An "Overview" of Pricing: Predicting Storage Unit Prices from Satellite Imagery

Project Category: Finance and Commerce

Madhav Goenka
SUNet ID: madhav01
Mechanical Engineering
Stanford University
madhav01@stanford.edu

1 Key Information

This project is not shared with another class.

2 Introduction

Satellite imagery is a rich source of information that can aid in public policy decision making. For example, poverty is not well characterized in much of the developing world, making it difficult to decide on an allocation of resources; this is where machine learning can help [1]. Using machine learning on high-quality satellite imagery (SIML) can take advantage of the richness and availability of the images to produce actionable insights on a huge variety of problems in just about any location.

Image processing, however, is not a simple task, especially in machine learning. To this end, we evaluate the approach outlined by Esther et al. [2], called "MOSAIKS," in our own prediction task. In this paper, the authors create place embeddings from satellite images. This one-time encoding transforms each image into a condensed vector, avoiding the costly manipulation of imagery downstream. The authors assert that their pre-made embeddings can be combined with any tabular data to improve prediction tasks with a geographic flavor.

In this project, we evaluate the efficacy of MOSAIKS in predicting the price of self-storage units at facilities throughout North America. Given tabular data containing basic information, such as square footage, of 3,000+ units, we evaluate the improvement the MOSAIKS embeddings add to the price prediction task. We compare this improvement to that generated by using other equally-accessible geographic-based information, namely, data from the American Community Survey (ACS).

The input to our algorithm was tabular data with various features of self-storage units, supplemented with either satellite image embeddings or ACS data (all of which is tabular). We then use Ridge Regression, Lasso Regression, and Gradient Boosting to output the predicted price of each self-storage unit in dollars.

3 Related Work

We examined a number of papers that attempt to predict an economic outcome based on data that is, at least partially, geographic in nature. For papers primarily using tabular data, we notice that they tend to use techniques like Support Vector Machine (SVM), Random Forest, Extremely Randomized Trees [3]. It is worth noting that in datasets with high dimensionality, tuning these models can be difficult (e.g., finding the correct kernel for an SVM, as in [4]). Moreover, among papers that use place embeddings, the ones we found tend to focus on distinguishing between different *types* of places, as opposed to predicting attributes of places themselves[5].

For papers that included satellite image analysis, we saw a number of other models, including deep learning models like Feed Forward NNs and Convolutional NNs. Models like this are trained on tasks such as mapping poverty[6], estimating barely yield[7], and detecting global dust aerosols[8]. These models tend to perform well when the images carry a lot of information that is not captured as well in tabular data.

4 Dataset and Features

Our dataset includes 33,085 self-storage units from North America (see Figure 3 for pictures of a few units). The core of the dataset comes from our project mentor, who pulled data from the websites of self-storage companies. This dataset includes features such as square footage, latitude and longitude, whether the unit is climate controlled, etc. The dataset also includes price, the outcome variable.

We supplemented this data with embeddings for each of the storage units using the longitude-latitude pair in our dataset. These were pulled from the MOSAIKS API. As another dataset, we supplemented the original data with roughly 1,000 features from the American Community Survey (ACS) for each longitude-latitude pair in our dataset. These include features such as the number of households with children under 18 in that census tract, the percent of the population with a disability, and many more.

To pre-process the core dataset, we removed columns that were strings, such as the city where the unit is. This meant that all of our place-related data was coming from the embeddings and ACS data.

Moreover, the 33,085 self-storage units come from 2,271 franchises. The training set contains 22,101 units from 1684 franchises. One evaluation set (which we call "in-distribution," or ID), contains 5,555 units from 427 of the facilities that also have units in the training set. Another evaluation set (whcih we call "out-of-distribution,", or OOD), contains 5,429 units from 160 franchises, none of which have units in the training set. We split both evaluation sets into sets to be used for validation and hold-out sets to be used for a final test of generalization (50-50 split). So, if an entrepreneur plans to open their own self-storage franchise, the test error from our model could be expected to give a reasonable guess of how far off from ideal its prediction on a new self-storage unit's price would be.

5 Methods

5.1 MOSAIKS Framework

We first outline how the place embeddings are created by the MOSAIKS framework. It begins with a set of satellite images $\{I_l\}_{l=1}^N$ for the N self-storage units. MOSAIKS generates task agnostic features $x(I_l)$ for each satellite image I_l by convolving $M \times M \times S$ "patch" P across entire image. M is the height and the width of the patch in units of pixels and S is the number of spectral bands.

Each patch P_k is randomly sampled from the set of training images $\{I_l\}_{l=1}^N$. In our analysis, we have used $M = 256$ (1km x 1km image of the self-storage unit from Google Static API) and $S = 3$ bands (red, blue, green). The inner-product of P_k with I_l for each of the bands is averaged and passed through a ReLU activation with bias $b_k = 1$ to give us the k^{th} feature $x_k(I_l)$. The dimension of the resulting feature space is K, the number of patches used, and we have $K = 4000$.

Next, we describe the machine learning algorithms we use for our prediction task.

5.2 Ridge and Lasso Regression

Ridge and Lasso regression are linear regression models with a small amount of bias introduced to reduce the complexity of a simple OLS model. Specifically, they are regularization techniques that reduces the complexity of the model by penalizing the values of the parameters of the model. Ridge regression decreases the complexity of a model but does not completely eliminate any features, whereas Lasso does completely eliminate features. Compared to OLS, the cost function for Ridge and Lasso is altered by adding the penalty term: L2 and L1 norm of the model coefficients respectively.

Ridge:

$$\sum_{i=1}^{M}(y_i - y_i')^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{n}\beta_j * x_{ij})^2 + \lambda * \sum_{j=0}^{n}\beta_j^2 \tag{1}$$

Lasso:

$$\sum_{i=1}^{M}(y_i - y_i')^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{n}\beta_j * x_{ij})^2 + \lambda * \sum_{j=0}^{n}|\beta_j| \tag{2}$$

where: y = value to be predicted; β =parameters of the model; x = features; λ = regularization rate

5.3 Gradient Boosting Decision Trees

Gradient boosting works by building simpler (weak) prediction models sequentially where each model tries to predict the error left over by the previous model. It has three essential elements: a loss function to be optimized, a weak learner to make predictions and an additive model to add weak learners to minimize the loss function using gradient descent.

The loss function in our case is the mean absolute error and decision trees are used as the weak learners. For the sake of brevity, we do not discuss decision trees in depth. A decision tree is a machine learning model that builds upon iteratively asking questions to partition data and reach a solution.

A gradient descent procedure is used to minimize the loss when adding the learners. The final prediction is made by taking the sum of all the individual learners.

We utilize eXtreme Gradient Boosting (XGBoost) for its superior compute speed and empirical performance over regression tasks for tabular data. XGBoost has a regularized learning objective for model addition. Formally, let $\hat{y}^{(t)}$ be the prediction of the i-th instance at the t-th iteration, we will need f_t to minimize the following objective

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y^{(t-1)}, \hat{y}^{(t-1)} + f_t(x_t)) + \Omega(f_t) \tag{3}$$

This means we greedily add the f_t that most improves our model and $\Omega(f_t)$ penalizes the complexity of the model (either L1 or L2 regularization). XGBoost also supports shrinkage and column subsampling that provides further regularization effect and helps better generalize.

6 Experiments and Results

Given that the model could be used practically in setting prices for self-storage units of a new franchise and outliers need not have outsized importance, **mean absolute error (MAE)**, with units in dollars, was chosen as the evaluation method for this problem statement.

Using different datasets, we compare the efficacy of different approaches. We first use just the self-storage unit core dataset (see "Tabular data" in Table 1) for our models. Next, we append the place embeddings from MOSAIKS and the ACS data separately onto the Tabular data to create two new datasets (see "Td w/ embed" and "Td w/ ACS" in Table 1, respectively) for evaluation.

We present our results on the OOD validation set in Table 1 below. We also evaluate the same models on the ID dataset, but do not include the results since they are not as generalizable; the general patterns still hold, but the MAE decreases by roughly 25% on average. We talk more about this in the Discussion section.

In Table 1, for all datasets, Ridge and Lasso (with regularization rate tuned on validation set) do not perform well due to their high bias and inability to model complex relationships between the price and the predictors. XGBoost performs the best across all datasets; all following discussion would be based on the XGBoost results.

Table 1: Mean Absolute Error (in Dollars) on OOD Validation Set

	Tabular data	Td w/ embed	Td w/ ACS
Ridge	69.89	66.11	73.38
Lasso	69.26	66.57	69.01
Gradient Boosting (XGB)	56.82	54.91	51.10

Compared to a naive estimator that always predicts the mean value of the dataset (MAE = 105.48), our best model, Gradient Boosting on tabular data with ACS data, had an error roughly 48% as large.

7 Discussion

XGBoost gives a MAE of $56.82 with just the core tabular data. The feature importance chart (Figure 1) shows that longitude, latitude, square area, length and width of the units are the most important features. This makes intuitive sense as we expect the geographic location of the unit to be the most important feature for price prediction.

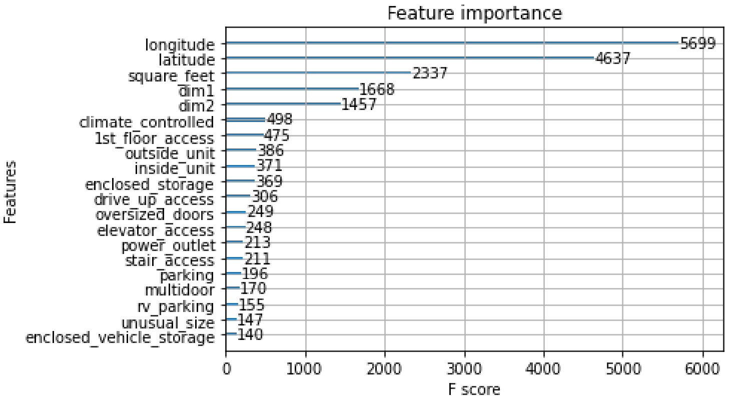

Figure 1: Feature Importance from XGBoost on Unsupplemented Tabular Data

Next, we notice that adding place embeddings leads to only a marginal reduction in MAE ($54.91 vs $56.82). This indicates that the place embeddings are not significantly helping with prediction. It is worth noting that when we supplement the data with place embeddings, only a few embedding features appear in the feature importance charts, and with moderate importance scores. We plot the price of the unit with these important feature embeddings (called "Emb 1," "Emb 2," and "Emb 3") in Figure 2; indeed, there doesn't appear to be much correlation between any of these features and price.

We also argue that the difference captured by the place embeddings in the satellite imagery of two storage units may not be indicative of the price of the unit. For example, consider the satellite imagery of two units with similar self-storage unit specific data, one located in Simpsonville, SC, the other in Queens, NY (Fig 2). The images appear similar and have similar embeddings, with the norm of the difference in their embeddings being just 3.2 compared to a mean embedding norm of 75.62. However, the unit in Simpsonville is priced at $148, while the unit in Queens is priced at $785. Conversely, for the units at Charlotte and Dover, the satellite images and embeddings are significantly different but have the same price of $166.

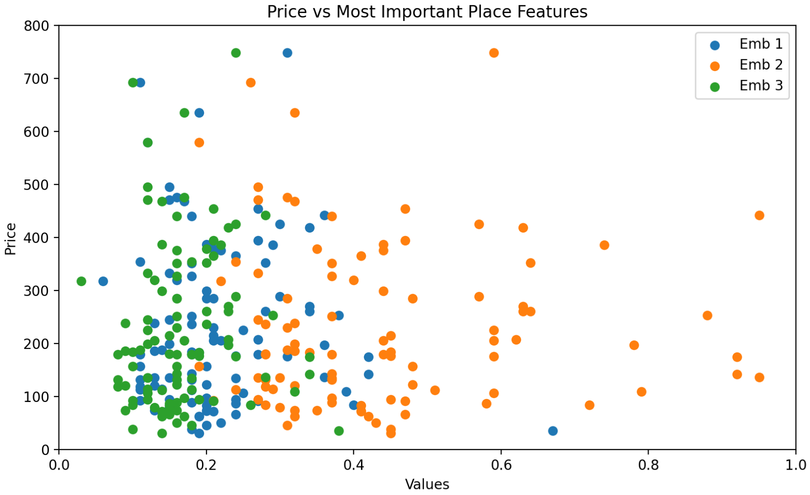

Figure 2: Price vs. Most Important Place Embedding Features

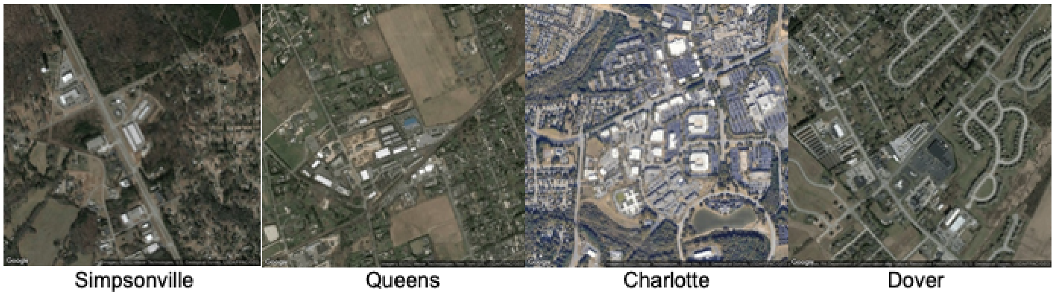

Figure 3: Select Satellite Images of Self-Storage Units

Finally, we see that a Gradient Boosting model on the tabular and ACS dataset performed the best. Given the observation above about the similarity of embeddings, we hypothesize that embedding data does not capture much of the variation in self-storage units relevant to price prediction. This could be due to the fact that prices are often more a function of the cost of living of a given area than the geographic features (water, trees, buildings, etc) of the immediate area. Consider that even in the highest cost-of-living areas, which tend to be dense cities like NYC, self-storage facilities tend to be out of the city center. Therefore, it is plausible that ACS data, which contains information about the types of residents in a given location, their income, etc, gives more relevant information for price prediction tasks than do embeddings. This is corroborated by the fact that the feature importance generated by XGBoost on the tabular + ACS data ranked household information (specifically, the total number of households in the locale, the number of households with a single male householder, and the number of households with a married couple) as being the three most important features, just below latitude and longitude, and above the square-footage of the unit.

In summary, the model with the lowest validation error was an XGBoost model on the core tabular data supplemented with ACS data. We tuned the model on the OOD validation data to get hyperparameters of: ratio of columns to be used when constructing each tree = 0.5, learning rate = 0.05,maximum tree depth for base decision tree learners = 5, minimum number of instances needed to be in each node = 500, and ratio of the training data to be sampled per iteration = 0.7.

When evaluated on the held-out test set, this model achieved an error of **50.23** which is close to the validation error of $51, suggesting that our model generalizes well. Given that this was evaluated on a hold-out, out-of-distribution test set, we expect the test error to be a good guess at the true error this model would achieve on a new self-storage unit. That is to say, if someone opening a self-storage unit used this model, using the same features, we would expect it to return the price that the unit should be set at to within $50.23 of the optimal price (assuming one exists, which is an economic question we leave to the reader).

8 Conclusion/Future Work

For the task of predicting self-storage unit prices from data about the units as well as the units' locations, we see that Gradient Boosting was the strongest performer. This likely has to do with its ability to learn complex relationships in the our highly-parameterized regime and avoid over-fitting. We saw that supplementing our core dataset, which contains basic information about each unit, with ACS data was more helpful in prediction than adding place embeddings, possibly due to the fact that demographic data of the broader area is more indicative of price than immediate geographic surroundings of the unit, as captured by satellite images.

References

[1] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[2] Esther Rolf, Jonathan Proctor, Tamma Carleton, Ian Bolliger, Vaishaal Shankar, Miyabi Ishihara, Benjamin Recht, and Solomon Hsiang. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature communications*, 12(1):1–11, 2021.

[3] Nari Kim and Yang-Won Lee. Machine learning approaches to corn yield estimation using satellite images and climate data: a case of iowa state. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 34(4):383–390, 2016.

[4] Marc Wieland and Massimiliano Pittore. Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images. *Remote Sensing*, 6(4):2912–2939, 2014.

[5] Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Song Gao. From itdl to place2vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *Proceedings of the 25th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 1–10, 2017.

[6] Isabelle Tingzon, Ardie Orden, Stephanie Sy, Vedran Sekara, Ingmar Weber, Masoomali Fatehkia, Manuel Garcia Herranz, and Dohyung Kim. Mapping poverty in the philippines using machine learning, satellite imagery, and crowd-sourced geospatial information. In *AI for Social Good ICML 2019 Workshop*, 2019.

[7] Alireza Sharifi. Yield prediction with machine learning algorithms and satellite images. *Journal of the Science of Food and Agriculture*, 101(3):891–896, 2021.

[8] Jangho Lee, Yingxi Rona Shi, Changjie Cai, Pubu Ciren, Jianwu Wang, Aryya Gangopadhyay, and Zhibo Zhang. Machine learning based algorithms for global dust aerosol detection from satellite images: Inter-comparisons and evaluation. *Remote Sensing*, 13(3):456, 2021.