Advanced Statistical Analysis & Machine Learning Evaluation

Sophie Gallet

Professor Christine Tuleau-Malot

Data ScienceTech Institute - 2020

Summary

Summary of the Multiple Tests based on a Gaussian Approximation of the Unitary Events method

Overview

The Multiple Tests based on a Gaussian Approximation of the Unitary Events (MTGAUE) method builds upon the Unitary Events (UE) method to detect synchronized spikes in different neurons. It aims at combining statistical rigor with well-designed cognitive neuroscience protocols.

Contrary to the assumption of the UE method, and while individual spike trains are still assumed to follow a Poisson distribution, it is shown that the coincidence count cannot be a Poisson process. Gaussian approximation is used instead, along with a different definition of coincidence counts: the delayed coincidence count. Benjamini and Hochberg multiple testing procedure is then added to control the False Discovery Rate.

An extensive simulation study showcases the accuracy and reliability of MTGAUE over a wide range of parameters (firing rates, delays, etc.) and the shortcoming of the symmetric and lower values tests of the UE method. The simulation finds that the detections by the UE tests by upper value are correct under certain conditions. Finally, real data is used to present practical tools to use the MTGAUE method.

Introduction

Theoretical aspects

Real-time information processing in the brain is coded by neurons' spike trains and more precisely specific features, such as firing rate and temporal structure. Looking beyond individual neurons, it is now widely accepted that spike trains from two or more neurons can synchronize. The question is how to study this synchronisation, given the current technological limitations for neuronal data collection. The authors aim to combine rigorous statistical analysis with precise cognitive neurosciences protocols in order to study spike trains from several neurons recorded simultaneously with accuracy.

Technical aspects

All methods described in this paper focus on understanding neurons' functional connectivities, as opposed to the neuronal network's anatomical structure, considered out of reach.

The first spike synchronisation methods to be developed were the cross-correlogram followed by the joint-PSTH, both detecting coincidences between spikes of two neurons. The first uses averaging over trials - thus suffering from dilution of coincident firing over short periods, and from the unrealistic assumption of stationary activity. The second corrected these issues by providing a matrix representation - proving problematic for statistical analysis and requiring more spikes than the cross-correlogram in the absence of averaging.

The next generation of methods, Unitary Events analysis with binning and its improvement multiple shifts Unitary Events, were designed to precisely detect spike coincidences between two or more neurons and present the advantage of dealing with non-stationarity along with being compatible with statistical analysis. However, they assume not only the individual neuron spike to follow a Poisson distribution, but the coincidence counts as well. The latter assumption is proven wrong in this paper, justifying the need for a new method.

Like the UE method, the authors' proposed method assumes a Poisson distribution for individual neuron spikes, but then uses Gaussian approximation for the coincidence counts, which allows for a wider range of estimated parameters. It also leverages Benjamini and Hochberg multiple testing procedure to control the False Discovery Rate.

Single test of independence

The authors first describe their method on a single window [a,b]. The test's null hypothesis is that the spike trains of the two neurons, N_1 and N_2, are independent on [a,b] (H_0), and the alternative that they're dependent on [a,b] (H_a). It is assumed that N_1 and N_2 are poisson variables (assumption 1) and stationary on the given window (assumption 2). M trials are performed.

To proceed forward with the test, the coincidence count must be defined. Instead of the definition of coincidence counts under the binning framework, which leads to information loss and requires a plug-in step that can modify the law, authors chose the delayed coincidence counts definition, developed in the multiple shifts methods. The coincidences count with delay δ on the window [a, b], denoted X , is defined by:

$$X = \int_{[a,b]^2} 1_{|x-y|\leq\delta}\, dN_1(x)\, dN_2(y),$$ where dN_1 (resp. dN_2) is the point measure associated with N_1 (resp. N_2).

If assumptions 1 and 2 are verified, and assuming that the delay is smaller than half the window size (assumption 3), then theorem 1 gives the expectation and variance of X:

$$m_0 = E[X] = \lambda_1\lambda_2[2\delta(b-a) - \delta^2]$$

$$Var(X) = E[X] + [\lambda_1{}^2\lambda_2 + \lambda_1\lambda_2{}^2][4\delta^2(b-a) - \tfrac{10}{3}\delta^3]$$

An important consequence of theorem 1 is that X cannot be a Poisson process.

Test of independence on a window [a, b]

Under the same assumptions as theorem 1, theorem 2 gives the following convergence in law:

$$\sqrt{M}\frac{(\bar{m}-\hat{m}_0)}{\sqrt{\sigma^2}} \xrightarrow{L} N(0,1), \text{ where } \widehat{\sigma^2} \text{ is the estimate for } \sigma^2$$ *(for complete expressions, see paper)*

It results from theorem 2 that the plug-in step $m_0 \rightarrow \hat{m}_0$ changes the variance's shape, while substituting $\sigma^2 \rightarrow \hat{\sigma^2}$ is the same asymptotically.

The Gaussian Approximation of the Unitary Events method (GAUE) contains three tests:

- the symmetric test $\Delta_{GAUE}{}^{sym}(\alpha)$ rejects H_0 when $\overline{m}$ *and* $\hat{m}_0$ are too different,
- the unilateral test by upper values $\Delta_{GAUE}{}^{+}(\alpha)$ rejects H_0 when $\overline{m}$ is too large,
- the unilateral test by lower values $\Delta_{GAUE}{}^{-}(\alpha)$ rejects H_0 when $\overline{m}$ is too small.

Theorem 2 indicates that the three GAUE tests are of level α asymptotically: the Type I error rate of these test is asymptotically less than α. The goal is to yield a Type I error as close to 5% as possible: a higher value means the test cannot be trusted, a lower one means we could miss adequate rejections (the test is too conservative). A simulation is performed to verify the Type I error rate of both methods: the GAUE tests yield a Type I error close to 5% in most situations, less when the number of points is small. On the other hand, UE tests don't perform as desired: the symmetric and lower cases lead to high Type I error, while the upper case is too conservative.

For both methods, the symmetric test encompasses both the upper and lower cases at a smaller level, and as such will be our focus for the rest.

Detection of Local Dependence on [0, T]

As stationarity cannot be assumed on [0,T], the window is broken down into smaller windows for which the assumption is (almost) true. The next step is to control detections.

The authors prefer the Benjamini-Hochberg procedure controlling the False Discovery Rate (FDR) over the FWER test. Translated for the context, if the p-values are independent and identically distributed (iid) under H_0 then the multiple testing procedure ensures a FDR lower than q. When combining the procedure with the GAUE tests, the assumptions regarding the p-values for Benjamini-Hochberg are not exactly met, but the simulation study shows that the impact of this is limited.

Simulation

The simulated data is created with the Hawkes process, allowing for randomness in the delay and in the number of additional coincidences, which isn't the case of the injection model used in the original UE paper.

First, one run of 1,900 tests on overlapping windows is performed in order to compare MTGAUE and UE's detection performances for a wide range of parameters. In almost all cases, MTGAUE performs better than UE: for various delays (especially higher than 0.02s), levels of interaction (especially small), number of trials (especially small), desired FDR levels (MTGAUE is stable, UE isn't). In some cases, especially when the delay is not too large, the UE method by upper values performs as well as MTGAUE. The symmetric and lower values tests prove to be unreliable.

Next, a thousand runs of simulations are performed to compute the FDR level, set for 0.05. The MTGAUE and the UE method by upper values respects this controlled value, while the UE by lower values and the symmetric tests do not. Finally, the study looks at negative interactions, which force a low number of coincidences and make detections more difficult. In this context, the MTGAUE method is robust, while the UE method isn't.

In conclusion, the MTGAUE method is able to make detections with a controlled FDR over a large range of parameters. The UE method is only reliable under certain conditions.

Real data

In the absence of real "labeled" data, the aim of this section is to provide practical tools to evaluate detections' quality.

The p-value is a classical criterion to assess the quality of detections, usually with a threshold of 0.05. The multiple testing context requires to use multiplicity adjusted p-value. Detections made by MTGAUE with a FDR level of q=0.05 correspond to detections with a multiplicity adjusted p-value under 0.05.

Next, the correct way to use MTGAUE is with symmetric tests. Once a multiplicity adjusted p-value has been assigned to the different windows, one can investigate whether detections are linked to a too high or a too low coincidence count (most common, up to 40% of the time).

Discussion and conclusion

The UE is a popular method with some shortcomings: the assumption that both individual spike trains and coincidence counts are Poisson processes, the plug-in step, the neglection of the effects of a large delay and, despite applying the UE method simultaneously on various sliding windows, the lack of control of the False Discovery Rate.

The authors here prove the existence of an edge effect and propose a precise statistical method involving Gaussian approximation and Benjamini-Hochberg multiple testing procedure. The new method is backed by an extensive simulation study showing that MTGAUE can reliably deal with larger datasets and a wide range of parameters.

Reference

Christine Tuleau-Malot, Amel Rouis, Patricia Reynaud-Bouret, Franck Grammont. Multiple Tests based on a Gaussian Approximation of the Unitary Events method. 2012. hal-00757323v1

Procespin

Introduction and Exploratory Data Analysis

The goal of this study is to propose a model between ln(y) and the explanatory variables x1, .., x10 from the procespin dataset.

Our dataset contains a small number of observations, 33, and 10 explanatory variables. All are quantitative, as is our response variable, ln(y). There are no missing values. We do not have any context on the data, nor on whether we're in an interpretative or predictive mindset.

A first look at the variables shows a correlation between the response variable and several explanatory variables such as X1, X3, X6 and X9, as well as strong correlations and/or linear relationships between some of the response variables ((X3, X6, X8, X9), (X4, X5)).

Candidate Models

We now fit several models on the available data to then be able to compare them.

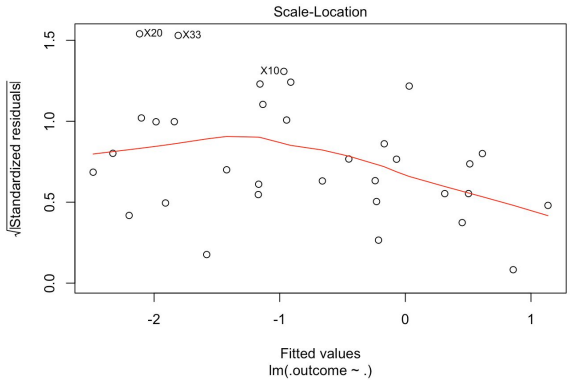
In each case, we use k-fold cross-validation - via the CARET package - as it enables us to use all our data to train the models and provides for a more reliable estimate than a simple train/test split.

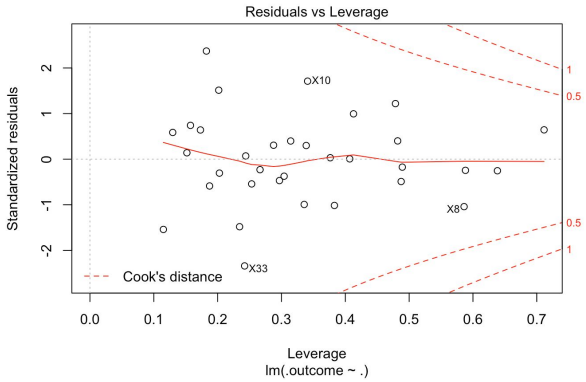
In terms of performance metrics, the root mean square error (RMSE) has the advantage of being in the same unit as the response variable, making it easy to understand. The R squared provides a measure of how much of the response variable's variance is explained by the explanatory variables. Both are averaged over the number of folds performed in the cross-validation method, and as such, are somewhat stable. Looking at the standard deviations for both metrics will provide additional information on the variation of performance over the data selected for training.

Full Linear Regression Model

The full model is a multiple linear regression model that uses the 10 explanatory variables, and as such is complex and likely suffers from multicollinearity in the explanatory variables. However, it is a good benchmark to start with. Its averaged performance and variance are summarized below, along with plots of the residuals. Residuals appear somewhat normally distributed and centered around 0, with no clear pattern, their variance can be considered constant (homoscedasticity). The assumptions of the absence of multicollinearity among explanatory variables is not met as seen in the EDA.

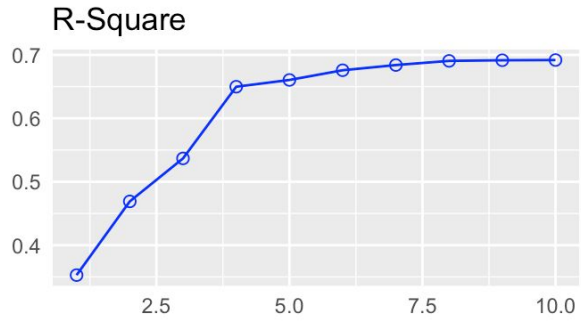
```
##   intercept      RMSE  Rsquared       MAE    RMSESD RsquaredSD     MAESD
## 1      TRUE 0.9840748 0.5619196 0.8194569 0.3436012   0.270563 0.3133206
```

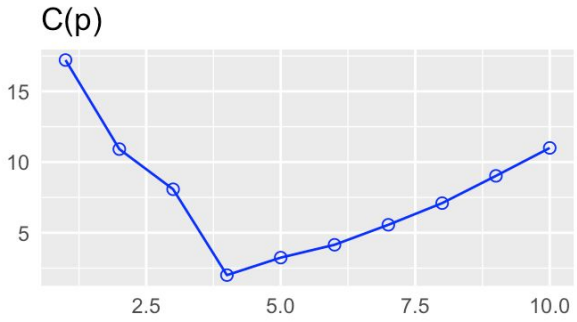


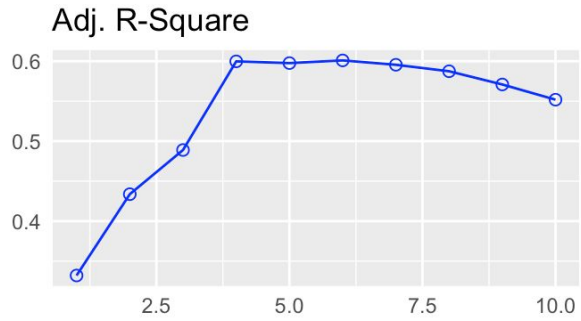


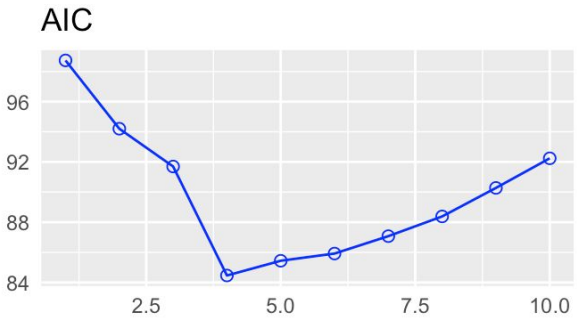
Variable Selection Overview

Using the convenient ols_step_best_subset() function that provides performance metrics for models of increasing complexity, we can see that all performance metrics improve considerably when adding predictors up to 4 variables, after which they either continue improving at a reduced rate or degrade. As we're looking to strike a balance between model sparsity and performance, the 4-variable model seems interesting. It showcases the highest Rsquared, the lowest AIC and the lowest MSEP (estimated error of prediction, assuming multivariate normality). The 4 selected variables are x1, x2, x4, x5.









Best Subsets Regression

Model Index	Predictors
1	x9
2	x1 x9
3	x1 x4 x5
4	x1 x2 x4 x5
5	x1 x2 x4 x5 x9
6	x1 x2 x4 x5 x6 x9
7	x1 x2 x3 x4 x5 x9 x10
8	x1 x2 x3 x4 x5 x8 x9 x10
9	x1 x2 x3 x4 x5 x6 x8 x9 x10
10	x1 x2 x3 x4 x5 x6 x7 x8 x9 x10

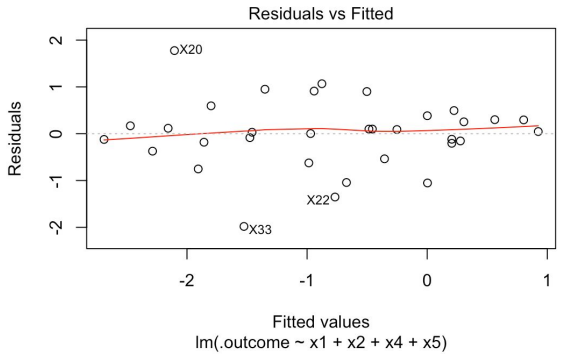
Subsets Regression Summary

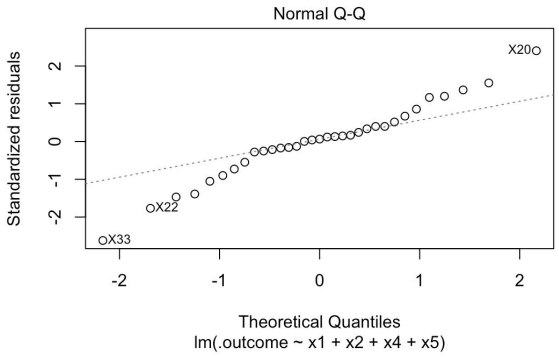
Model	R-Square	Adj. R-Square	Pred R-Square	C(p)	AIC	SBIC	SBC	MSEP	FPE	HSP	APC
1	0.3528	0.3319	0.273	17.2157	98.7347	3.7774	103.2242	34.1723	1.0981	0.0345	0.7307
2	0.4690	0.4337	0.368	10.9160	94.2025	-0.2590	100.1885	29.0022	0.9576	0.0303	0.6371
3	0.5368	0.4889	0.432	8.0760	91.6959	-1.9725	99.1784	26.2036	0.8881	0.0283	0.5909
4	0.6498	0.5998	0.54	2.0049	84.4646	-6.1924	93.4437	20.5432	0.7142	0.0230	0.4752
5	0.6605	0.5977	0.511	3.2422	85.4424	-4.1334	95.9179	20.6827	0.7370	0.0240	0.4904
6	0.6758	0.6010	0.526	4.1498	85.9208	-2.1343	97.8928	20.5407	0.7495	0.0247	0.4987
7	0.6840	0.5956	0.509	5.5642	87.0752	0.3974	100.5437	20.8552	0.7788	0.0261	0.5182
8	0.6906	0.5874	0.493	7.0957	88.3828	3.1289	103.3479	21.3102	0.8138	0.0278	0.5415
9	0.6916	0.5709	0.458	9.0240	90.2756	6.0962	106.7372	22.2065	0.8666	0.0302	0.5766
10	0.6919	0.5519	0.391	11.0000	92.2396	9.0897	110.1977	23.2386	0.9260	0.0331	0.6161

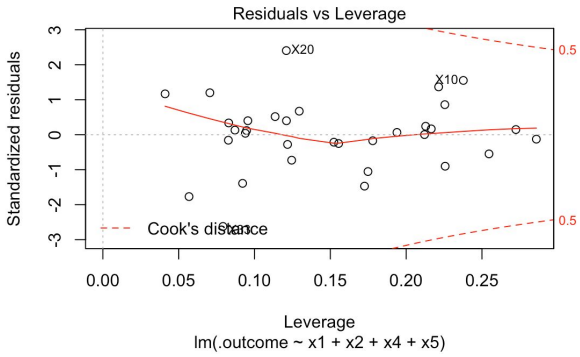
Backward model selection with AIC criterion

The full model is now simplified via the stepwise selection, with a backward direction and the AIC as the selection criterion. As expected, the selected formula contains 4 explanatory variables, which are x1, x2, x4 and x5, and the intercept.

While the model is sparser, the assumption of heteroscedasticity is not met, and the residuals are not normally distributed.







PCA model

The principal component analysis combined with cross-validation finds that the optimal number of components is 2.

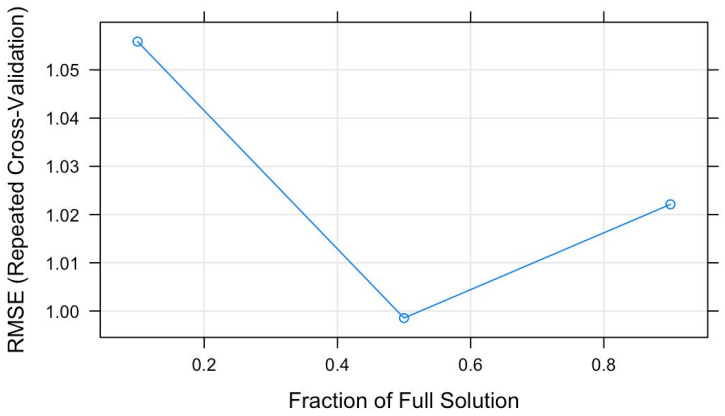
```
##   ncomp     RMSE  Rsquared       MAE    RMSESD RsquaredSD      MAESD
## 1     1 1.047565 0.3811637 0.9040539 0.2732099  0.2092049 0.2500636
## 2     2 1.003939 0.4375179 0.8329035 0.2277832  0.2201584 0.1905085
## 3     3 1.073363 0.3838161 0.8972427 0.1905332  0.2194583 0.1529403
```

Ridge and Lasso models

Optimal hyperparameters for Ridge regression are k=6, λ=0.1, as shown in the graph and plot below.

The optimal hyperparameter for Lasso regression is .5, as shown in the graph and plot below.

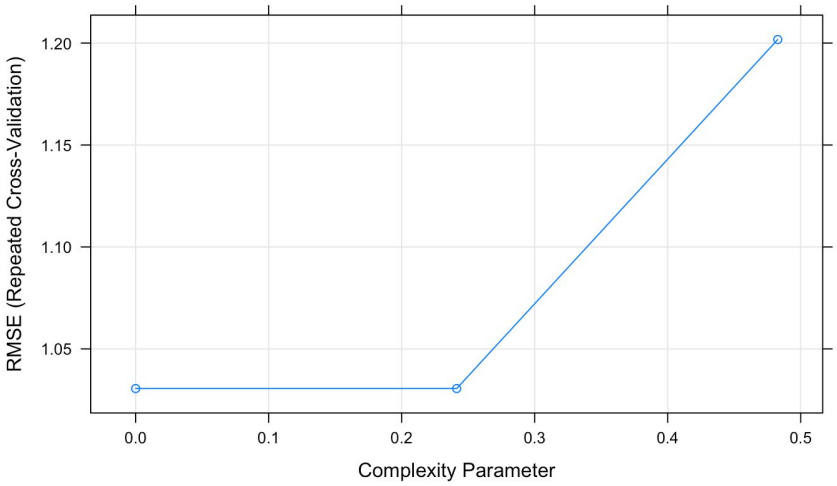
##	fraction	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
## 1	0.1	1.0558714	0.4294256	0.9386967	0.1743267	0.2635928	0.1533869
## 2	0.5	0.9985391	0.4831560	0.8012657	0.2795939	0.2341670	0.2516732
## 3	0.9	1.0221151	0.5066826	0.8327844	0.3553333	0.2416115	0.3310858



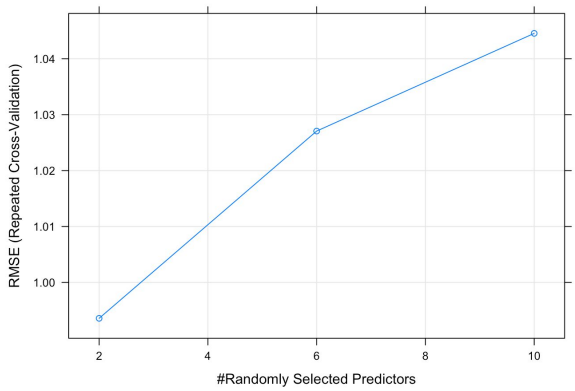
Regression Tree and Random Forest models

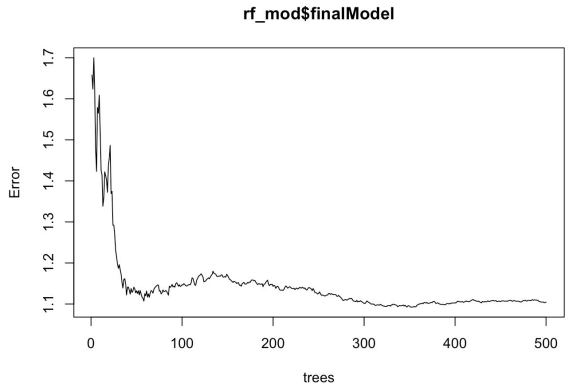
The best parameter for the regression tree with rpart is cp=0 or cp=0.2413643 (they achieve the same exact model).

```
##          cp     RMSE   Rsquared        MAE     RMSESD RsquaredSD      MAESD
## 1 0.0000000 1.030561 0.4040489 0.8838706 0.2515030  0.2430567 0.2194092
## 2 0.2413643 1.030561 0.4040489 0.8838706 0.2515030  0.2430567 0.2194092
## 3 0.4827286 1.201724 0.2663170 1.0485500 0.1775215  0.1314081 0.1747211
```



The best parameter for the Random Forest regression is 2. The model also uses 500 trees.





Comparison and Conclusion

Combining together the previous metrics, we get the following table:

The models can be split into two categories:

The first 3/4 models all have a low RMSE, high Rsquared, but more variance across folds compared to the last models. Among them, the Ridge regression model trumps the others. Note that the ridge model uses the 6 following variables: x9, x1, x2, x7 x4, x5 and x3.

The last models have a higher RMSE, a lower R squared, but present the advantage of being more stable across folds. Among them, the Random Forest model appears best.

Depending on the context and goal, one will choose one method or the other.

Given the small number of observations, the recommended next step if possible would be to gather more data and re-run the analysis.

Real Estate Investment

Introduction

In the context of a real estate investment, can we accurately predict the price of a house given its features? This question leads to another: what are the characteristics that are the most significant to understand real estate price?

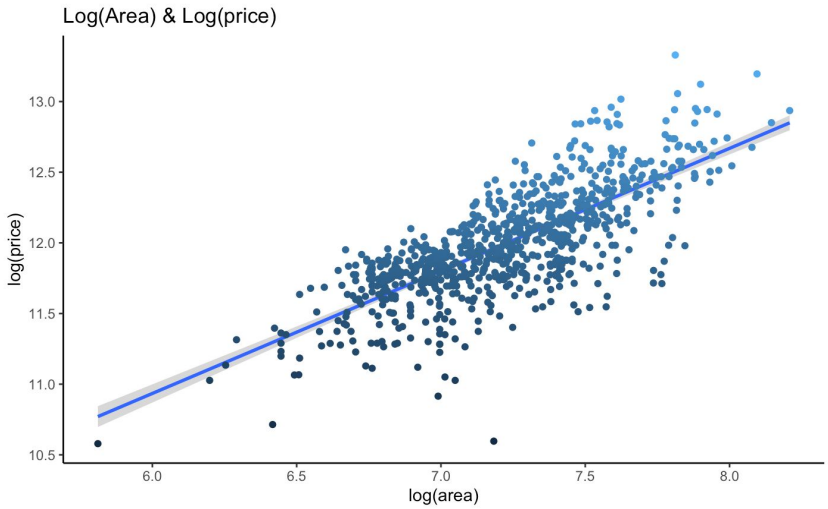
To answer this question, I'll use the real estate data from Ames, USA. I accessed the data via Duke University, but it is also available on Kaggle.com. The data I'm using is already split between a training and a test set, and a validation set has also been set aside.

The train set contains 1,000 observations, 81 features. There are missing values, distributed unevenly across variables.

Exploratory Data Analysis

The exploratory data analysis shows interesting relationships between the response variable, the price of a house, and most of the variables available in the dataset. The following three particularly stood out, and should thus be included in the initial modeling:

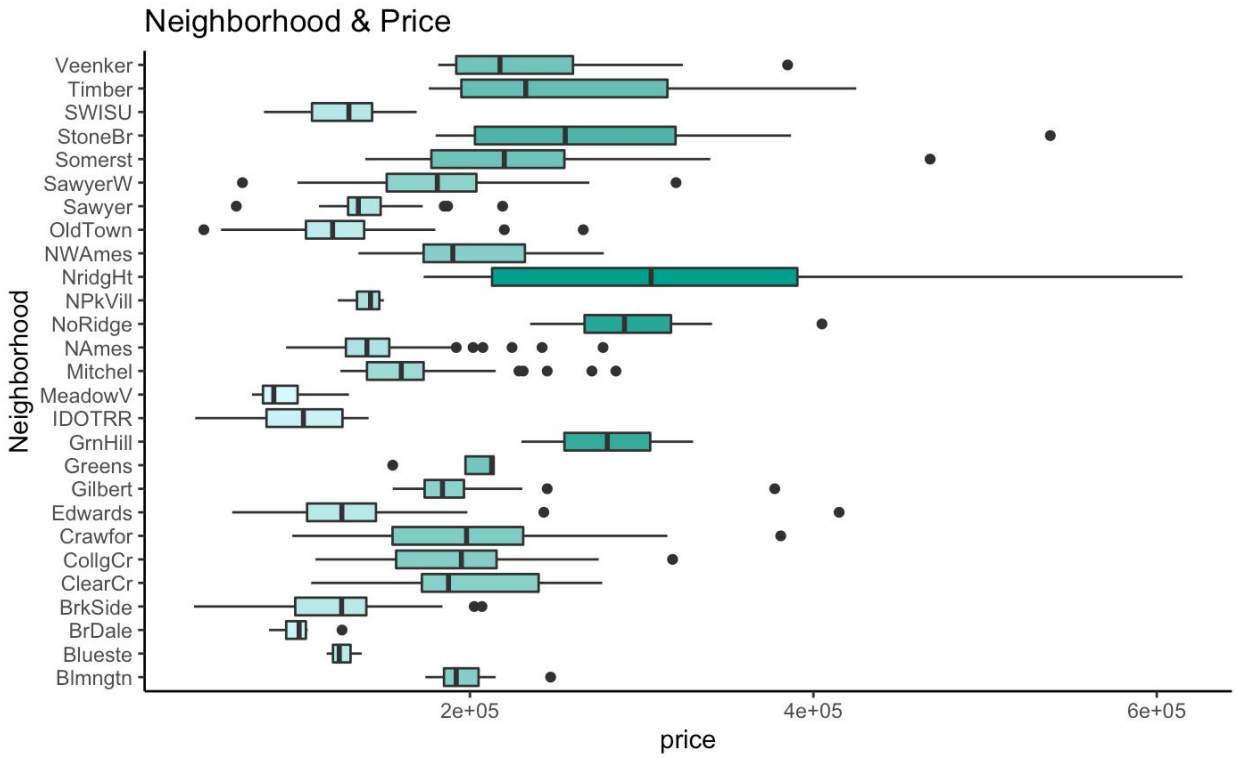
- the relationship between the size of a house and its price, where both are transformed using natural logarithm transformation. The plot shows a strong positive linear relationship between log(area) and log(price), suggesting that on average a bigger house will likely sell at a higher price.



- the relationship between the overall quality of the house and its price. The side-by-side boxplots show that on average houses of better quality sell at a higher price than houses of lesser quality: each category has a higher median than the category below, and in most

cases, a first quartile higher than the median of the category below. We note that the spread, measured using the Inter Quartile Range (IQR) tend, on average, to increase with quality. There are a few outliers, between the categories 4 to 8, mostly on the higher side. Finally, we note that three categories (1,2 & 10) have 5 or less observations, which can be problematic.

- the relationship between neighborhoods and house prices. The side-by-side boxplots display clear differences between house prices per neighborhood, both in terms of average value (median) and spread (IQR). There are outliers for most neighborhoods, mostly on the higher side. The boxplots suggest that on average different neighborhoods will lead to different house prices. We note that five neighborhoods have less than 5 observations, which is problematic.



Initial Model

We start the process of building a model by creating a simple, intuitive initial model based on the results of the exploratory data analysis. Based on the EDA, I selected 10 predictor variables to create a linear model for log(price) using those variables. The selected numerical variables all have a high correlation with log(price) , while the categorical features have important log(price) differences throughout their levels.

```
Call:
lm(formula = log_price ~ log_area + log_Lot.Area + Overall.Qual +
    Neighborhood + Bedroom.AbvGr + Year.Built + Garage.Cars +
    Garage.Qual + X1st.Flr.SF + Bsmt.Qual, data = ames_train)
```

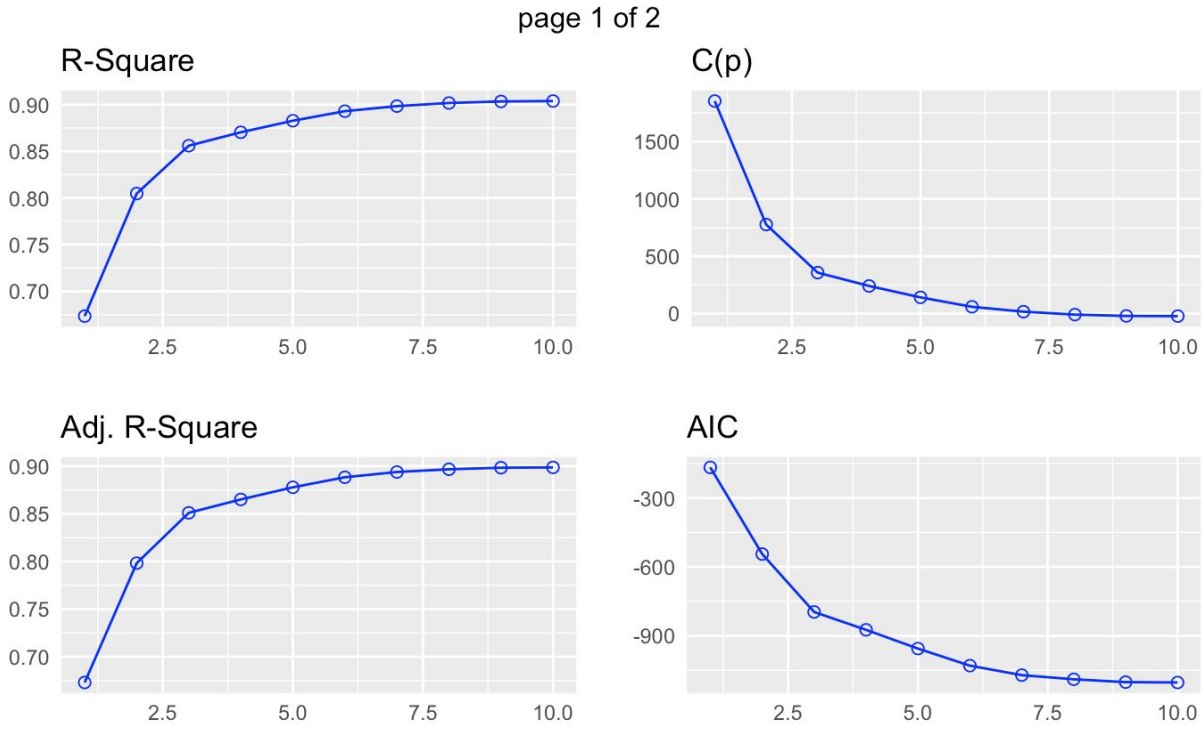
The summary shows that certain variables, with a high t value and low p-value, appear to be important predictors: log(area) for instance. Given its coefficient of 0.379, holding constant all other variables in the model, on average, an increase of 1 in log(area) will increase the log price by .379% percent. Similarly, all other variables in the model held constant, an increase of 1 unit in Overall quality will increase the log(price) by 0.081%. In less analytical terms, we're seeing that a bigger house, or a house of better quality, is likely to be sold at a higher price, on average, and when all other variables are held constant.

On the other hand, for the variable referring to basement quality, the negative coefficient linked to no basement indicates that on average, all other variables in the model held constant, a house without a basement will sell at a lower price, and more precisely the log of the price will be approximately and on average 33% lower.

Note that this first model already achieves a high R squared of 89.9%.

Variable selection

Using the convenient function from the OLSRR package, we can get an idea of the best model depending on various criteria (AIC, adjusted R2, etc.)



The MSEP, the adjusted Rsquared and the AIC all agree for the top 3 models (highest adjusted R2, lowest MSEP and AIC), which are ordered according to the number of explanatory variables. It confirms that our initial model with 10 variables is good. Given the small differences in performance between the model with 8 predictors and the one with 10, and our preference for sparser models, I'll retain the former as the new initial model, with the following explanatory variables:

- log(area)
- log(Lot.Area)
- Overall.Qual

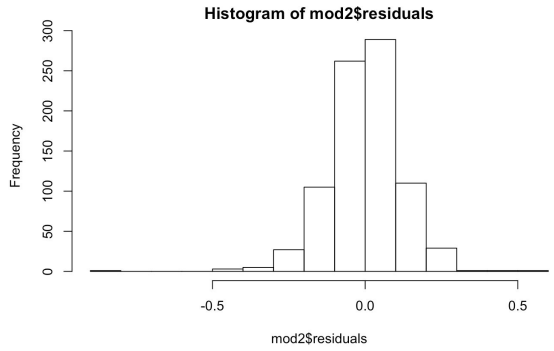
- Neighborhood
- Year.Built
- Garage.Qual
- X1st.Flr.SF
- Bsmt.Qual

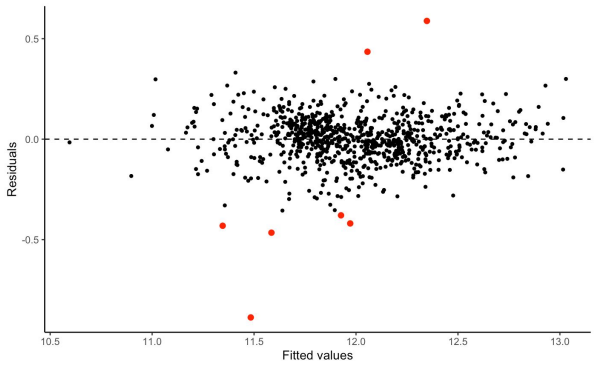
Residuals

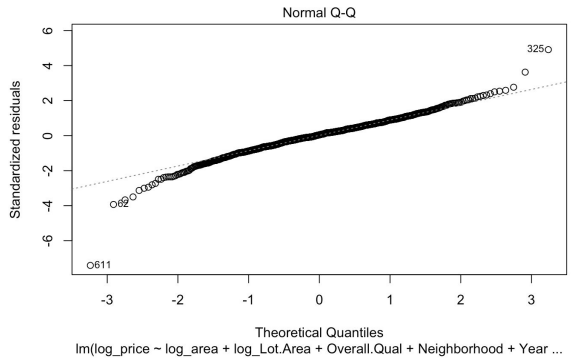
To check whether the model can be considered valid, we look at the residuals.

The plot residuals vs. fitted values shows that the condition of constant variability of residuals is met. It is also clear that residuals form a random scatter centered at 0. However, we can see outliers and points with significant leverage. Looking at the 10 observations with the highest absolute value residuals, all were built before 1956, when approximately 25% of our training data is in that category. There is thus more uncertainty in price for older houses.

The last condition that needs to be met for multiple linear regression is the independence of residuals, deriving from the independence of observations. The observations include all (unique) houses sold in Ames between 2006 and 2010. As a consequence, I'm not certain we can assume the independence of observations, and thus residuals.







RMSE

The initial model training RMSE is $21,902.36, while its test RMSE is $22,488.52.

The RMSE for the test data is higher than for the training data which is to be expected. However, the difference is reasonable and shows that there is no strong overfitting.

Final Model

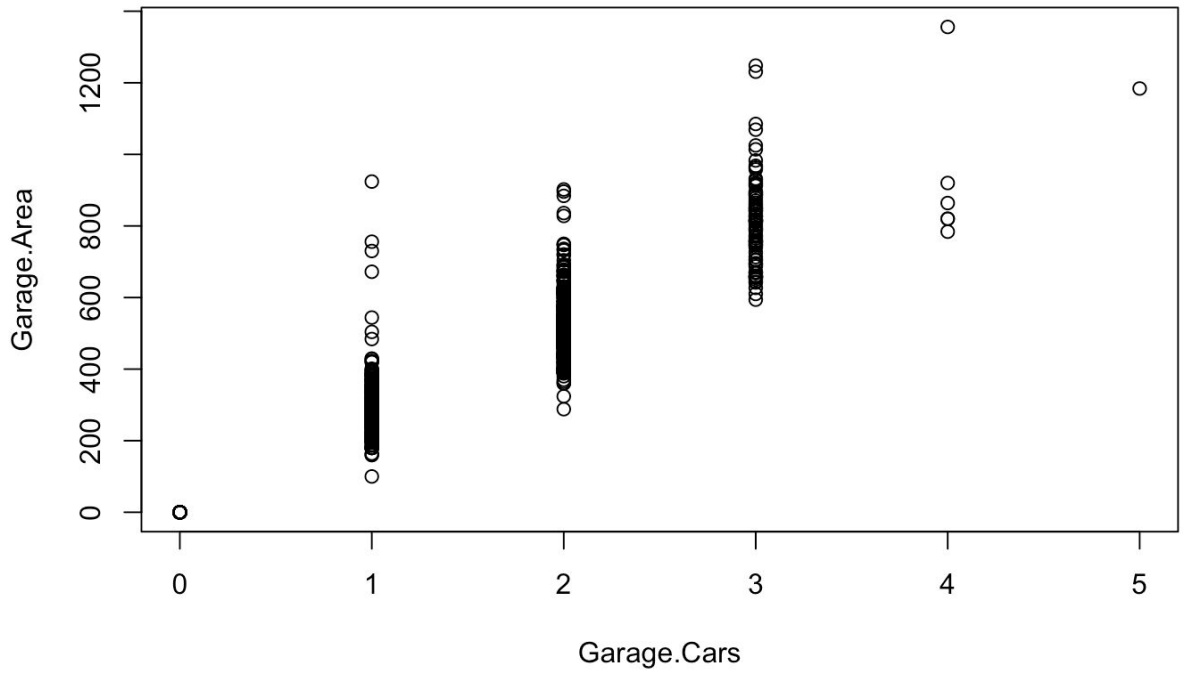
Below is the summary of the final model, with an adjusted R squared of .9229, an improvement compared to the initial model.

Transformations

I log transformed the numerical variables price (response variable), area and Lot.Area, and Total.Bsmt.SF as they were all (right) skewed. I had to add 1 to the latter before performing the log transformation, as it had values equal to 0. You can see the skew in the below histograms.

Variable Interactions

I included one variable interaction relative to the garage, where I multiplied the Garage.Cars and the Garage.Area variables. I decided to include it as I found an interaction effect between the two variables and adding the interaction term significantly improved the model. Initially, I included another one, Bath, relative to the number of bathrooms, but the AIC stepwise selection method recommended to do without (see next section).



Variable Selection

I used the results of the EDA to select the 20 variables for a ‘big’ model, then performed stepwise AIC selection, as it rewards 'goodness of fit' and penalizes model complexity. Note that the model selected also has the highest R squared and adjusted R squared. I further simplified my model after testing out-of-sample data and seeing some overfitting (see next section).

Best Subsets Regression

Model Index	Predictors
1	Overall.Qual
2	log(area) Neighborhood
3	log(area) Overall.Qual Neighborhood
4	log(area) BsmtFin.Type.1 Overall.Qual Neighborhood
5	log(area) log(Lot.Area) BsmtFin.Type.1 Overall.Qual Neighborhood
6	log(area) log(Lot.Area) BsmtFin.Type.1 Overall.Qual Neighborhood Year.Built
7	log(area) log(Lot.Area) BsmtFin.Type.1 Overall.Qual Neighborhood Year.Built Year.Remod.Add
8	log(area) log(Lot.Area) BsmtFin.Type.1 log(Total.Bsmt.SF + 1) Overall.Qual Neighborhood Year.Remod.Add Central.Air
9	log(area) log(Lot.Area) BsmtFin.Type.1 log(Total.Bsmt.SF + 1) Overall.Qual Neighborhood Year.Remod.Add Garage Central.Air
10	log(area) log(Lot.Area) BsmtFin.Type.1 log(Total.Bsmt.SF + 1) Overall.Qual Neighborhood Year.Built Year.Remod.Add Exter.Qual Central.Air
11	log(area) log(Lot.Area) BsmtFin.Type.1 log(Total.Bsmt.SF + 1) Overall.Qual Neighborhood Year.Built Year.Remod.Add Garage Exter.Qual Central.Air

Subsets Regression Summary

Model	R-Square	Adj. R-Square	Pred R-Square	C(p)	AIC	SBIC	SBC	MSEP	FPE	HSP	APC
1	0.6734	0.6730	0.6718	2699.8477	-167.1441	-2538.1550	-152.9654	39.7715	0.0478	1e-04	0.3282
2	0.8047	0.7982	0.7904	1282.3693	-544.1468	-2965.2967	-407.0860	23.8066	0.0295	0.0000	0.1967
3	0.8560	0.8510	0.8439	730.3990	-796.0899	-3217.0234	-654.3029	17.5786	0.0218	0.0000	0.1454
4	0.8806	0.8755	0.8689	466.6642	-940.2296	-3370.8077	-770.0852	14.5949	0.0183	0.0000	0.1209
5	0.8938	0.8891	0.8834	325.8382	-1036.0383	-3466.2583	-861.1676	12.9955	0.0163	0.0000	0.1077
6	0.9030	0.8987	0.8931	227.8652	-1110.0369	-3539.7670	-930.4400	11.8780	0.0149	0.0000	0.0986
7	0.9100	0.9058	0.9005	154.6698	-1170.1411	-3599.2523	-985.8180	11.0390	0.0139	0.0000	0.0917
8	0.9172	0.9132	0.9079	78.9946	-1237.5525	-3665.5779	-1048.5031	10.1697	0.0128	0.0000	0.0846
9	0.9212	0.9174	0.912	37.1574	-1277.4365	-3704.6299	-1083.6609	9.6834	0.0122	0.0000	0.0807
10	0.9244	0.9204	0.9147	5.3559	-1305.2322	-3735.5745	-1097.2779	9.3101	0.0118	0.0000	0.0776
11	0.9269	0.9229	0.9171	-20.0000	-1331.6229	-3761.0819	-1118.9424	9.0095	0.0114	0.0000	0.0752

Testing

Testing out-of-sample data using the RMSE as an indicator of model fit showed that there was some overfitting on the training data when using the AIC model. The RMSE of the training data was more than 10% lower than that of the test data. As it was significant, I tried a variety of adjustments to the model (mostly removing independent variables that were not as relevant and thus simplifying the model) and checked the impact of each on overfitting and the adjusted R squared. I opted for a simpler model with a slightly lower adjusted R squared (.9229 vs. .9275) than AIC model, but that led to less overfitting. Looking at the coverage probabilities of the test data, we find that the proportion of out-of-sample prices that fall in the 95% prediction interval are slightly less than .95, suggesting that the final model reflects uncertainty relatively well.

Final Model Assessment

The residual plot below of residuals vs. fitted data, shows that residuals are randomly scattered around 0, that the variability is constant, and that overall the final model is a good fit for the data (which is confirmed by the high adjusted R squared). However, as with the initial model, there are a handful of outliers, with positive and negative residuals.

While being relatively parsimonious (a dozen independent variables selected) the final model yields a high adjusted R squared of .92, and the residual plot shows that the conditions for multiple linear regression are met. Overall the model is a good fit for the data.

The RMSE of the validation data is lower than that of the test data, but still slightly higher (by about 7%) than the RMSE of the training data.

The coverage probability, which represents the percentage of 95% predictive confidence intervals that contain the true price of the house, is 93.97% for the validation data set. This is reasonably close to .95, and thus the final model reflects uncertainty pretty fairly.

Conclusion

Out of the 80+ variables available in this dataset, the final model uses only a dozen of them and is able to predict the price of a house, with a RMSE of approximately $21,000 and an adjusted R squared of .92. Some of the most influential features are the size of the house and its quality, while data on the number of rooms, bathrooms or kitchens proved to be less significant in comparison, and thus were left out of the model.

In a next step, the data could be supplemented with real estate information from various cities to generalize the model to other locations.