Performance Analysis of Some Machine Learning Algorithms for Regression Under Varying Spatial Autocorrelation

Sebastian F. Santibanez
Urban4M - Humboldt University of Berlin / Department of Geography
135 San Lorenzo #530, Coral Gables, FL 33173 USA.
sebastian.santibanez@hu-berlin.de

Tobia Lakes
Humboldt University of Berlin / Department of Geography
Unter den Linden 6, 10099, Berlin, Germany
tobia.lakes@geo.hu-berlin.de

Marius Kloft
Humboldt University of Berlin / Department of Computer Sciences
Unter den Linden 6, 10099, Berlin, Germany
kloft@hu-berlin.de

Abstract

Machine learning is a computational technology widely used in regression and classification tasks. One of the drawbacks of its use in the analysis of spatial variables is that machine learning algorithms are in general, not designed to deal with spatially autocorrelated data. This often causes the residuals to exhibit clustering, in clear violation of the condition of independent and identically distributed random variables. In this work we analyze the performance of some well-established Machine Learning algorithms and one spatial algorithm in regression tasks for situations where the data presents varying degrees of clustering. We defined "performance" as the goodness of fit achieved by an algorithm in conjunction with the degree of spatial association of the residuals. We generated a set of synthetic datasets with varying degrees of clustering and built regression models with synthetic autocorrelated explanatory variables and regression coefficients. We then solved these regression models with the algorithms chosen. We identified significant differences between the machine learning algorithms in their sensitivity to spatial autocorrelation and the achieved goodness of fit. We also exposed the superiority of machine learning algorithms over generalized least squares in both goodness of fit and residual spatial autocorrelation. Our findings can be useful in choosing the best regression algorithm for the analysis of spatial variables.

1 Introduction

The quantitative and spatial analysis of complex human-environment and ecological systems are often focused on predicting spatial variables [10]. For accomplishing these tasks diverse algorithms have been widely used in the literature [3, 4, 8, 10, Li]. The growing availability of spatial data and the high complexity of the coupled human-natural system [14] pose an added challenge to spatial analysis. Machine learning (ML) for its part, is rapidly gaining popularity for modeling complex phenomena and mining large datasets. In order for a modeling algorithm to be spatially adept, the residuals of a regression analysis should not exhibit clustering. In fact, the presence of spatial autocorrelation in the residual of a statistical model can lead to an erroneous interpretation of the results [11]. Unfortunately, as most ML algorithms are not designed to handle spatial data, there is no guarantee that the results of an analysis will be satisfactory from a geographic point of view; furthermore, despite the importance of its effects, the implications of spatial association in geographic modeling remains unclear [11].

In this paper we explore the performance of selected well-established ML algorithms for the regression of spatially autocorrelated data. We also compare the performance of ML algorithms against the spatial algorithm *generalized least squares* (GLS). Our research question is hence: How do different ML algorithms perform on spatially autocorrelated data?

2 Materials and Methods

We approach our research question by analyzing the performance of regression algorithms on synthetic datasets, which show a systematic variation on the degree of autocorrelation.

2.1 Synthetic data

We built a collection of simulated raster images (50 x 50 pixels) with varying degrees of spatial autocorrelation loosely following the approach on [2]. These raster maps can be seen as an unconditional gaussian simulation based on a defined kriging [16] structure. In order to obtain a variation of the level of clustering in the data we allowed the parameter "nugget" in the variogram to progressively account for 1%, 10%, 20%, 30%, 40% and 50% of its Sill. We kept the range parameter fixed to 15 pixels (~1/3 of the side of the generated surfaces). For each nugget value we generated 23 random simulations of clustered raster maps; 13 rasters to be used as regression coefficients and 10 as explanatory variables in a regression model that simulates a given spatial process.

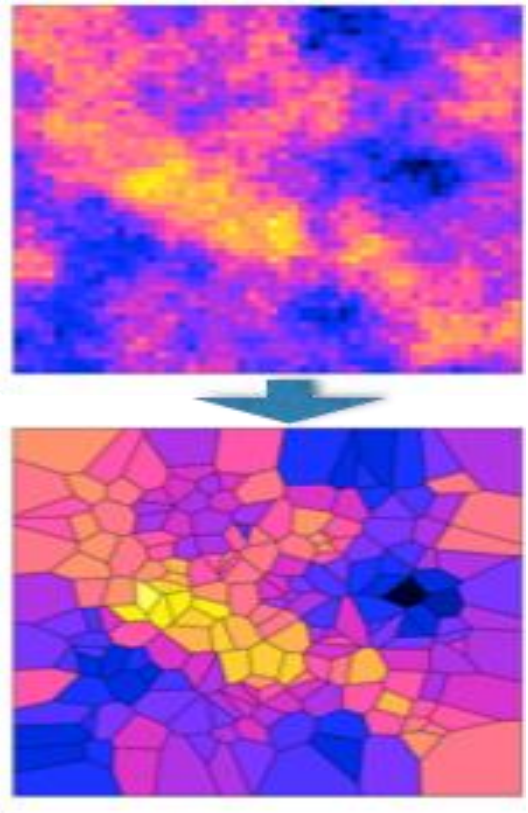
In order to simulate vector data we generated a layer of 200 Voronoi polygons from a two dimensional (X,Y) random uniform seed. We intersected each polygon with each raster created and extracted the mean value of the intersecting pixels (fig. 1).

The regression coefficients and explanatory variables, now aggregated in polygons, were combined to generate a synthetic response variable by using the following expression:

$$y_n = x_1 * \beta_1 + x_2 * \beta_2 + \cdots + x_{10} * \beta_{10} + x_1 * x_2 * \beta_{11} + x_3 * x_4 * \beta_{12} + x_5 * x_6 * \beta_{13} \quad (1)$$

Where:
$\beta_1 \ldots \beta_{13}$: Synthetic rasters of regression coefficients
$x_1 \ldots x_{10}$: Synthetic rasters of explanatory variables

Figure1: Raster map (up) and aggregated vector map (down) for nugget = 1%.



All synthetic data was generated in R with the library "gstat" [17]. The ML algorithms were implemented via "Caret" in R and the optimization was done through standard grid search with repeated 5-fold cross validation. GLS was implemented via "nlme" in R. The estimates for the ML algorithms were obtained through repeated 5-fold cross validation while the GLS solution was calculated with standard 5-fold cross validation.

2.2 Machine Learning Algorithms

For our study we selected the following well-established ML algorithms: Random Forest [5], Neural Network [19], Neural Network with PCA [19], Cubist (Improvement on M'5 algorithm by [18]), Partial Least Squares [7], Gradient Boosting Machine [13] and Support Vector Machines [9]. Also, we used GLS [6] as a spatial algorithm for comparison purposes.

Table 1 summarizes the algorithms used, the optimization parameters, and an example of its practical use in analysis of spatial variables.

Our criteria for assessing the performance of these algorithms are based on two different factors. Firstly we consider the goodness of fit reached by each algorithm in terms of R^2 and Normalized Root Mean Square Error (NRMSE). Secondly we measure the degree of Spatial Autocorrelation on the residuals of the regression as delivered by the Moran's I statistic.

Table 1. Algorithms used, optimization parameters and references in the literature.

Algorithm	Optimization Parameter	References in the Literature
RF	Mtry	[20]
NNET	Size, decay	[11,15]
NNET PCA	Size, decay	[15]
Cubist	Committees, neighbors	[20]
SVM with linear kernel	C	[15,20]
SVM RBF	C, sigma	[15,20]
PLS	Number of PC	[15]
GBM	ntree	[1]
GLS		[3]

3 Results

In general the algorithms Cubist, SVM with Radial Basis Function, NNET and NNET with PCA seem to perform better than the rest in terms of R^2 and NRMSE. On the other hand, GLS is the worse performer in R^2 and NRMSE.

Figures 2, 3 and 4, at the end of this document, summarize the results achieved.

Tables 2, 3 and 4 show the best and worse performer in R^2, NRMSE and residual SAC respectively.

As expected, goodness of fit is best when clustering in the data is highest. This reveals a drawback in the cross validation strategy that might lead to an overestimation of the quality of the regression, which is especially problematic when assessing the transferability of the models [21, 12].

The R^2 values range from 0.828 (best) to 0.510 (worst) while the NRMSE is between 0.077 (best) and 0.137 (worst). SVMRBF, NNET, NNETPCA and Cubist perform consistently well, especially under conditions of high clustering in the data. GLS on the other hand, performs worst and exhibits larger variances.

In terms of resistance to spatial autocorrelation, it is interesting how again the algorithms Cubist, SVM with Radial Basis Function, NNET and NNET with PCA seem to perform better than the rest. The spatial benchmark GLS, on the other hand, delivered again the worst resistance to residual SAC.

The differences in goodness of fit between the algorithms tested are more notorious in situations of higher clustering (Nugget = 1% and 10%). As random noise is added to the datasets -as an increase in the nugget parameter- the variances of the metrics calculated increase making the tested algorithms more comparable. Interestingly, the differences in resistance to residual SAC seem to remain clear even in conditions of relative high randomness in the data (Nugget = 40%).

Table 2. Best and worst R^2 for different values of the nugget parameters.

Nugget	Best R^2	Worst R^2
1%	SVMRBF (0.828)	GLS (0.613)
10%	SVMRBF (0.796)	GLS (0.590)
20%	NNETPCA (0.802)	GLS (0.613)
30%	SVMRBF (0.781)	GLS (0.640)
40%	NNETPCA (0.723)	GLS (0.555)
50%	Cubist (0.610)	GLS (0.510)

Table 3. Best and worst NRMSE for different values of the nugget parameters.

Nugget	Best NRMSE	Worst NRMSE
1%	SVMRBF (0.077)	GLS (0.118)
10%	SVMRBF (0.079)	GLS (0.119)
20%	NNETPCA (0.085)	GLS (0.142)
30%	SVMRBF (0.090)	GLS (0.119)
40%	NNETPCA (0.088)	GLS (0.120)
50%	Cubist (0.112)	GLS (0.137)

Table 4. Best and worst residual SAC for different values of the nugget parameters.

Nugget	Best Moran's I	Worst Moran's I
1%	SVMRBF (0.107)	GLS (0.587)
10%	SVMRBF (0.166)	GLS (0.498)
20%	NNETPCA (0.124)	GLS (0.481)
30%	SVMRBF (0.206)	GLS (0.450)
40%	NNETPCA (0.187)	GLS (0.429)
50%	Cubist (0.198)	GLS (0.357)

Our findings clearly show differences in performance of the tested ML algorithms for different degrees of spatial autocorrelation. This goes inline with earlier studies that have applied different ML algorithms for one regression task [11, 12]. However, what is new with this study is that we systematically assess goodness for different degrees of spatial autocorrelation.

Also, our findings reveal that the spatial method GLS delivers the worst performance in all the assessed situations.

We are aware, however, of some limitations in our study. First of all, we rely on synthetic data where the variogram model allows in theory, to control the amount of spatial autocorrelation that the data will exhibit. However when measuring the actual amount of spatial autocorrelation in the resulting simulated data, the differences for different nuggets are minimal, especially for Nugget = 1% and 10% (Table 5).

Table 5. Calculated Moran's I per varying nugget in the synthetic data

Nugget	Moran's I on simulated vector data
1%	0.6748462 +/- 0.065
10%	0.67419 +/- 0.087
20%	0.659541 +/- 0.067
30%	0.64000 +/- 0.081
40%	0.6073205 +/- 0.090
50%	0.5497491 +/- 0.084

Second, we acknowledge that the k-fold cross validation strategy might tend to deliver overly optimistic results as the records selected for testing are (at random) in the vicinity of the records selected for training, which causes unwanted correlations in the results leading to poor transferability of the models.

Third, our synthetic polygons are generated through a random uniform seed and therefore, its geometric structure might not faithfully represent real (plausible) spatial entities.

Finally, due to convergence issues, the GLS solution was obtained with regular 5-fold cross validation. This could partly explain the large variance in the GLS solution.

4 Conclusions and outlook

This brief study showed the differences in performance of some well-established ML algorithms when dealing with regression of spatial variables and how they compare against a well-known spatial algorithm. The results suggest that some ML algorithms are naturally more resistant to spatial autocorrelation. This is an interesting finding as it suggests the possibility of optimizing a regression task by selecting an adequate algorithm. Furthermore, the algorithms that exhibit the best goodness of fit (such as Cubist, SVMRBF, NNET and NNETPCA) are also the strongest in terms of resistance to spatial autocorrelation, which strongly suggest an overall superiority of the before mentioned algorithms. The chosen spatial algorithm (GLS) performed poorly probably because it is designed to handle SAC only in the residuals whereas our study incorporates SAC in the regression coefficients and explanatory variables.

The next step in our research will be to assess the performance of ML algorithms on real geographic data and compare it against a larger suite of spatial algorithms. Also, our future work will study the transferability of regression models built with different ML techniques; Finally, we will explore the benefits of blending results from different ML algorithms as ways to improve the goodness of fit and resistance to spatial autocorrelation in regression tasks.

5 References

[1] Bahn V., McGill B., Testing the predictive performance of distribution models. *Oikos*, 122(3):321-331, 2013.

[2] Beale C., et al., Regression Analysis of Spatial Data. *Ecology Letters*, 13:246-264, 2010.

[3] Begueria., et al., Modeling the Spatial Distribution of Soil Properties by Generalized Least Squares Regression: Towards a general theory of spatial variates. *Journal of Soil and Water Conservation*, 68(3):172-184, 2013

[4] Berwald J., et al., Using Machine Learning to Predict Catastrophes in Dynamical Systems. *Journal of Computational and Applied Mathematics*, 236(9):2235-2245, 2012

[5] Breiman L., Random Forests, *Machine Learning*, 45:5-32, 2001

[6] Browne MW., Generalized Least Squares Estimators in Analysis of Covariance Structures, *South African Statistical Journal*, 8(1):1-24, 1974

[7] Buphinder S., et al, Improved PLS Algorithms, *Journal of Chemometrics*, 11(1):73-85, 1998

[8] Cheong Y., et al. Assessment of land use factors associate with dengue cases in Malaysia using Boosted Regression Trees. *Spatial and Spatio-temporal Epidemiology*. 10:75-84, 2014.

[9] Cortes C., VapnikV., Support Vector Networks. *Machine Learning*, 20:273-297, 1995

[10] Cracknell M. & Reading A., Geological Mapping Using Remote Sensing Data: A Comparison of Five Machine Learning Algorithms, Their Response to Variations in the spatial distribution of training data and the use of explicit spatial information. *Computers & Geosciences*, 63:22-33, 2013.

[11] Dormann C., et al. Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. *Ecography*, 30:609-628, 2007.

[12] Elith J., & Leathwick J. Species Distribution Models: Ecological Explanation and Prediction Across Space and Time. *Annual Review of Ecology, Evolution, and Systematics*, 40(1):677-697, 2009.

[13] Friedman J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics.* 29(5):1189-1232

[14] Liu J., et al. Complexity of Coupled Human and Natural Systems, *Science*, 317:1513-1516, 2007.

[15] Okujeni A., et al. A comparison of advanced regression algorithms for quantifying urban land cover. *Remote Sensing*. 6:6324-6346, 2014.

[16] Oliver, M. A. Kriging: A Method of Interpolation for Geographical Information Systems. *International Journal of Geographic Information Systems*. 4:313–332, 1990.

[17] Pebesma, E.J. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30(7):683-691, 2004.

[18] Quinlan J.R., Learning with Continuous Classes, *Proceedings of 5th Australian Joint Conference on Artificial Intelligence, World Scientific, Singapore*, 1992

[19] Ripley B.D., Pattern Recognition and Neural Networks. Cambridge, 1996

[20] Walton J. Subpixel Urban Land Cover Estimation : Comparing Cubist , Random Forests , and Support Vector Regression, *Phtotogrammetric Engineering and Remote Sensing*, 74(10): 1213-1222, 2008.

[21] Wenger S. & Olden J., Assessing Transferability of Ecological Models: an Underappreciated Aspect of Statistical Validation.

Figure 2: R^2 per level of SAC in the synthetic data and per algorithm

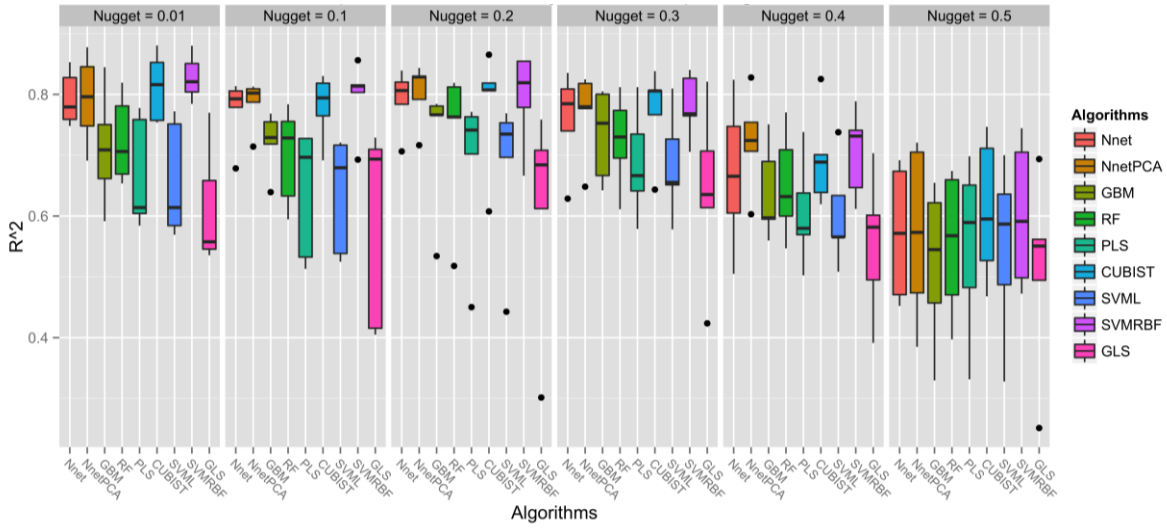


Figure 3: NRMSE per level of SAC in the synthetic data and per algorithm

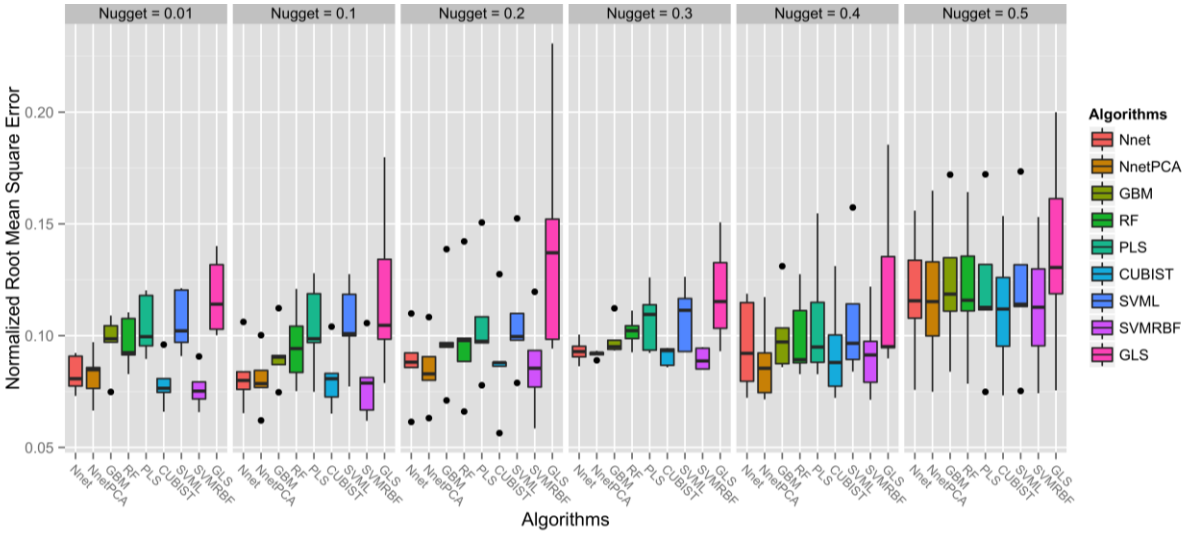


Figure 4: Residual SAC per level of SAC in the synthetic data and per algorithm