

Myths of Segregation

Explaining and Predicting Urban Segregation

Solving the problem of urban segregation (the emergence of “black” and “white” neighborhoods) has been a major concern for policy makers anyplace and anytime. It is often thought that this problem is the result of racism. Using a computer model, Schelling showed that this is not true: in his simple but effective simulation even fairly tolerant societies end up segregated.

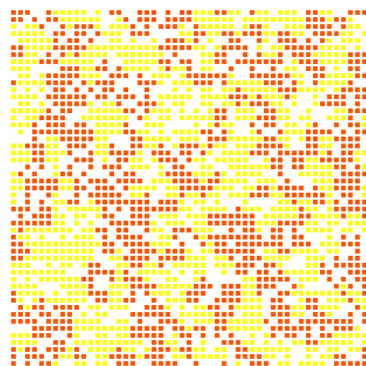
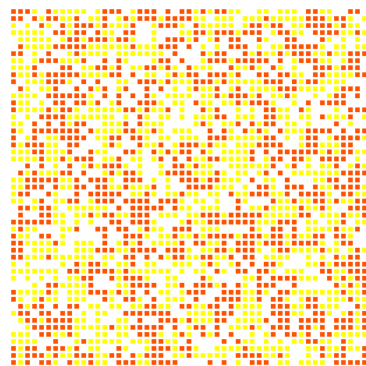
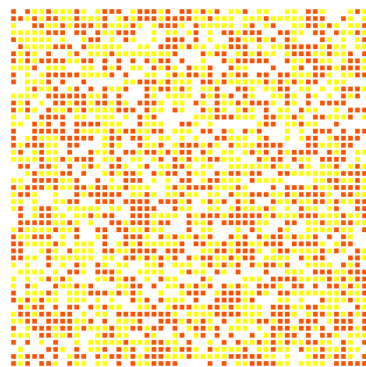
In this project, I set out solving the segregation problem by extending Schelling’s computer model, and analysing it’s results using machine learning algorithms.



Although unfortunately this study does not provide any major breakthroughs, it is an interesting trail-blazing experience of this unique combination of tools to solve sociological problems.

The emergence

of segregation:



The Problem

Schelling showed that segregation is not caused by racist beliefs. He invented Schellingdale: a grid with actors, where every actor wants to live in a neighborhood (adjacent cells of the grid) with at least 30% of his own “color”; something that cannot be considered very racist. However, if we distribute them randomly over the grid, some actors will inevitably be “unhappy” because they live with less than 30% of their own kind. If we let these people move, some of them will settle in a place where they cause someone from the other “color” to move. This induces a chain reaction in which the actors will end up with about 80% of their own kind in their hood: Urban segregation, without the “help” of racism. (Schelling 1971; Schelling 1978)

The Extension

Others have extended Schelling’s models to include other important variables, but have concluded that the segregation phenomenon is very persistent. My extensions seem to have significant effects. A Linear Regression on the outcomes of my extended model gave surprisingly good results. One would ideally want to find out under what conditions segregation does not occur. However, since my variables are bounded the results do not generalize to extreme conditions,

one of which is “no segregation”. Inferring a situation like that from the linear model will not give accurate predictions. It is thus impossible to “solve” the problem with this simple model.

Dependent variable: percent-similar

	B	SE B	β
Constant	59.372	3.888	
%-different-wanted	-.265	.026	-.311**
%-similar-wanted	.666	.033	.652**
Types	-4.746	.646	-.226**
%-covered	-.106	.045	-.069*

Note: R2 = .810; * p < 0.05; ** p < 0.001

Predicting Regions of Segregation I and II

The first model was not very useful, since the border conditions that interested me got a lot of prediction error.

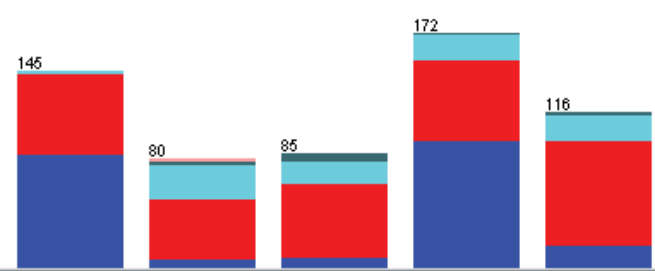
Therefore, I set out to solve another prediction problem: predicting the end-segregation in a certain region of the grid based on the setup-situation. I let the unhappy actors move to the nearest patch where they will be happy.

Furthermore, I let them look around in a bigger radius (bigger “hood-size”). I extracted setup-situation variables from the center (inner) region and immediate surroundings (outer). In order to get a better idea of the structure of this prediction problem, I first ran a clustering algorithm and an association rule learner on the dataset.

Clustering

I tried to make sense of my data using Simple K-means clustering. Experimenting with the number of classes, I found that having 5 clusters seemed to give me the most elegant clusters.

An even more simple solution was found when I restricted the algorithm to only use my first four variables (initial inner group segregation, ratio, % unhappy, and empty). This means that these four variables probably provide the clearest view of the final segregation.



Association

An association rule learner was run on the dataset to investigate the inherent structural relationship of variables. Discretizing all my variables into 3 equal segments seemed to work best for my dataset. In conjunction, the confidence of the association learner was set to 0.7 in order to come up with a reasonable amount of rules.

The rules found did not include my outcome variable, which means that the relationship between my predictors and the outcome is a non-trivial one. Furthermore, half of the rules found showed an interaction between the inner and outer region, which means that there is a lot of interaction between my variables, even between those that on the surface look unrelated.

Implications from Clustering and Association

From the clustering algorithm I learned that the dataset has a few variables that are more important than the others. Clustering matched some of the structure of my outcome variable, showing that the setup-situation can be used to (at least partly) predict the end-segregation. The association rules showed that there are interactions between my independent variables. This shows that a simple linear model is probably insufficient to get a good prediction for this problem.

Algorithm Optimization

Using a too complex algorithm, however, will make it overfit the training data, making it perform worse in general. In order to find the optimal level of complexity, I optimized two algorithms that allow for a wide range of complexity: Support Vector Regression (SMOreg) and a Locally Weighted Learning meta-learner with Linear Regression. Both optimization procedures showed that the algorithms can easily overfit the data; The best LVL algorithm did not make use of local weighing, and the SMOreg performed best when the model had the “flattest” settings.

Ensemble Methods

Another way of allowing more specific models on the data is by linearly combining the numeric predictions of other (non-linear) models. I optimized a linear combination of REPTrees, and found that this did not exceed the performance of the SMOreg model described above.

Actually none of the optimized algorithms outperformed the baseline performance of the unoptimized linear regression model. I suspect that the amount of randomness in my data prevents a more detailed model to be learned.

Using 5 unstratified train-test pairs

Algorithm	r^2
Unoptimized linear regression	0.6749
Optimized SMOreg	0.6736
Optimized LVL+Linreg	0.6718
Optimized AdditiveReg+REPTree	0.5966

Note: optimized algorithms use different settings per fold.

Future directions

Combining sociological computer models with machine learning algorithms is an interesting approach. Based on my findings, I propose some recommendations for future directions in this field of study.

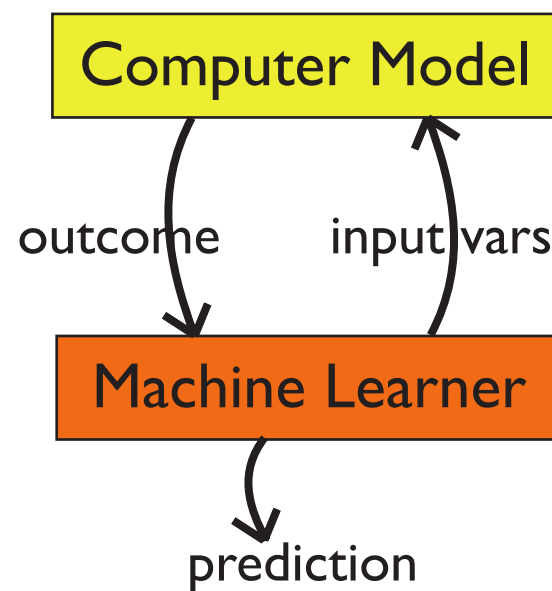
Using Binary Data

The computer model used produces noisy data. Although the best prediction might be complex and non-linear, it will get “caught” for overfitting noise.

Since I am interested in the distinction between “no segregation” and “any segregation”, I can dichotomize my outcome variable. This makes the algorithms less likely to overfit random fluctuations in end-segregation.

Conclusion

Although I was not able to solve the problem of urban segregation (which is, admittedly, quite an audacious goal), I found that using a combination of sociological computer models and machine learning showed some interesting new insights in this field. This unique way of modeling and predicting social problems seems to be promising.



No-nearest Association

Although the association learning indicated interactions between independent variables, the regression algorithms did not confirm this. It may be that the interaction found were actually just based on random noise. Possible (real) interactions exist because unhappy actors move to the nearest patch. Interactions are real if they decrease by removing this condition, otherwise they are probably noise.

Automatic Modeling and Prediction

One advantage of using a computer model for machine learning, is that you have control over the independent variables. If a learning algorithm “misses” parts of the data space, they can easily be provided. This process can be automated as shown in the diagram above.