Using Machine Learning Gradient Boosting to model commercial activities

Ricardo Ribeiro Barranco
European Commission - Joint Research Centre (JRC)
Via Enrico Fermi 2749
21027 Ispra, Italy
ricardo.barranco@ec.europa.eu

Abstract

As urban population grow, commercial activities play a fundamental role in providing goods, services and are part of cities fabric. Policy makers and urban planners need new tools to study these areas. This paper describes how machine learning can be applied to predict the density of commercial activities in cities. A supervised machine learning method called Gradient Boosting was applied to a group of spatial data sets. These were used to fit a model, allowing to analyse their combined interactions and predict commercial activities in London at 1 km^2 grid level for 2010 and 2030. The spatial resolution allowed comparing current and future trends. This type of activity is expected to lower closest to the city centre while increasing in further distant areas. Machine Learning has a great potential as a planning tool and the methodology presented in this paper could be further expanded to other cities.

Keywords: GIS, Machine Learning, Gradient Boosting, Land-Use, Spatial Planning, Commercial Activities

1 Introduction

1.1 Cities

Because of their importance and role as a primary "human ecosystem," cities have always attracted great attention from researchers from both the social and earth sciences. Cities have played a key role in human and social development throughout history and continue to do so, as they draw vast numbers of people into a safe, organized, and a culturally rich environment, enabling creative interaction, developing critical mass, and generating economies of scale (Bettencourt et al. 2007; Batty 2013).

With urban population expected to continue growing in coming decades (Bettencourt 2013), commercial activities play a fundamental role in providing goods, services and are part of cities' fabric. The study of these activities is a relevant matter for policy makers and urban planners. New tools are needed to answer the questions about current and future trends.

1.2 Machine Learning

The field of Statistics is constantly challenged by the problems that science, policy and industry brings to its door. With the advent of computers and the information age, statistical problems have exploded both in size and complexity. Vast amounts of data are being generated in many fields, and the statistician's job is to make sense of it all: to extract important patterns and trends, and understand "what the data says." We call this learning from data (Friedman et al. 2001).

This paper describes how machine learning can be applied to predict the densities of commercial activities in cities. It will start by describing a particular supervised machine learning method called Gradient Boosting. Then will give an overview of the data used. The methodological chapter will demonstrate how to combine the two. To finish, the visualisation and analysis of the results will help make an assessment of the predictions.

2 Machine Learning: Boosting

"Boosting" is a general method for improving the performance of any learning algorithm. Theoretically, boosting can significantly reduce the error of any "weak" learning algorithm that consistently generates classifiers which need only to be a little bit better than random guessing.

By repeatedly running a given weak learning algorithm on various distributions of the training data, and then combining the weak learner classifiers into a single composite classifier (Freund and Schapire 1996).

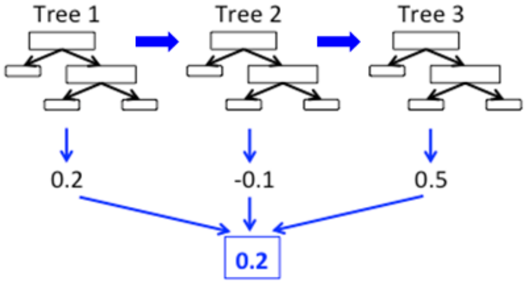
2.2 Gradient Boosting

Gradient boosting (GB) algorithm iteratively constructs and boosts a series of decision trees, each being trained and pruned on examples that have been filtered by previously trained trees. The incorrectly classified examples by the previous trees are resampled with higher probability to give a new probability distribution for the next ace in the ensemble to train on (Drucker 1995; Breiman 1997; Friedman 1997; Friedman 2001).

It constructs additive regression models by sequentially fitting a parameterized function (base learner) to current "pseudo'-residuals by least squares or other measurements at each iteration. The pseudo-residuals are the gradient of the loss functional being minimized, with respect to the model values at each training data point evaluated at the current step (Friedman 2002; Hastie 2009).

In the example decision trees ensemble regression bellow (figure 1), each tree predicts a real value. These three predictions are then combined to produce the ensemble's final prediction. The predictions combinations algorithms can use different techniques depending on the prediction task. In the case below the average between decision trees gives the final prediction.

Figure 1: Ensemble of decision trees used for regression.



Source: Joseph Bradley and Manish Amde posted in Engineering Blog January 21, 2015

Despite the potential benefits of promised by the theoretical results, the true practical value of can only be assessed by testing the method on real machine learning problems. The first provably effective boosting algorithms were presented by Schapire and Freund (Freund and Schapire 1996).

3 Data used

The data used in this paper was mainly generated by the LUISA (Land Use-based Integrated Sustainability Assessment) territorial modelling platform. This platform is part of the European Commission – Joint Research Centre (JRC) located in Ispra, Italy.

3.1 LUISA modelling platform

LUISA is primarily used for the ex-ante evaluation of EC policies that have a direct or indirect territorial impact. Beyond a traditional land use model, LUISA adopts a new approach towards activity-based modelling based upon the endogenous dynamic allocation of population, services and activities.

LUISA allocates (in space and time) the settlement of socio-economic activities (e.g. housing, industry, services, tourism, etc.) and the setting of infrastructures (e.g. for transport, energy, etc.) (Claudia et al. 2014).

3.2 LUISA outputs

The final output of LUISA is in the form of a set of spatially explicit indicators that can be grouped according to specific themes (land-use, bio-physical, ecological, economic, and social). The indicators are projected in time until typically year 2030 or 2050, and can be represented at various levels (grid, national, regional or other).

The following table 1 enumerates the specific datasets used and their type**:**

Table 1: Datasets and types used to create the model.

Dataset (code)	Type
Population (Pop)	integer
Accessibility (Ai)	float
Average Travelling Distance (Avd)	float
Maximum Travelling Distance (MaxDi)	float
Built-up percentage (BuiltUp)	integer
Distance to Functional Urban Area centre (LDist)	float
Distance to closest city centre (CDist)	float
Bus stop density (Bus)	integer
Km of local roads density (LocalRoad)	float
Gross Value Added (GVA)	float
Degree of urbanisation (DegUrb)	class
Building age (BuildAge)	class
Commercial activities density (ComDens)	integer

4 Methodology

4.1 Feature Engineering

The first typical step on machine learning involves harmonizing and pre-processing the data. This is communally known as "feature engineering" and can have an important influence on the predictions.

Because gradient boosting uses a numerical matrix as input, all used datasets were converted to 1 km^2 grid rasters. Knowing the longitude/latitude of each grid-cell, it was possible to create a matrix where each row represented a cell and the columns its feature values. Null values were converted to zero in order to fill-up any data gaps.

Since London Functional Urban Area (FUA) was selected for this exercise, only the grid cells within its boundaries were considered. These boundaries follow the most recent Organisation for Economic Co-operation and Development definition (OECD 2014).

4.2 Parameter optimization

The following methodological steps were done using the *scikit-learn* python package. It is an open-source, simple and efficient tools package, mainly focused for data mining and analysis. It is built on *NumPy*, *SciPy* and *matplotlib*.

Machine learning involves tuning parameters specific to each model. Gradient Boosting has around 15, each one with a range of possible values. Since these combinations can sum-up to several millions, there are several search techniques that go through these combinations and output accuracy measurements for each. This allows selecting potential models we might further test.

Randomized Search implements a randomized search over these parameters, where each setting is sampled from a distribution over possible parameter values. This has two main benefits over an exhaustive search (Bergstra and Bengio 2012):

- A budget can be chosen independent of the number of parameters and possible values;
- Adding parameters that do not influence the performance does not decrease efficiency.

Table 4.2 depicts the parameters tested, their values range and how they impact the model. In all the other parameters not included, the default value was used.

Table 4.2 Randomized Search parameters, used value, description and impact on the model

Parameter	Used Value	Description	Impact
n_estimators	[1000, 2000, 3000, 5000]	The number of trees to fit sequentially	-Tune using Cross-Validation for a given learning rate -Higher value for low learning rate but computationally expensive
learning_rate	[1, 0.1, 0.001, 0.0001]	The effect of each tree on the outcome is shrunk by this factor.	-Lower always preferred -Inversely proportional to n_estimators -Use high value for tuning and lower for final submissions
max_depth	[1, 2, 3, 4, 5, 6]	The maximum depth of each tree. None specified no limit on depth.	-Lower values prevent overfitting -Risk of underfitting with too low values -Tune using Cross-Validation -typical 5-20
subsample	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]	The fraction of observations to be used in individual tree	- Reduces variance in model - Tune using Cross-Validation -Typical value 0.8
loss	['ls', 'lad', 'huber', 'quantile']	The cost function to be minimized by optimization	-Use default value if not sure: -Class: deviance / exponential -Regressor: ls / lad / huber / quantile

4.3 Selected model

Using R^2 as score metric, from 200 tested models 3 were selected for further analysis. For these the Deviance was plotted and the best performing one was selected (figure 4.1). The final model had the following parameters:

alpha=0.9,criterion='friedma_mse',init=None,learning_rate=0.001,loss='ls',max_depth=3,max_features='sqrt',max_leaf_nodes=None,min_impurity_split=1e-07,min_samples_leaf=1,min_samples_split=2,min_weight_fraction_leaf=0.0,n_estimators=10000,presort='auto',random_state=None, subsample=1.0, verbose=0, warm_start=False).

4.4 Bias-Variance trade-off (under/overfit)

Once selected the model, the input data described on point 4.1 was split into Train/Test sets. Using the most common values in this procedure, the Train set represented 80% of the total while Test 20%. This step allows training a model on part of the data and then test it on a sample not known.

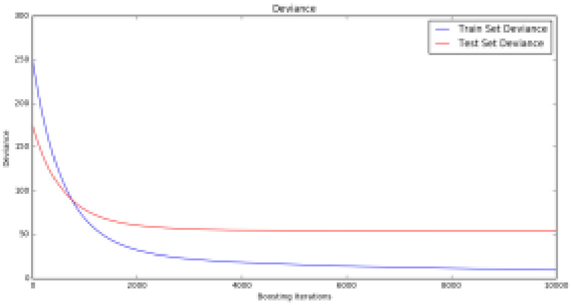
A model has strong bias when there is a strong error that is introduced by approximating a complicated relationship, by a much simple model. It is the difference between the truth and what you expect to learn. In this case the model is under-fitting.

By the contrary a strong variance is the amount by which a model would change if we estimated it using a different training data. If a model has high variance, then small changes in the training data can result in large changes in the model. This is also known as overfitting.

By plotting the Train/Test deviance for each number of estimators and seeing the difference between them, we can have an estimation if our model is under/overfitting.

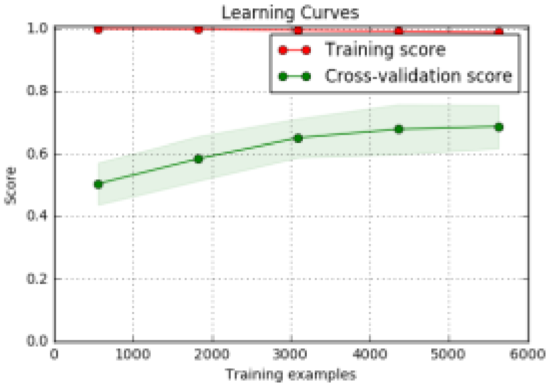
As it can be seen on figure 2 bellow, the train and test deviances follow the same trend and the gap between them is not too wide. Further, the test deviance line continues to lower and does not invert its sign. This demonstrate that the model is not under or overfitting.

Figure 2: Train and test sets deviance lines.



Using cross-validation (CV: a more robust train/test split technique) the normalized scores were plotted (figure 3). The test cross-validation reaches a R^2 score of 0.69. It can also be seen the model learning as more decision trees are constructed and ensembled. The green area surrounding the cross-validation score represents the standard variation and it is almost constant during the entire model fitting.

Figure 3: Training and cross-validation scores and standard deviation



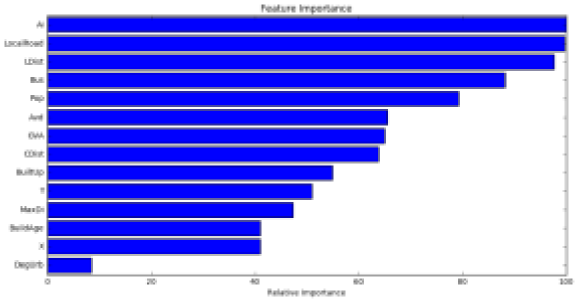
5 Model analysis

5.1 Feature importance

Features importance can be measured and ranked relative to the overall model fit using the corresponding inbuilt *scikit-learn* attribute. The feature with highest value is set to 100 and all others normalized relative to it. Figure 4 depicts the normalized values.

Accessibility, local road density, distance to Funtional Urban Area (FUA) centre, bus stops density and population were the most important when training this specific model. The Degree of Urbanisation (classification in 3 classes: Cities, Towns & Suburbs and Rural Areas (Dijkstra and Poelman 2014)) was ranked last. This might be partially related due being a class feature.

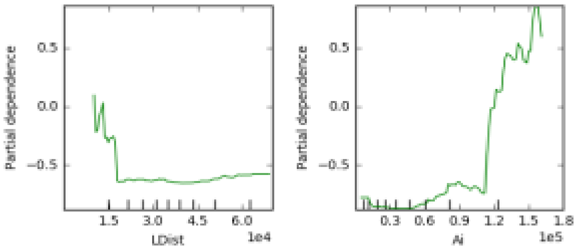
Figure 4: Normalized feature importance.



5.2 Partial dependence

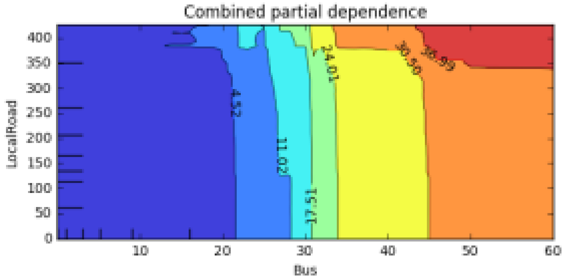
Partial dependence plots show the dependence between the target function and a set of features, marginalizing over the values of all the others. While the increase of distance to FUA centre influences negatively the density of commercial activities, the accessibility has an opposite effect. Higher values favour the allocation of commercial activities (figure 5).

Figure 5: Partial dependence for Distance to FUA and Accessibility.



Combining two variables can give a perspective on how they interact. Local road density only starts impacting the model when bus stops density is above the 15 and reach its highest when bus stops are at least 45 (figure 6).

Figure 6: Combined partial dependence between local roads and bus stops.



Applying the same approach, population levels above 250 and built-up percentages higher than 80% favour more significantly the presence of commercial activities on those cells.

6 Predictive modelling

6.1 Applying the model to 2030

Machine Learning is often synonymous of predictive modelling (Geisser 1995; Finlay 2014). This chapter demonstrates how the previous trained model was used to make future predictions.

Once the model was trained with LUISA 2010 datasets, it was then possible to use it to make predictions. By following the matrix structure presented at table 4.1 but using LUISA outputs for 2030, we could then ask the model to predict for each grid cell, the value of commercial activities. This assumed that the same interactions and dependences between our features and the predictor for 2010, would be kept for 2030.

An important advantage of Machine Learning is once trained a model, the prediction for a new dataset is done extremely fast. For London's 7035 grid cells, the model took less than 3 seconds to predict each value for 2030.

6.2 Mapping 2010 and 2030 commercial activities

Using the longitude/latitude coordinates for each cell, it was possible to study the changes between the two considered years. For that, the data was normalised. The 2010 cell with highest commercial activities was set at 100, while cells with non-existent activities were given 0. Applying the 2010 min-max normalisation to 2030, allowed their comparison. Figures 7 and 8 map London's 2010 and 2030 commercial activity and population per km^2.

Figure 7: London Commercial Activities and Population per km^2 for 2010

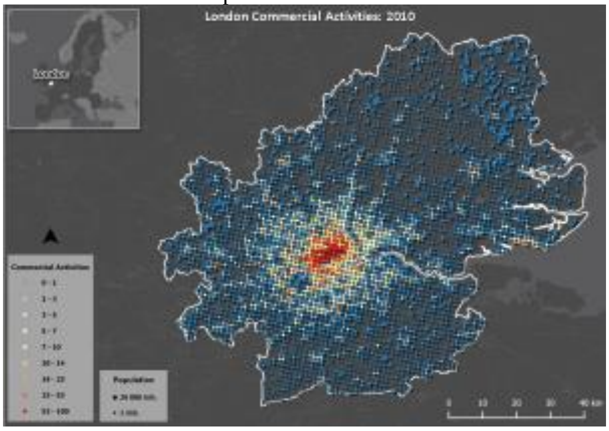
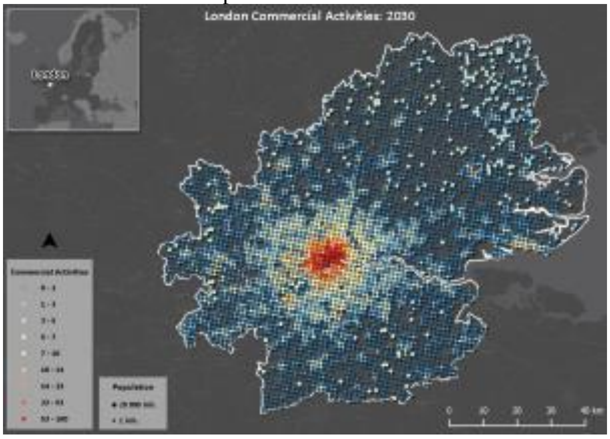


Figure 8: London Commercial Activities and Population per km^2 for 2030



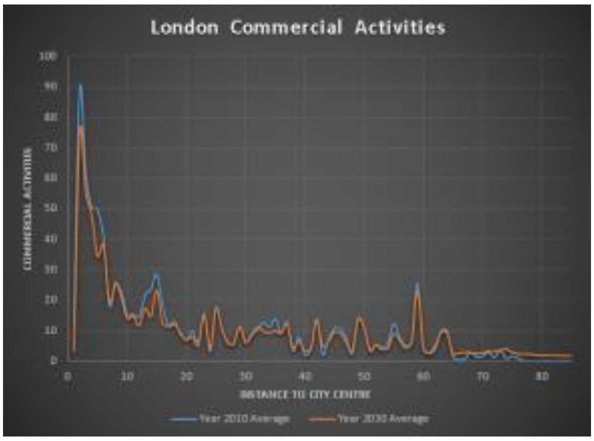
7 Results and discussion

At first glance, the gradient boosting model predicts some changes mainly in London's FUA centre and northeast areas. Natural breaks was used in the colour classification due the high values located in the centre.

7.1 Concentric circles

One way to measure the changes while moving further way from the city centre, is by calculating the average of commercial activities within each concentric circle. This circles are defined by having their centre in the FUA historical centre and by incrementing 1 km to the radius of the previous one. This was done till the last circle had a 85 km radius. This approach was executed on both 2010 and 2030 results and are represented on the following graph (figure 9).

Figure 9: London's commercial activities by concentric circles.



Comparing both years, the trained model predicts a reduction closest to the FUA centre (distances bellow 20 km), a stable situation for medium distances (between 20 and 65 km) and an increase in the activities for distant areas (above 65 km).

8 Conclusions

Machine Learning models, like gradient boosting, have potential to be used in urban planning and policy making. They can provide new ways for analysing in-depth the considered features, including their interactions and once properly trained and validated used to make predictions.

Policy scenarios can also be included. Using the case study described in this paper, future investments in road infrastructures, public transportation, and higher urban compactness could be reflected in the datasets used for the 2030 predictions.

Further developments include the application of this methodological approach to other cities and functional urban areas within the European Union. This would allow characterizing their current situation and future trends

Further, once commercial activities data is available for other years, the model could be trained on the real density values change and the predicted results compared. This would allow validating the model, further increasing its value as a planning tool.

References

Baranzelli, C., Jacobs-Crisioni, C., Batista, F., Perpiña Castillo, C., Barbosa, A., Torres, J. A., Lavalle, C., (2014) The Reference scenario in the LUISA platform – Updated configuration 2014 Towards a Common Baseline Scenario for EC Impact Assessment procedures. *European Commission Joint Research Centre Publications.*

Batty, M., (2013). A theory of city size. *Science 340*(6139): 1418–1419.

Bergstra, J. and Bengio, Y., (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), pp. 281-305.

Bettencourt, L.M. A., (2013). The origins of scaling in cities. *Science 340*(6139): 1438–1441

Bettencourt, L.M., Lobo J., Helbing D., Kühnert C., and Wes, G.B., (2007). Growth, innovation, scaling, and the pace of life in cities. *Proceedings of the National Academy of Sciences 104*(17): 7301–7306.

Breiman, L., (1997). Arcing the edge. *Technical Report 486*, Statistics Department, University of California at Berkeley.

Dijkstra, L. and Poelman, H., (2014). *A harmonised definition of cities and rural areas: the new degree of urbanisation, European Commission (No. 01).* Regional Policy Working Papers.

Drucker, H. and Cortes, C., (1995), November. Boosting decision trees. In *Proceedings of the 8th International Conference on Neural Information Processing Systems* (pp. 479-485). MIT Press.

Finlay, S., (2014). *Predictive analytics, data mining and big data: Myths, misconceptions and methods.* Springer.

Freund, Y. and Schapire, R.E., (1996), July. Experiments with a new boosting algorithm. *In Proceedings of the Thirteenth International Conference,* San Francisco (Vol. 96, pp. 148-156).

Friedman, J.H., Hastie, T. and Tibshirani, R., (2001). *The elements of statistical learning* (Vol. 1). Springer, Berlin: Springer series in statistics.

Friedman, J.H., (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp.1189-1232.

Friedman, J.H., (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), pp.367-378.

Geisser, S., (1995). Predictive inference: an introduction. *Journal of the Royal Statistical Society-Series A Statistics in Society, 158*(1), p.185.

Hastie T., Tibshirani R. and Friedman J., (2009). 10. Boosting and Additive Trees. *Elements of Statistical Learning Ed. 2*, Springer. pp. 337–384.

OECD (2013). *Definition of Functional Urban Areas (FUA) for the OECD metropolitan database. OECD Publications*.

Schapire, R.E. and Singer, Y., (1999). Improved boosting algorithms using confidence-rated predictions. *Machine learning, 37*(3), pp.297-336.