors for online users to make a potential visiting decision. The development of robust computer vision algorithms makes the analysis of visual content possible. In this work, we employ a Recurrent Neural Network (RNN) to predict real estate price using the state-of-the-art visual features. The experimental results indicate that our model outperforms several of other state-of-the-art baseline algorithms in terms of both mean absolute error (MAE) and mean absolute percentage error (MAPE).**

***Index Terms*—visual content analysis,real estate,deep neural networks**

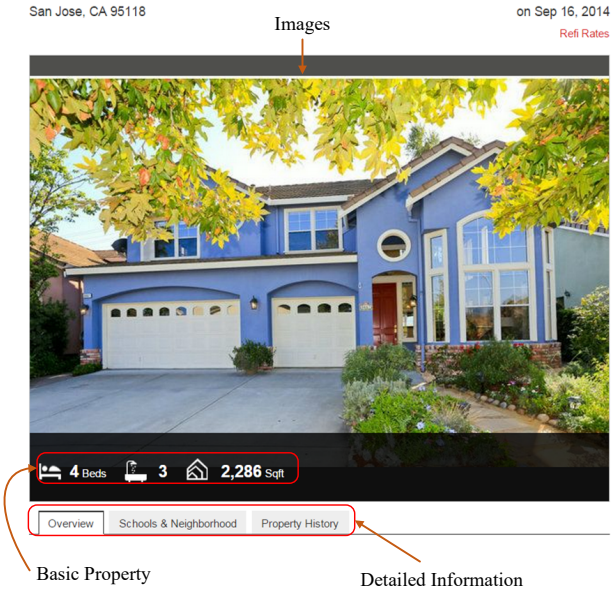


Fig. 1. Example of homes for sale from Realtor.

I. Introduction

Real estate appraisal, which is the process of estimating the price for real estate properties, is crucial for both buys and sellers as the basis for negotiation and transaction. Real estate plays a vital role in all aspects of our contemporary society. In a report published by the European Public Real Estate Association (EPRA http://alturl.com/7snxx), it was shown that *real estate in all its forms accounts for nearly 20% of the economic activity*. Therefore, accurate prediction of real estate prices or the trends of real estate prices help governments and companies make informed decisions. On the other hand, for most of the working class, housing has been one of the largest expenses. A right decision on a house, which heavily depends on their judgement on the value of the property, can possibly help them save money or even make profits from their investment in their homes. From this perspective, real estate appraisal is also closely related to people's lives.

Current research from both estate industry and academia has reached the conclusion that real estate value is closely related to property infrastructure [1], traffic [2], online user reviews [3] and so on. Generally speaking, there are several different types of appraisal values. In particular, we are interested in the *market value*, which refers to the trade price *in a competitive Walrasian auction setting* [4]. Today, people are likely to trade through real estate brokers, who provide easy access online websites for browsing real estate property in an interactive and convenient way. Fig. 1 shows an example of house listing from Realtor (http://www.realtor.com/), which is the largest real estate broker in North America. From the figure, we see that a typical piece of listing on a real estate property will introduce the infrastructure data in text for the house along with some pictures of the house. Typically, a buyer will look at those pictures to obtain a general idea of the overall property in a selected area before making his next move.

Traditionally, both real estate industry professionals and researchers have relied on a number of factors, such as economic index, house age, history trade and neighborhood environment [5] and so on to estimate the price. Indeed, these factors have been proved to be related to the house price, which is quite difficult to estimate and sensitive to many different human activities. Therefore, researchers have devoted much effort in building a robust house price index [6], [7], [8], [9]. In addition, quantitative features including *Area, Year, Storeys, Rooms* and *Centre* [10], [11] are also employed to build neural network models for estimating house prices. However, pictures, which is probably the most important factor on a buyer's initial decision making process [12], have been ignored in this process. This is partially due to the fact that visual content is very difficult to interpret or quantify by computers

Manuscript received March 28, 2016; accepted February 10, 2017.

Q. You and J. Luo are with the Department of Computer Science, University of Rochester, Rochester, NY 14623 USA. E-mail: {qyou, jluo}@cs.rochester.edu.

R. Pang is with PayPaI. E-mail: pangrr89@gmail.com

L. Cao is with Electrical Engineering & Computer Sciences, Columbia University and customerserviceAI. E-mail: liangliang.cao@gmail.com

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2710804, IEEE Transactions on Multimedia

compared with human beings.

A picture is worth a thousand words. One advantage with images and videos is that they act like universal languages. People with different backgrounds can easily understand the main content of an image or video. In the real estate industry, pictures can easily tell people exactly how the house looks like, which is impossible to be described in many ways using language. For the given house pictures, people can easily have an overall feeling of the house, *e.g.* what is the overall construction style, how the neighboring environment looks like. These high-level attributes are difficult to be quantitatively described. On the other hand, today's computational infrastructure is also much cheaper and more powerful to make the analysis of computationally intensive visual content analysis feasible. Indeed, there are existing works on focusing the analysis of visual content for tasks such as prediction [13], [14], and online user profiling [15]. Due to the recently developed deep learning, computers have become smart enough to interpret visual content in a way similar to human beings.

Recently, deep learning has enabled robust and accurate feature learning, which in turn produces the state-of-the-art performance on many computer vision related tasks, *e.g.* digit recognition [16], [17], image classification [18], [19], aesthetics estimation [20] and scene recognition [21]. These systems suggest that deep learning is very effective in learning robust features in a supervised or unsupervised fashion. Even though deep neural networks may be trapped in local optima [22], [23], using different optimization techniques, one can achieve the state-of-the-art performance on many challenging tasks mentioned above.

Inspired by the recent successes of deep learning, in this work we are interested in solving the challenging real estate appraisal problem using deep visual features. In particular, for images related tasks, Convolutional Neural Network (CNN) are widely used due to the usage of convolutional layers. It takes into consideration the locations and neighbors of image pixels, which are important to capture useful features for visual tasks. Convolutional Neural Networks [24], [18], [19] have been proved very powerful in solving computer vision related tasks.

We intend to employ the pictures for the task of real estate price estimation. We want to know whether visual features, which is a reflection of a real estate property, can help estimate the real estate price. Intuitively, if visual features can characterize a property in a way similar to human beings, we should be able to quantify the house features using those visual responses. Meanwhile, real estate properties are closely related to the neighborhood. In this work, we develop algorithms which only rely on 1) the neighbor information and 2) the attributes from pictures to estimate real estate property price.

To preserve the local relation among properties we employ a novel approach, which employs random walks to generate house sequences. In building the random walk graph, only the locations of houses are utilized. In this way, the problem of real estate appraisal has been transformed into a sequence learning problem. Recurrent Neural Network (RNN) is particularly designed to solve sequence related problems. Recently, RNNs have been successfully applied to challenging tasks including machine translation [25], image captioning [26], and speech recognition [27]. Inspired by the success of RNN, we deploy RNN to learn regression models on the transformed problem.

The main contributions of our work are as follows:

- To the best of our knowledge, we are the first to quantify the impact of visual content on real estate price estimation. We attribute the possibility of our work to the newly designed computer vision algorithms, in particular Convolutional Neural Networks (CNNs).
- We employ random walks to generate house sequences according to the locations of each house. In this way, we are able to transform the problem into a novel sequence prediction problem, which is able to preserve the relation among houses.
- We employ the novel Recurrent Neural Networks (RNNs) to predict real estate properties and achieve accurate results.

II. Related Work

Real estate appraisal has been studied by both real estate industrial professionals and academia researchers. Earlier work focused on building price indexes for real properties. The seminal work in [6] built price index according to the repeat prices of the same property at different times. They employed regression analysis to build the price index, which shows good performances. Another widely used regression model, Hedonic regression, is developed on the assumption that the characteristics of a house can predict its price [7], [8]. However, it is argued that the Hedonic regression model requires more assumptions in terms of explaining its target [28]. They also mentioned that for repeat sales model, the main problem is lack of data, which may lead to failure of the model. Recent work in [9] employed locations and sale price series to build an autoregressive component. Their model is able to use both single sale homes and repeat sales homes, which can offer a more robust sale price index.

More studies are conducted on employing feed forward neural networks for real estate appraisal [29], [30], [31], [32]. However, their results suggest that neural network models are unstable even using the same package with different run times [29]. The performance of neural networks are closely related to the features and data size [32]. Recently, Kontrimas and Verikas [33] empirically studied several different models on selected 12 dimensional features, *e.g.* type of the house, size, and construction year. Their results show that linear regression outperforms neural network on their selected 100 houses.

More recent studies in [1] propose a ranking objective, which takes geographical individual, peer and zone dependencies into consideration. Their method is able to use various estate related data, which helps improve their ranking results based on properties' investment values. Furthermore, the work in [3] studied online user's reviews and mobile users' moving behaviors on the problem of real estate ranking. Their proposed sparsity regularized learning model demonstrated competitive performance.

In contrast, we are trying to solve this problem using the attributes reflected in the visual appearances of houses. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2710804, IEEE Transactions on Multimedia

particular, our model does not use the meta data of a house (*e.g.* size, number of rooms, and construction year). We intend to utilize the location information in a novel way such that our model is able to use the state-of-the-art deep learning for feature extraction (Convolutional Neural Network) and model learning (Recurrent Neural Network).

III. Recurrent Neural Network for Real Estate Price Estimation

In this section, we present the main components of our framework. We describe how to transform the problem into a problem that can be solved by the Recurrent Neural Network. The architecture of our model is also presented.

A. Random Walks

One main feature of real estate properties is its location. In particular, for houses in the same neighborhood, they tend to have similar *extrinsic* features including traffic, schools and so on. We build an undirected graph G for all the houses collected, where each node v_i represent the i-th house in our data set. The similarity s_{ij} between house h_i and house h_j is defined using the *Gaussian* kernel function, which is a widely used similarity measure[1]:

$$s_{ij} = \exp\left(\frac{dist(h_i, h_j)}{2\sigma^2}\right), \tag{1}$$

where $dist(h_i, h_j)$ is the geodesic distance between house h_i and h_j. σ is the hyper-parameter, which controls the similarity decaying velocity with the increase of distance. In all of our experiments, we set σ to 0.5 miles so that houses within the 1.5 (within 3σ) miles will have a relatively larger similarity. The ϵ-neighborhood graph [34] is employed to build G in our implementation. We assign the weight of each edge e_{ij} as the similarity s_{ij} between house h_i and the house h_j.

Given this graph G, we can then employ random walks to generate sequences. In particular, every time, we randomly choose one node v_i as the root node, then we proportionally jump to its neighboring nodes v_j according to the weights between v_i and its neighbors. The probability of jumping to node v_j is defined as

$$p_j = \frac{e_{ji}}{\sum_{k \in N(i)} e_{ki}}, \tag{2}$$

where $N(i)$ is the set of neighbor nodes of v_i. We continue to employ this process until we generate the desired length of sequence. The employment of random walks is mainly motivated by the recent proposed DeepWalk [35] to learn feature representations for graph nodes. It has been shown that random walks can capture the local structure of the graphs. In this way, we can keep the local location structure of houses and build sequences for houses in the graph. Algorithm 1 summarizes the detailed steps for generating sequences from a similarity graph.

We have generated sequences by employing random walks. In each sequence, we have a number of houses, which is related in terms of their locations. Since we build the graph on top of house locations, the houses within the same sequence are highly possible to be close to each other. In other words, the prices of houses in the same sequence are related to each other. We can employ this context for estimating real estate property price, which can be solved by recurrent neural network discussed in following sections.

[1]http://en.wikipedia.org/wiki/Radial_basis_function_kernel

B. Recurrent Neural Network

With a Recurrent Neural Network (RNN), we are trying to predict the output sequence $\{y_1, y_2, \ldots, y_T\}$ given the input sequence $\{x_1, x_2, \ldots, x_T\}$. Between the input layer and the output layer, there is a hidden layer, which is usually estimated as in Eq.(3).

$$h_t = \Delta(W_h^i h_{t-1} + W_x x_t + b_h) \tag{3}$$

Δ represents some selected activation function or other complex architecture employed to process the input x_t and h_t. One of the most widely deployed architectures is Long Short-Term Memory (LSTM) cell [36], which can overcome the *vanishing* and *exploding* gradient problem [37] when training RNN with gradient descent. Fig. 2 shows the details of a single Long Short-Term Memory (LSTM) block [38]. Each LSTM cell contains an input gate, an output gate and an forget gate, which is also called a *memory cell* in that it is able to *remember* the error in the error propagation stage [39]. In this way, LSTM is capable of modeling long-range dependencies than conventional RNNs.

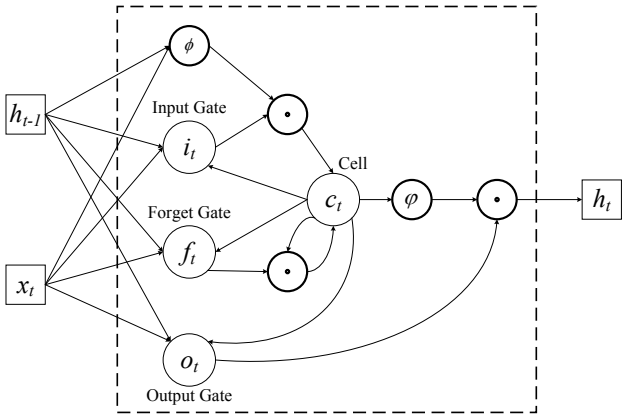


Fig. 2. An illustration of a single Long Short-Term Memory (LSTM) Cell.

For completeness, we give the detailed calculation of h_t given input x_t and h_{t-1} in the following equations. Let $W_\cdot^i$, $W_\cdot^f$, $W_\cdot^o$ represent the parameters related to input, forget and output gate respectively. $\odot$ denotes the element-wise multiplication between two vectors. ϕ and ψ are some selected activation functions and σ is the fixed logistic sigmoid function. Following [38], [27], [40], we employ $\tanh$ for both

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2710804, IEEE Transactions on Multimedia

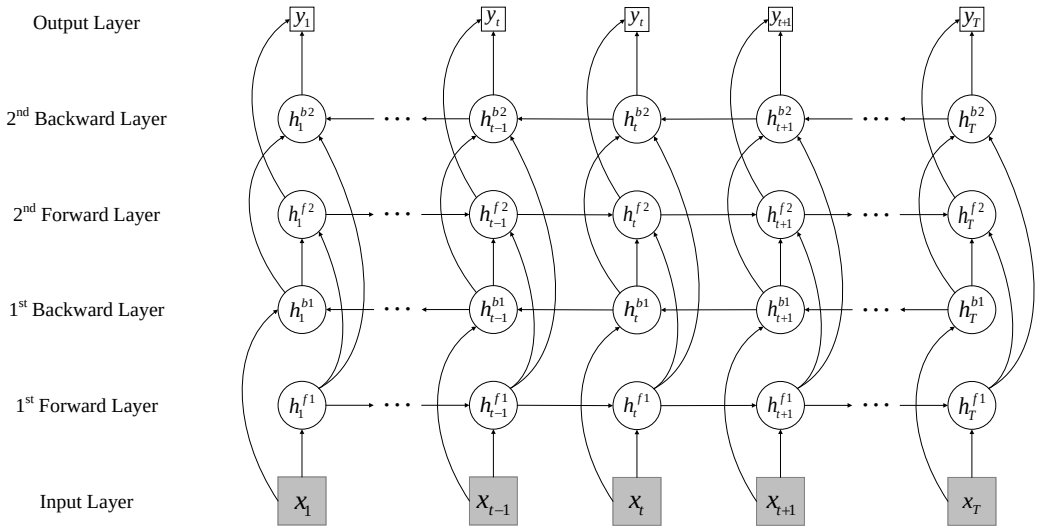


Fig. 3. The Multi-layer Bidirectional Recurrent Neural Network (BRNN) architecture for real estate price estimation. There are two bidirectional recurrent layers in this architecture. For real estate price estimation, the price of each house is related to all houses in the same *sequence*, which is the main motivation to employ bidirectional recurrent layers.

ϕ in Eq.(6) and ψ in Eq.(8).

$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1} + W_c^i c_{t-1} + b_i) \quad (4)$$
$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1} + W_c^f c_{t-1} + b_f) \quad (5)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_x^c x_t + W_h^c h_{t-1} + b_c) \quad (6)$$
$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1} + W_c^o c_t + b_o) \quad (7)$$
$$h_t = o_t \odot \psi(c_t) \quad (8)$$

C. Multi-layer Bidirectional LSTM

In previous sections, we have discussed the generation of sequences as well as Recurrent Neural Network. Recall that we have built an undirected graph in generating the sequences, which indicates that the price of one house is related to all the houses in the same sequence including those in the later part. Bidirectional Recurrent Neural Network (BRNN) [41] has been proposed to enable the usage of both earlier and future contexts. In bidirectional recurrent neural network, there is an additional backward hidden layer iterating from the last of the sequence to the first. The output layer is calculated by employing both forward and backward hidden layer.

Bidirectional-LSTM (B-LSTM) is a particular type of BRNN, where each hidden node is calculated by the long short-term memory as shown in Fig. 2. Graves *et al.* [40] have employed Bidirectional-LSTM for speech recognition. Fig. 3 shows the architecture of the bidirectional recurrent neural network. We have two Bidirectional-LSTM layers. During the forward pass of the network, we calculate the response of both the forward and the backward hidden layers in the 1st-LSTM and 2nd-LSTM layer respectively. Next, the output (in our problem, the output is the price of each house) of each house is calculated using the output of the 2nd-LSTM layer as input to the output layer.

Algorithm 1 RandomWalks

Input: $H = \{h_1, h_2, \ldots, h_n\}$ geo-coordinates of n houses
σ hyper-parameter for Gaussian Kernel
t threshold for distance
M total number of desired sequences

Calculate the Vincenty distance between any pair of houses
Calculate the similarity between houses according to the Gaussian kernel function (see Eq.(1)).
repeat
  Initialize $s_c = \{\}$
  Randomly pick one node h_i and add h_i to s_c
  set $h_c = h_i$
  while size$(s_c) < L$ **do**
    Pick h_c's neighbor node h_j with probability p_j defined in Eq.(2)
    add h_j to s_c
    set $h_c = h_j$
  end whileadd s_c to S
until size $(S) = M$
return The set of sequence S

The objective function for training the Multi-Layer Bidirectional LSTM is defined as follows:

$$L = \frac{1}{N} \sum_{n=1}^{N} \sum_{j} \parallel \hat{y}_{ij} - y_{ij} \parallel^2 \quad (9)$$

where W is the the set of all the weights between different layers.y_{ij} is the actual trade price for the j-th house in the generated i-th sequence and $\hat{y}_{ij}$ is the corresponding estimated price for this house.

When training our Multi-Layer B-LSTM model, we employ the RMSProp [42] optimizer, which is an adaptive method for automatically adjust the learning rates. In particular, it normalizes the gradients by the average of its recent magnitude.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2710804, IEEE Transactions on Multimedia

Algorithm 2 Training Multi-Layer B-LSTM

Input: $H = \{h_1, h_2, \ldots, h_n\}$ geo-coordinates of n houses
$X = \{x_1, x_2, \ldots, x_n\}$ features of the n house
$Y = \{y_1, y_2, \ldots, y_n\}$ prices of the n houses

1: S = RandomWalks (see Algorithm 1)
2: Split S into mini-batches
3: **repeat**
4: Calculate the gradient of L in Eq.(9) and update the parameters using RMSProp.
5: **until** Convergence
6: **return** The learned model M

We conduct the back propagation in a mini-batch approach. Algorithm 2 summarizes the main steps for our proposed algorithm.

D. Prediction

In the prediction stage, the first step is also generating sequence. For each testing house, we add it as a new node into our previously build similarity graph on the training data. Each testing house is a new node in the graph. Next, we add edges to the testing nodes and the training nodes. We use the same settings when adding edges to the new ϵ-neighborhood graph. Given the new graph G', we randomly generate sequences and keep those sequences that contain one and only one testing node. In this way, for each house, we are able to generate many different sequences that contain this house. Fig. 4 shows the idea. Each testing sequence only has one testing house. The remaining nodes in the sequence are the known training houses.

a) Average: The above strategy implies that we are able to build many different sequences for each testing house. To obtain the final prediction price for each testing house, one simple strategy is to average the prediction results from different sequences and report the average price as the final prediction price.

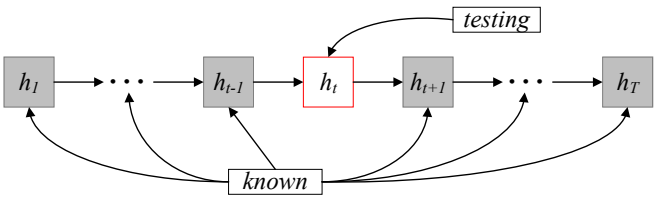


Fig. 4. Testing sequence $h_1 \rightarrow h_2 \rightarrow \cdots \rightarrow h_T$. In each testing sequence, there is one and only one testing node in that sequence. The remaining nodes are all come from training data.

IV. Experimental Results

In this section, we discuss how to collect data and evaluate the proposed framework as well as several state-of-the-art approaches. In this work, all the data are collected from Realtor (http://www.realtor.com/), which is the largest realtor association in North America. We collect data from San Jose, CA, one of the most active cities in U.S., and Rochester, NY, one of the least active cities in U.S., over a period of one year. In the next section, we will discuss the details on how to preprocess the data for further experiments.

A. Data Preparation

The data collected from Realtor contains description, school information and possible pictures about each real property as shown in Fig. 1 show. We are particularly interested in employing the pictures of each house to conduct the price estimation. We filter out those houses without image in our data set. Since houses located in the same neighborhood seem to have similar price, the location is another important features in our data set. However, after an inspection of the data, we notice that some of the house price are abnormal. Thus, we preprocess the data by filtering out houses with extremely high or low price compared with their neighborhood.

TABLE I shows the overall statistics of our dataset after filtering. Overall, the city of San Jose has more houses than Rochester on the market (as expected for one of the hottest market in the country). The house prices in the two cities also have significant differences. Fig. 5 shows some of the example house pictures from the two cities, respectively. From these pictures, we observe that houses whose prices are above average typically have larger yards and better curb appeal, and vice versa. The same can be observed among house interior pictures (examples not shown due to space).

TABLE I
THE AVERAGE PRICE PER SQFT AND THE STANDARD DEVIATION (STD) OF THE PRICE OF THE TWO STUDIED CITIES.

City	# of Houses	Avg Price	std of Price
San Jose	3064	454.2	132.1
Rochester	1500	76.4	21.2

Realtor does not provide the exact geo-location for each house. However, geo-location is important for us to build the ϵ-neighborhood graph for random walks. We employ Microsoft Bing Map API (https://msdn.microsoft.com/en-us/library/ff701715.aspx) to obtain the latitude and longitude for each house given its collected address. Fig. 6 shows some of the houses in our collected data from San Jose and Rochester using the returned geo-locations from Bing Map API.

According to these coordinates, we are able to calculate the distance between any pair of houses. In particular, we employ Vincenty distance (https://en.wikipedia.org/wiki/Vincenty's_formulae) to calculate the geodesic distances according to the coordinates. Fig. 7 shows distribution of the distance between any pair of houses in our data set. The distance is less than 4 miles for most randomly picked pair of houses. In building our ϵ-neighborhood graph, we assign an edge between any pair of houses, which has a distance smaller than 5 miles ($\epsilon = 5$ miles).

B. Feature Extraction and Baseline Algorithms

In our implementation, we experimented with GoogleNet model [43], which is one of the state-of-the-art deep neural architectures. In particular, we use the response from the last $avg - pooling$ layer as the visual features for each image. In

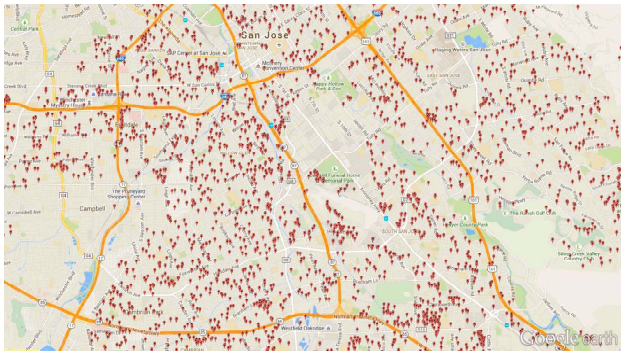
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2710804, IEEE Transactions on Multimedia

(a) Rochester (b) San Jose

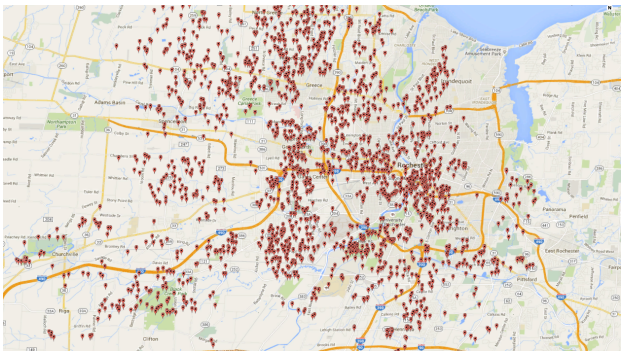
Fig. 5. Examples of house pictures of the two cities respectively. Top Row: houses whose prices (per Sqft) are above the average of their neighborhood. Bottom Row: houses whose prices (per Sqft) are below the average of their neighborhood.

TABLE II
PREDICTION DEVIATION OF DIFFERENT MODELS FROM THE ACTUAL SALE PRICES. NOTE THAT RNN-BEST IS THE UPPER-BOUND PERFORMANCE OF THE RNN BASED MODEL PROPOSED IN THIS WORK.

City	LASSO		DeepWalk		RNN-best		RNN-avg	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
San Jose	70.79	16.92%	68.05	16.12%	**17.98**	**4.58%**	**66.3**	**16.11%**
Rochester	14.19	24.83%	13.68	23.28%	**5.21**	**9.94%**	**13.32**	**22.69%**



(a) San Jose, CA



(b) Rochester, NY

Fig. 6. Distribution of the houses in our collected data for both San Jose and Rochester according to their geo-locations.

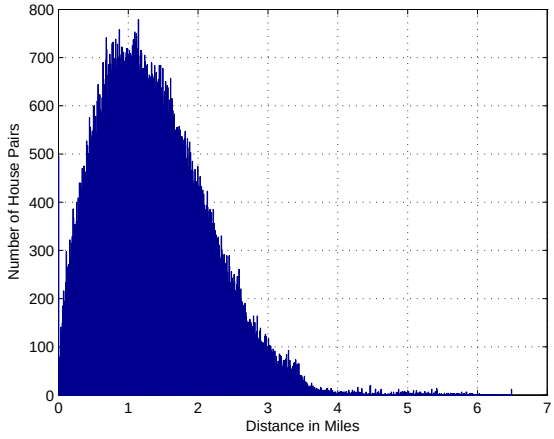


Fig. 7. Distribution of distances between different pairs of houses.

this way, we obtain a $1,024$ dimensional feature vector for each image. Each house may have several different pictures on different angles of the same property. We average features of all the images of the same house (also known as *average-pooling*)[2] to obtain the feature representation of the house.

[2]We also tried max-pooling. However, the results are not as good as average-pooling. In the following experiments, we report the results using average-pooling.

We compare the proposed framework with the following algorithms.

1) Regression Model (LASSO): Regression model has been employed to analyze real estate price index [6]. Recently, the results in Fu *et al.* [3] show that sparse regularization can obtain better performance in real estate ranking. Thus, we choose to use LASSO (http://statweb.stanford.edu/~tibs/lasso.html), which is a $l1$-constrained regression model, as one of our baseline algorithms.

2) DeepWalk: Deepwalk [35] is another way of employing random walks for unsupervised feature learning of graphs. The main approach is inspired by distributed word representation learning. In using DeepWalk, we also use ϵ-neighborhood graph with the same settings with the graph we built for generating sequences for B-LSTM. The learned features are also fed into a LASSO model for learning the regression weights. Indeed, deepwalk can be thought as a simpler version of our algorithm, where only the graph structure are employed to learn features. Our framework can employ both the graph

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2710804, IEEE Transactions on Multimedia

structure and other features, *i.e.* visual attributes, for building regression model.

C. Training a Multi-layer B-LSTM Model

With the above mentioned similarity graph, we are able to generate sequences using random walks following the steps described in Algorithm 1. For each city, we randomly split the houses into training (80%) and testing set (20%). Next, we generate sequences using random walks on the training houses only to build our training sequences for Multi-layer B-LSTM.

For both cities, we build $200,000$ sequences for training, with a length of 10. Similarly, we also generate testing sequences, where each sequence contain one and only one testing house (see Fig. 4). On the average, we randomly generate 100 sequences for each testing house. The B-LSTM model is trained with a batch size of 1024. In our experimental settings, we set the size of the first hidden layer to be 400 and the size of the second hidden layer to be 200.

The evaluation metrics employed are mean absolute error (MAE) and mean absolute percentage error (MAPE). Both of them are popular measures for evaluating the accuracy of prediction models. Eq.(10) and Eq.(11) give the definitions for these two metrics, where p_i is the predicted value and t_i is the true value for the i-th instance.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|t_i - p_i| \tag{10}$$

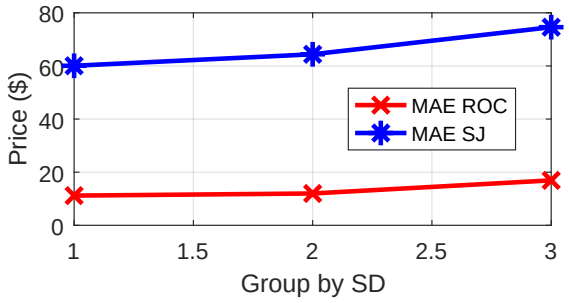
$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{t_i - p_i}{t_i}\right| \tag{11}$$

We use the same training and testing split to evaluate all the approaches. TABLE II shows the regression results for all the different approaches in the two selected cities. For each testing house, we generate about 100 sequences. In TABLE II, we report both the best and the average price of the predicted price. For Rochester, the average standard deviation of the predicted prices over all the houses is 5.6, which is 7.33% of the average price in Rochester (see TABLE I). Comparably, the average standard deviation for San Jose is 34.64, which is 7.63% of the average price in San Jose. The *best* is the price closest to the true price among all the available sequences for each house[3]. Overall, our B-LSTM model outperforms other two baseline algorithms in both cities. All of the evaluation approaches perform better in San Jose than in Rochester in terms of MAPE. This is possible due to the availability of more training data in the city of San Jose. DeepWalk shows slightly better performance than LASSO, which suggests that location is relatively more important than the visual features in the realtor business. This is expected

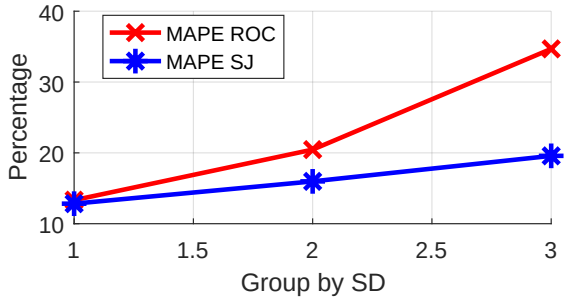
D. Confidence Level

For each testing house, the proposed model can give a group of predictions. We want to know whether or not the proposed model can distinguish the confidence level of its prediction. In particular, we group the testing houses evenly into three groups for each city. The first group has the smallest standard deviation of the prediction prices. The second group is the middle one and the last group is the one with the largest standard deviation.

[3]This is the upper bound of the prediction results. We choose the closest price using the ground truth price as reference.



(a) MAE



(b) MAPE

Fig. 8. Performance of B-LSTM-avg in different groups. All the testing houses are grouped by the predicted standard deviation.

Fig. 8 shows the MAE and MAPE for the different groups. The results show that standard deviation can be viewed as a rough measure of the confidence level of the proposed model on the current testing house. Small standard deviation tends to indicate a high confidence of the model and overall it also suggests a smaller prediction error.

V. Conclusion

In this work, we propose a novel framework for real estate appraisal. In particular, the proposed framework is able to take both the location and the visual attributes into consideration. The evaluation of the proposed model on two selected cities suggests the effectiveness and flexibility of the model. Indeed, our work has also offered new approaches of applying deep neural networks on graph structured data. We hope our model can not only give insights on real estate appraisal, but also can inspire others on employing deep neural networks on graph structured data.

References

[1] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, and Z.-H. Zhou, "Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering," in *SIGKDD*. ACM, 2014, pp. 1047–1056.

[2] K. Wardrip, "Public transits impact on housing costs: a review of the literature," 2011.

[3] Y. Fu, Y. Ge, Y. Zheng, Z. Yao, Y. Liu, H. Xiong, and N. Yuan, "Sparse real estate ranking with online user reviews and offline moving behaviors," p. 120129, 2014.

[4] A. Beja and M. B. Goldman, "On the dynamic behavior of prices in disequilibrium," *The Journal of Finance*, vol. 35, no. 2, pp. 235–248, 1980.

[5] E. L'Eplattenier, "How to run a comparative market analysis (cma) the right way," http://fitsmallbusiness.com/comparative-market-analysis/, 2016.

[6] M. J. Bailey, R. F. Muth, and H. O. Nourse, "A regression method for real estate price index construction," *Journal of the American Statistical Association*, vol. 58, no. 304, pp. 933–942, 1963.

[7] R. Meese and N. Wallace, "Nonparametric estimation of dynamic hedonic price models and the construction of residential housing price indices," *Real Estate Economics*, vol. 19, no. 3, pp. 308–332, 1991.

[8] S. Sheppard, "Hedonic analysis of housing markets," *Handbook of regional and urban economics*, vol. 3, pp. 1595–1635, 1999.

[9] C. H. Nagaraja, L. D. Brown, L. H. Zhao *et al.*, "An autoregressive approach to house price modeling," *The Annals of Applied Statistics*, vol. 5, no. 1, pp. 124–149, 2011.

[10] T. Lasota, Z. Telec, G. Trawiński, and B. Trawiński, "Empirical comparison of resampling methods using genetic fuzzy systems for a regression problem," in *Intelligent Data Engineering and Automated Learning-IDEAL 2011*. Springer, 2011, pp. 17–24.

[11] O. Kempa, T. Lasota, Z. Telec, and B. Trawiński, "Investigation of bagging ensembles of genetic neural networks and fuzzy systems for real estate appraisal," in *Intelligent Information and Database Systems*. Springer, 2011, pp. 323–332.

[12] W. Di, N. Sundaresan, R. Piramuthu, and A. Bhardwaj, "Is a picture really worth a thousand words?:-on the role of images in e-commerce," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 633–642.

[13] X. Jin, A. Gallagher, L. Cao, J. Luo, and J. Han, "The wisdom of social multimedia: using flickr for prediction and forecast," in *Proceedings of the international conference on Multimedia*. ACM, 2010, pp. 1235–1244.

[14] Q. You, L. Cao, Y. Cong, X. Zhang, and J. Luo, "A multifaceted approach to social multimedia-based prediction of elections," *Multimedia, IEEE Transactions on*, vol. 17, no. 12, pp. 2271–2280, Dec 2015.

[15] Q. You, S. Bhatia, and J. Luo, "A picture tells a thousand words?about you! user interest profiling from user generated visual content," *Signal Processing*, vol. 124, pp. 45–53, 2016.

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[17] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[18] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *IJCAI*. AAAI Press, 2011, pp. 1237–1242.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." in *NIPS*, vol. 1, no. 2, 2012, p. 4.

[20] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, "Rapid: Rating pictorial aesthetics using deep learning," in *ACM MM*. ACM, 2014, pp. 457–466.

[21] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014, pp. 487–495.

[22] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[23] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 437–478.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2014.

[26] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015, pp. 3156–3164.

[27] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*. IEEE, 2013, pp. 6645–6649.

[28] F. T. Wang and P. M. Zorn, "Estimating house price growth with repeat sales data: what's the aim of the game?" *Journal of Housing Economics*, vol. 6, no. 2, pp. 93–118, 1997.

[29] E. Worzala, M. Lenk, and A. Silva, "An exploration of neural networks and its application to real estate valuation," *Journal of Real Estate Research*, vol. 10, no. 2, pp. 185–201, 1995.

[30] P. Rossini, "Improving the results of artificial neural network models for residential valuation," in *Fourth Annual Pacific-Rim Real Estate Society Conference, Perth, Western Australia*, 1998.

[31] P. Kershaw and P. Rossini, "Using neural networks to estimate constant quality house price indices," Ph.D. dissertation, INTERNATIONAL REAL ESTATE SOCIETY, 1999.

[32] N. Nghiep and C. Al, "Predicting housing value: A comparison of multiple regression analysis and artificial neural networks," *Journal of Real Estate Research*, vol. 22, no. 3, pp. 313–336, 2001.

[33] V. Kontrimas and A. Verikas, "The mass appraisal of the real estate by computational intelligence," *Applied Soft Computing*, vol. 11, no. 1, pp. 443–448, 2011.

[34] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[35] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *SIGKDD*. ACM, 2014, pp. 701–710.

[36] F. Gers, "Long short-term memory in recurrent neural networks," *Unpublished PhD dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*, 2001.

[37] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *ICML*, 2013, pp. 1310–1318.

[38] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *The Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2003.

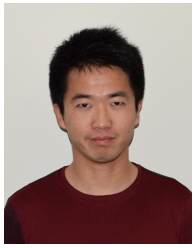
[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 273–278.

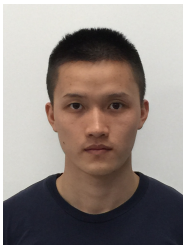
[41] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *Signal Processing, IEEE Transactions on*, vol. 45, no. 11, pp. 2673–2681, 1997.

[42] T. Tieleman and G. Hinton, "Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning," University of Toronto, Tech. Rep., 2012.

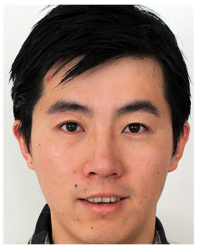
[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, June 2015.



Quanzeng You received both B.E. and M.E. from Dalian University of Technology. He is currently a 4th year Ph.D. student with Department of Computer Science, University of Rochester. His advisor is Prof. Jiebo Luo. His research focuses on social multimedia, social networks and data mining. He is interested in developing effective machine learning algorithms that can help us understand the data. His recent research is high level visual understanding, including image captioning and visual sentiment analysis.



Ran Pang Ran Pang is currently enrolled in a master program at the Department of Computer Science in the University of Rochester. He is interested in artificial intelligence and his research focus on social multimedia and data mining.



Liangliang Cao is currently a senior research scientist at Yahoo! Labs and an adjunct faculty at Columbia University. His research lies in the intersection of computer vision, multimedia, and big data analytics. Dr. Cao has authored over 40 papers in top conferences and journals, including the International Conference on Computer Vision, the Computer Vision and Pattern Recognition Conference, the European Conference on Computer Vision, the Conference on Neural Information Processing Systems, the ACM Multimedia, the International World Wide Web Conference, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and the PROCEEDINGS OF THE IEEE. He was the general Chair of the Greater New York Area Multimedia and Vision Meeting in 2012 and 2013. He was an area chair of WACV 2014 and ACM Multimedia 2012. He was a guest editor of the ACM Transactions on Multimedia Computing, Communications, and Applications, and the Computer Vision and Image Understanding journal.

Jiebo Luo (S'93M'96SM'99F.'09) joined the University of Rochester in Fall 2011, after over 15 years at Kodak Research Laboratories (Rochester, NY), where he was a Senior Principal Scientist leading research and advanced development. He is a fellow of the International Society for Optics and Photonics, and the International Association for Pattern Recognition. He has been involved in numerous technical conferences, and served as the Program Co-Chair of ACM Multimedia 2010 and the IEEE CVPR 2012. He is the Editor-in-Chief of the Journal of Multimedia, and has served on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Pattern Recognition, Machine Vision and Applications, and the Journal of Electronic Imaging.