Using Graph Methods to Capture Spatial Relationships in Home Prices

Atish Sawant
New York, USA
atishsawant87@gmail.com

***Abstract*—This study seeks to study whether graphical models such as GCN, GraphSage, and GAT outperform traditional regression based methods for predicting real estate prices. Single family home prices are recorded upon transfer with a county assessor's office. Using this data, this report seeks to predict the sales price of homes that have not yet sold. If successful, the report would open up a new way for consumers to value their home without the need for expensive and infrequent appraisals for one of the most significant investments of their life.**

I. Introduction

For most individuals the single largest investment they will ever make is the purchase of their home, and yet there is little clarity on whether they are overpaying, or getting a bargain. This has led to the proliferation and entrenchment of real estate agents who take a 6% commission on the sale of a home. These agents are incentivized to sell quickly rather than maximize value. Further, homes can sit on the market for extended periods of time if they are mispriced, hampering a family's ability to move.

Several startups and established companies are attempting to attack the problem of valuing one's home. Zillow has one of the most well-known estimates on the market, but even that has significant errors. Their published median average error is 7.2% for Atlanta, and they are within 5% of the sales price only 38.3% of the time. For another city like Cleveland there is a median error of 10.0% and only 29.0% of the time are they within 5% of the sales price.[1]

Human appraisers significantly outperform existing models since they look at prior sales in the area and try to constrain their comparisons to homes that are similar. They then make adjustments to those sales prices based on the slight differences in characteristics between the comparison home and home up for appraisal. The human touch does introduce bias. When there is a contract already in place for the sale of the home, the appraisal comes back at a price higher than the sales price over 92% of the time. In these cases the appraiser is not seeking to evaluate the price of a home, but check of box necessary for closing. In one unbiased case, Fannie Mae repossessed homes after the financial crisis and had their portfolio appraised. The median average appraiser error was 1.8% and they were within 10% of the contract price 60.6% of the time.[2]

Broad level geographic features such as crime, school district, income, and location have long been used in qualitatively determining the "value" of an area. More recently, those same variables have been used in the creation of AVMs. However, while geographic and home features contribute heavily to the final sale price of a home, there is no clarity on the relationship between other homes in the area, and prior sales in the area. The application of graphical models has the ability to tease out spatial relationships between nearby properties that have otherwise been ignored or misused.

The goal of the study determine whether graphical models are better at identifying price relationships between homes than the currently used regressions and decision trees. The intuition behind this approach is that homes are laid out in a graphical form, forcing the spatial relationships into a regression framework is rigid and likely leaves out important relationships. Furthermore, there is the complication of time. Appraisers use their judgment in determining how long ago is too long for a home to be a valid comparison and how much they have to adjust for the passage of time. In this way, we can view the problem through the lens of a spatio-temporal graph. Ultimately, if a strong graphical model is found, it can help lay the foundations for models which allow homeowners to accurately value their home and potentially sell it without the need for appraisers or reliance on error prone automated valuation models.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

II. Related Work

Graph theory has been a very active field of research in recent years. Breakthroughs such as GraphSage and Graph Attention Networks have become state of the art for graph tasks. These techniques bring spatial features to the forefront, and are best suited for tasks where the layout or relations between elements is critical to their meaning. Adding the time variable to graph connected is one of the cutting edges of the field right know with the research going into spatio-temporal graphs. These techniques incorporate the idea that graph connectedness and node embeddings and features can evolve over time. However, in the field of real estate very little attention has so far been paid to applying graph theory to many modeling problems at hand.

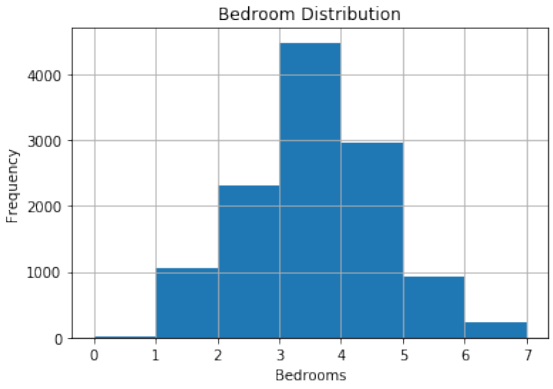
GraphSage is one of the techniques that we will look to for outperformance versus regressions. The key idea within GraphSage is that it looks towards its neighbors and aggregates their features to create an embedding for the target node. One of the hyperparameters allows for varying levels of search depths. The technique can be thought of as an expansion of the original GCN algorithm. Within the issue of pricing homes, aggregating the features of the homes nearby and using that information to predict the price is an intuitive practice and analogous to the original intention of the algorithm to classify graph nodes.

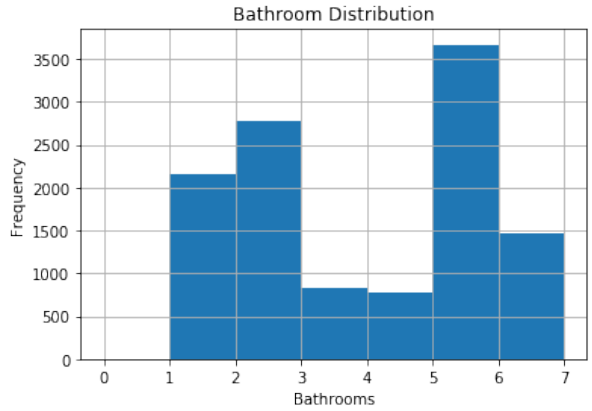
Although minimal graph theory research has been conducted in the field of real estate, there are still important papers and ideas from other fields that can be incorporated. Of particular interest is Yu, Yin, and Zhu's work in spatio-temporal GCNs (STGCN).[3] In their paper they used this technique to monitor and predict traffic flow based on time. Prior times series methods did not incorporate the spatial relationship between different traffic stations effectively. Similarly, one could show that home price spatial relationships are more effectively modeled through STGCN's since their relationship with neighbors changes over time.

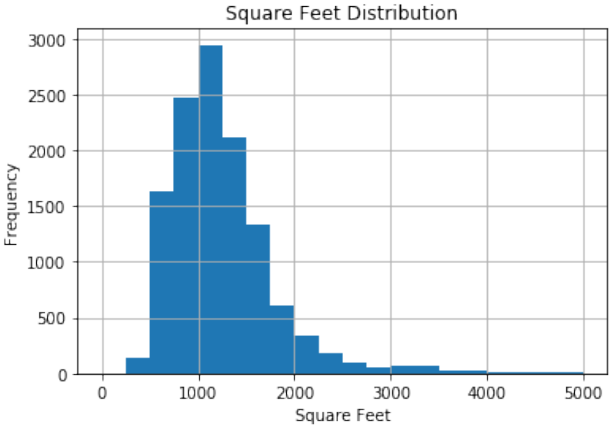
Currently much of the work that has been done for real estate automated valuation models(AVMs) has centered around variants of regression. Exact details are difficult to come by since these algorithms are proprietary and core to the business. However, one of difficulties of using regressions for this problem is properly incorporating spatial relationships into the model. The time component must also be forced into a feature. This is likely why human appraisers are still able to outperform models in such a data rich problem. They are able to make judgments from their experience on which homes are similar enough, and how to adjust prices based on how far back into the past they go. Incorporating this type of information into a regression model is difficult while graph-based models are more suited to model these issues.

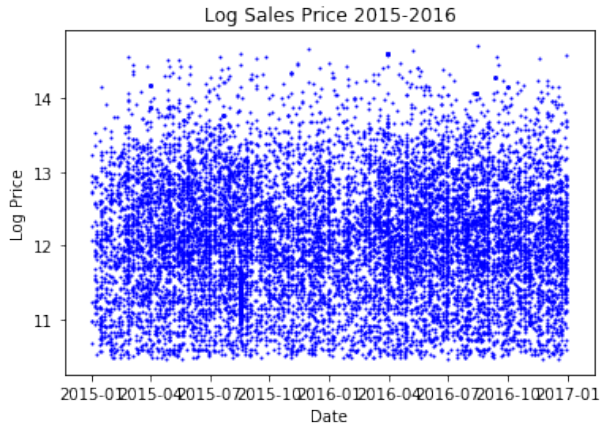
III. Dataset and Features

Deed transfers are recorded in the county assessor's office and are the ultimate authority on identifying the owner of a property as well as the transfer price since taxes are paid on the sales price. For this project county assessor records for Fulton county are used. The city of Atlanta and many of its outlying suburbs are located in Fulton county. The data was limited to January 1, 2015 to December 31, 2016. For the purposes of training and testing, the final 3 months were held out as the test set, while the 3 months preceding the test set were held out as the validation set for hyperparameter tuning. Within this time period there are approximately 53,000 rows of data. The dataset contains all basic information on the house such as the date it was sold, square footage, bedrooms, bathrooms, and flags for certain features such as pools, decks, and patios. Further, there are some key infrastructure notes such as heating type, style of the home, and property use. For categorical variables such as style of home, heating type, and flagged items such as pools were one-hot encoded into the feature matrix. All homes over 2,500,000 were removed from the dataset since they may represent a completely different sector of the market driven by its own dynamics. Typical relationships between square footage and price likely breakdown and higher end features and finishes which are not present in the dataset may drive prices. All homes less than 35,000 were also removed since the odds of a clerical error or a foreclosure sale increase and those are factors we are not looking to solve. Homes that were missing key pieces of information such as their longitude and latitude were also dropped since we will be relying on this information to create edges between properties. All transactions explicitly labelled as bankruptcies were also removed. This leaves approximately 30,000 transactions for 23,000 individual homes. Following are major basic statistical features of the dataset:









This is a small sample of the features in the dataset, but most are relatively normally distributed with a right-hand skew. The bathroom distribution is irregular and may relate to the layout styles of different homes. The log sales price is important since it shows that the prices remained roughly flat through the time period that we are looking to analyze.

The data still needs to be constructed in a way that we can use both graph models and baseline regression and decision tree models on. For this task, the data was split into two cohorts 2015 sales and 2016 sales. Each sale in 2016 was paired with the 10 nearest neighbors on a spatial basis using Euclidean distance based on the latitude and longitude of the home. All features of the 2015 sale were appended to the feature matrix of the 2016 sale. As a result, the feature matrix for each 2016 sale included 439 features. To convert that same information into a graphical form, for each of the 10 2015 sales which was found to be nearest to a 2016 sale a single directed edge was drawn from the 2015 sale to the 2016 sale. Therefore each 2016 sale had an incoming degree of 10, and an outgoing degree of 0. In this dataset construction, time is reduced to yearly intervals, but in future work more granular time period approaches can be tried and seasonality adjustments must be implemented. The target variable in the graphical, regression, and decision tree models is the price the home sold at.

IV. Models

In this paper we implement three different graphical models. The first is a simple Graph Convolutional Network (GCN) which propagates information from other nodes to the target node through neural network layers. The second is GraphSage, which builds upon a GCN and also incorporates neighborhood feature aggregation through summation or averaging. Finally, GAT which places weights on which neighbors have the most importance to the target node., will be implemented. Two different baselines will be implemented. The first will be a simple linear regression, and the second will be a random forest regression. Thus, we will be able to compare the performance of graphical methods to both regressions and decision tree based models.

A. Linear Regression

Linear regression is a good baseline model of prediction since it has no spatial or temporal component beyond what is fed into the input space through feature engineering. For inputs it used the features of the 10 nearest neighbors determined by Euclidean distance as well as the target home's features. Each of the nearest neighbors was a home that sold in 2015. The formula is given as:

$$(X^TX)^{-1}X^TY$$

B. Graph Convolutional Network (GCN)

Graph Convolutional Networks work in a semi-supervised manner where they propagate information through a neural network from a partially labelled dataset to the adjoining nodes based on node connectedness.[2] There is the implicit assumption that nodes that are connected share the same label. In this case we are tweaking the standard GCN slightly and turning the classification problem into a regression problem. Rather than attempting to label the node's class we will be trying to predict the nodes sale price. As a result the final layer in the neural network is not a softmax layer but a singular output and the loss used is MSE. The directed connections in the graph are from the labeled 2015 home sales to the unlabeled 2016 target nodes. Each home in the 2016 set has 10 incoming edges from its 10 closest spatial neighbors based on Euclidean distance. This basic setup remains true for all 3 graphical models. The formula for a GCN is given as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) .$$

H is the representation at the next layer, W is the weight matrix at the current layer, sigma is a non-linearity. In our implementation ReLU was used.

C. GraphSage

GraphSage is an extension of a Graph Convolutional Network (GCN). In GraphSage the target node's embedding is informed through the aggregation and sampling of the neighboring nodes.[3] The neural network layer still exists, but the inputs are enriched through the aggregation functions. As a result it tends to outperform in instances where each node has a rich feature representation. Given the data that we have for each home is rich, and so GraphSage should have strong performance. The formula is as given:

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot \text{MEAN}(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$$

This equation describes the mean aggregation of GraphSage. Here we are taking the node attributes of every neighbor of node 'v', concatenating it with the representation of node 'v' in the prior layer and multiplying by a weight matrix W before passing it through a non-linearity. Once again we used a ReLU.

D. Graph Attention Network (GAT)

Graph attention networks leverage masked self-attentional layers to attend over their neighborhoods' features. Thus they are able to implicitly specify different weights to different nodes in the neighborhood.[4] Research has shown that a multi-headed approach to training leads to greater stability and results. However, in this implementation for simplicity a single-headed approach was taken. This network has the advantage of being able to weight the inputs coming into the target node prior to making any sort of classification or in our case price prediction.

E. Random Forest

Random forests are a tree-based ensemble learning method. The model was born out of a desire to limit the overfitting done by classic decision trees, and as a result subsamples a part of the column space and creates weak learners. By creating N weak learnings and then averaging over the output of all of them, overfitting is reduced and better outputs are produced. We use a Random Forest regression to predict the output of the price, and have the depth of each weak learner to 8, and used 50 trees.

V. Results and Discussion

Table1: Nearest Neighbor Results

Model	Avg Error	Mean % Err
GCN	61k	34.9%
GraphSage	58k	33.5%
GAT	62k	35.5%
RF	69k	39.5%
LinearReg	83k	47.8%

Table2: Neural Network Parameter Specifications

Model	Layers	Hid D	LR	Wt Dec	Epochs
GCN	1	64	0.0001	0.005	500
GraphSage	1	64	0.0001	0.005	500
GAT	1	64	0.0001	0.005	500

The graphical models performed the best. Specifically, the performance order was GraphSage, followed by GCN, and finally GAT. The random forest model outperformed the linear regression, but both significantly trailed the graphical models.

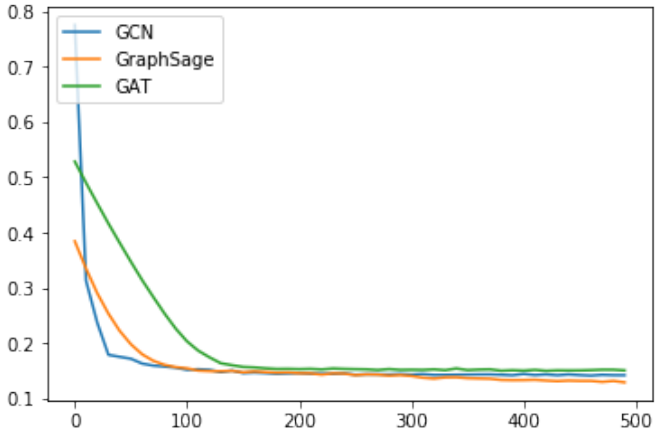


Figure 1 Nearest Neighbor Graph Network Performance Epochs vs Mean Average Log Error

The linear regression was the worst performer by a large margin although none of the models did particularly well. Even GraphSage which was the top performer had an overall error rate of over 30%. Compared to the error rates reported by the top AVMs on the market which are in the single digits, it appears that something is missing in this formulation of the experiment. One thing to consider is that while the median property price is 175k, there is a positive skew to the distribution of prices as the mean is 259k. However, the log transformation should have reduced the effect of this skew.

For the graph networks there is a large amount of potential hyperparameter tuning that can be done. In this case the, layers had to be 1 since only once set of neighbors from 2015 was connected to the target node. Creating more layers would not lead to additional learning since there would not be a larger neighborhood to pull from. To make an easier comparison between the models, the epochs and hidden dimensions were kept constant across the models. Given the error chart, it appears as if we could have continued to train the GraphSage model and achieved better results. For GCN and GAT it appears as if they had both already converged.

GAT is the most surprising of the models. Given that it has the ability to weight neighboring nodes based on how they relate to the target node. There was the expectation that it would perform at or better than GraphSage. A potential reason for this underperformance could be due to the implementation with a single-head versus a multi-head due to time constraints. Casting a wider net in hyperparameter tuning and moving away from a one size fits all approach may also prove fruitful.

Overall, it is somewhat surprising that all the networks in general were so far off from the actual target price. Given that most state of the art AVMs are in the single digits error range, all of these models significantly underperform. Is it possible that the closest homes on a spatial basis are not the best homes to create directed edges from?

This question led to a further experiment. Rather than using the 10 nearest neighbors from 2015 sales on a spatial basis, another dataset was constructed using the nearest neighbors based on Euclidean similarity of the features only. In this dataset geography was completely thrown out. A similar procedure to the prior dataset construction was completed, and the same models were retrained with this data.

Table 3: Similar Neighbor Results

Model	Avg Error	Mean % Err
GCN	53k	30.3%
GraphSage	47k	26.9%
GAT	54k	31.1%
RF	10.2k	4.7%
LinearReg	15.2k	8.3%

The results were surprising. This time the graph networks were the worst performers while a random forest was the best performer. The linear regression average error fell all the way to 15.2k for an average error of 8.3%. The random forest did even better with an average error of 10.2k for an average error of 4.7%. However, the graphical models did improve marginally versus the spatial dataset, but nowhere near the performance gains for the random forest and linear regression. GraphSage was once again the best performer with an error of 47k per home for an average error of 26.9%.

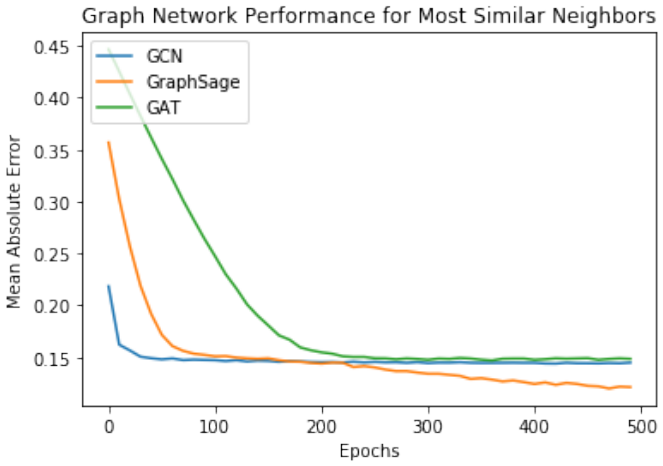


Figure 2 Most Similar Neighbor Graph Network Performance Epoch vs Mean Average Log Error

The same network parameters were used for the graphical models. It appears as though GraphSage could have continued to improve with additional training as it had not yet converged.

The results for the linear regression and random forest are inline with some of the best reported results for real estate AVMs. These results bring up an interesting question of whether or not spatial proximity or feature proximity matter more. The graphical models can clearly perform on either set albeit at a far worse level than that of the linear regression and random forest.

The random forest and linear regression shine when the most similar properties are input into the feature matrix. The question would be how far could we push the spatial relationship versus similarity tension. Since all of these homes are located within one county, there is a limit to how far one home can be from another. So, in this particular formulation of the problem similarity may strongly trump spatial proximity. If this experiment were run on several counties instead, spatial proximity may rise to the forefront.

These experiments also raise another question. How do we best draw edges between nodes. Here we took two approaches—node similarity and node spatial proximity. Despite the underwhelming performance of the graphical models, node similarity did to significantly better than spatial proximity. This makes intuitive sense since there is the underlying assumption that connected nodes would tend to share classification or in this case price ranges.

While validation and test splits were performed, there was still some curiosity as to whether some structure had been found in the market – could we take the model trained on the sales from 2015 to predict 2016 sales, and without any training use it to predict 2017 sales based on 2016 sales. The same exact dataset construction was done for feature similarity. In this case of the linear regression the average error was larger at 18.7k and the percent error at 10.8%. The random forest saw an increase in percent error to 8.9% and an average error of 15.5k. Clearly the similarity predicts prices across time periods. The best graphical model performed at a 25.0% error. While the graphical model was the worst performer it was very interesting to note that there was minimal drop-off. This may hint at the graphical models finding greater structure than the linear regression and random forest and an area for further study. Adding in additional ADP features such as income, unemployment, and various demographic features should allow us to further improve these performances significantly.

VI. Additional Work

The work done here lays the foundation for a much larger project down the road. One of the first areas for additional research is identifying the best way to create edges within the graphs. Even for the non-graphical models this is critically important. How do we choose which properties to use as inputs for our target property? Using embeddings and then identifying the closest properties in the embedding space could be a useful alternative to simple Euclidean distances. Or finding the nearest properties within some spatial distance, and leaving the spatial distance as a hyperparameter as a method of combining both approaches.

The layers right now are limited to one because of the way that the edges were created. For all of the properties sold in 2015, we could recursively connect neighbors based on which properties were most similar to them. This may give the graphical models an edge as they would become more expressive with additional layers. At the same time, care must be taken to ensure that a fully connected graph is not the result since that would lead to a generic rather than a specific price being output by the graphical models.

Neural networks are notoriously difficult to train. Additional architectures and a wider hyperparameter space would have to be tried to see if the graphical models failed to outperform simply due to poor training.

On the theme of creating edges, it would be interesting to see how to connect edges across time periods too. Could we improve performance by dividing 2015 into four quarters. Each quarter would connect to the 10 most similar properties in the quarter before. Then, when we are crafting a prediction in 2016 Q1 we could add more layers to the graph models and incorporate a time element. If this does add performance, venturing into Spatio-Temporal Graph Convolution Networks would be an exciting extension.

In this particular dataset the prices were stable between periods. However, additional research would have to be done to identify how to appropriately handle seasonality and a trending market.

Models such as Zillow use more than just price data. Income, demographic, landmark, school district, among other factors are all incorporated into their predictions. Adding these features to our nodes, would further enrich the information that we have, and should lead to better performance across all models.

While I still believe that graphical models have a role to play in this problem, the most immediate step would be to enrich the data, and use Random Forests or try XGBoost in other cities to ensure that Atlanta was not a fluke. The graphical models may have failed, but they inadvertently illuminated another path to potential success. The Random Forest model accuracy is very close to state of the art without incorporating additional data sources.

VII. References

1. “What's Zestimate? Zillow's Zestimate Accuracy.” Zillow, http://www.zillow.com/zestimate.

2. Michael D. Eriksen, et al. Contract Price Confirmation Bias: Evidence from Repeat Appraisals. Fannie Mae, 2016.

3. B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in Proc. of IJCAI, 2018, pp. 3634–3640.

4. Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907, 2016.

5. William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. Neural Information Processing Systems (NIPS), 2017

6. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” in Proc. of ICLR, 2017.

VIII. Notes

Real estate data is inherently very messy. This dataset was painstakingly cleaned from multiple hundreds of columns, and wrangled into the appropriate format to even run a model. I know the experiments may appear simple, but they took a lot of effort to run as a one person team.