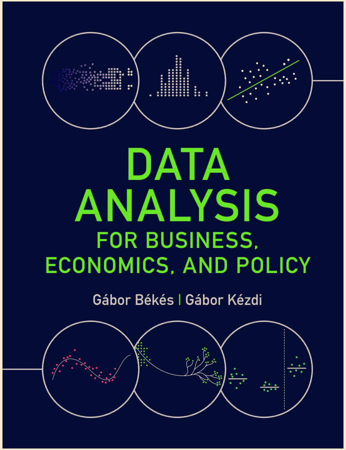
Békés-Kézdi: Data Analysis, Chapter 16: Random Forest and Boosting


DATA ANALYSIS
FOR BUSINESS, ECONOMICS, AND POLICY
Gábor Békés | Gábor Kézdi

Data Analysis for Business, Economics, and Policy

Gábor Békés (Central European University)
Gábor Kézdi (University of Michigan)

Cambridge University Press, 2021

gabors-data-analysis.com

Central European University

Any comments or suggestions:
gabors.da.contact@gmail.com

Bagging, Boostrap, Random Forest

Trees

- CART is great for
  - Capturing non-linear, complicated panels
  - Yielding a model that can be explained

- The problem with CART is it's poor prediction performance
  - Avoiding the overfitting is not working out greatly.

If one tree not good enough, have many trees instead

- CART problem: dependence on individual observations is high
  - Early decisions may depend on small differences between choices.
- Instead of a single tree, let us have many.
  - Create many similar datasets
  - Grow trees
  - Aggregate
  - **Bagging**= **B**ootstrap **agg**regation

Remember: Bootstrap process

- Start with original dataset and draw many repeated samples with replacement.
- The observations are drawn randomly one by one from the original dataset; once an observation is drawn it is "replaced" to the pool so that it can be drawn again, with the same probability as any other observation.
- The drawing stops when it reaches the size of the original dataset.

Remember: Bootstrap process

- One outcome dataset
  - Same sized sample to the original dataset.
  - Includes some of the original observations multiple times,
  - Does not include some of the other original observations. For observations included, each variable is kept as is.
- Result is many datasets
  - Many repeated samples that are different from each other.

Bagging process

- Instead of one tree, we'll grow many (K) trees
- Create K bootstrapped samples
  - Same sized, same properties yet actual differences
- Build K trees, and estimate
  - Grow a tree on each sample
    - With a pre-defined stopping rule instead of pruning
  - Output: set of decision rules.

- Assemble the K set of rules and estimate it on test sample
- Average out predicted values (K values for each observation)
- The average is the prediction.

Bagging process

- Assemble the K trees (=set of rules) and estimate it on **test** sample
- Average out predicted values (K values for each observation)
- The average is the prediction.
- Cross-validation: may do this five times
- Bagging = **B**ootstrap the sample - create many samples + **Agg**regation - average results
- Increase stability of results - better out-of-sample performance
- It may be used for prediction, but we'll add a tweak...

Random forest

- Bagging with a tweak – decorrelate trees
- Keep the idea of using bootstrapped samples
- BUT!
- Instead of allowing all variables to be used at any given mode...
- ...we randomly select m variables
  - m is about the sqrt of number of variables p.
  - $m = 4$ is often used as minimum
- At each node, we pick one variable out of m

- Yielding a set of decorrelated trees
- Helps reduce the risk of overfitting

Random forest vs bagging

- Why are we considering a random sample of m predictors instead of all p predictors for splitting?
  - If we have a very strong predictor in the data set along with a number of other moderately strong predictor,
  - –>in the collection of bagged trees, most or all of them will use the very strong predictor for the first split!
  - –> all bagged trees will look similar.
  - Hence all the predictions from the bagged trees will be highly correlated.
    - Averaging many highly correlated quantities does not lead to a large variance reduction.
  - Random forests "de-correlate" the bagged trees leading to more reduction in variance.

Random forest vs bagging

- Decorrelate by using fewer possible predictors
- Thus, artificially making each model worse...
- But in sum, we are making a better model...
  - ... in a slightly counter-intuitive way

Tuning parameter

- The advantage of random forest over other methods is that it needs relatively little tuning.
- Tuning = set of parameters
  - Often selected by CV
  - OLS has no such parameter – it is based on a formula
  - LASSO has λ, CART Pruning had α
- Other machine learning methods have typically more tuning parameters
  - Also called hyperparameters

Random Forest tuning

- T=Number of trees
  - T=500 as default
- *m* - the number of variables checked for a spit
  - typically the square root of number of variables.
  - Could be determined by cross-validation
- Depth of trees (size) = Minimum node size
  - Where tree building stops

The practice of prediction with random forest

- Random forest is an ensemble method combining the results of hundreds of regression trees
- The most important elements of the random forest are
  - bagging: aggregating the predictions of many trees grown on bootstrap samples of the data
  - each tree is grown to be large, using a simple stopping rule
  - decorrelating trees: when growing each tree, we use only a subset of the variables for each split
- In practice, carrying out prediction with a random forest is quite easy due to good software solutions with sensible default options

Case study: Airbnb London data

- Airbnb prices
- Whole of London,UK
- http://insideairbnb.com/
- 50K observations
- 94 variables, including many binaries for location and amenities
- Key variables: size, type, location, amenities
- Quantitative target: price (in USD)

Case study: Airbnb: From data to Random Forest

- Some tasks same as in regression
  - Data cleaning
  - Filtering on types we care about
  - Encoding information (here: amenities is text–> set of binaries)
- Some tasks are not needed
  - No functional form decision
  - No variable selection
- No interaction picking
- Some new tasks
  - Set tuning parameters (size of trees, how many variables to try out).
  - Can add an algo that tries out a bunch of combinations.

Case study: Airbnb London data – running the algo

- From cleaned data
- Just run the algo
  - With minimal tuning
- That is it.
- Output
  - Large pool of decision rules: T=500 set of trees
  - Nothing to interpret
  - Black box model
- RMSE is 44.5 vs 48.1 (OLS)

Looking into the black box

Black box model

- Random Forest (and other ML models) are often called "Black Box" models.
- They make a prediction, but in lack of formula, we do not really know how an actual prediction is created

- Business - when could this be a problem?

Bagging / Random Forest - a drawback

- One drawback of the process is we no longer have a nice tree, which we could interpret.
- We have, instead K trees,
- The average is the predicted value.

- It is now hard to interpret the model!
- We can always pick a single tree to look at.

- Look at variable importance - a measure of how useful a variable is for prediction
- Add a new way to look at partial correlations

The variable importance plot

- How do we decide which variables are most useful in predicting the response?

- Variable importance plot
- For each variable it captures the overall contribution to reducing RMSE
- Shows relative importance

- Calculated for each tree and averaged over all trees.

Partial dependence plot

- Look at the relationship between predicted values and predictors
- For each value of a predictor, we can look at predicted values: Partial dependence plot (PDP)
- The PDP is a graph
  - values of the x variable on the horizontal axis
  - the values of average y on the vertical axis.
- Shows how average y differs for different values of x_i when all the other x values are the same.
- The "partial" = differences with respect to this x_i variable, *conditional* on all other x variables
  - differences attributed to them are "partialled out"
  - Sounds causal: it is, mechanically. (Not in real world.)

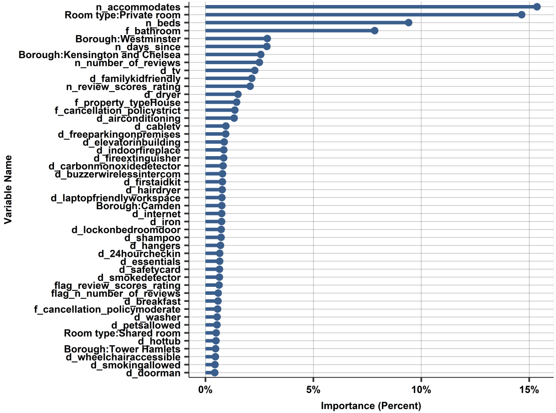
Partial dependence plot

- Graphical visualizations of the marginal effect
- The partial dependence function tells us how the value of the variable x_i influences the model predictions
  - after we have “averaged out” the influence of all other variables.
    - For linear regression models, the PDP plots is a straight line whose slopes are equal to the model’s beta parameter.
- Can create a plot with one or two variables

Partial dependence plot: problems

- If there are important interactions (as it is likely to be the case), these may be misleading.
  - Unlike in linear model, we do not explicitly take this into account.
- Problem If there are variables strongly correlated with x_i.
  - Calculation of partial effect could be misleading

Case study: Airbnb London data –VarImpPlot 1

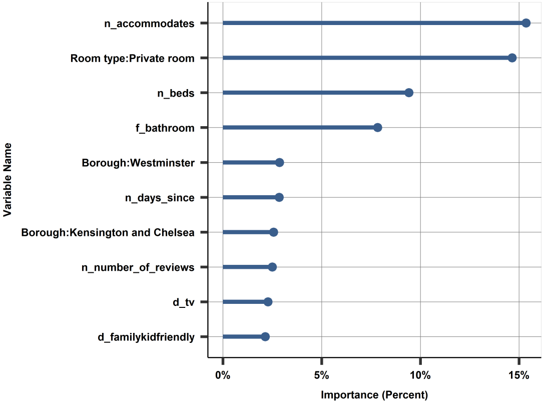


- Random Forest Variable Importance Plot
    - Normalized by total improvement.
    - All variables above cutoff.
- Too hard to read....

Variable importance plot - a nuisance in Caret

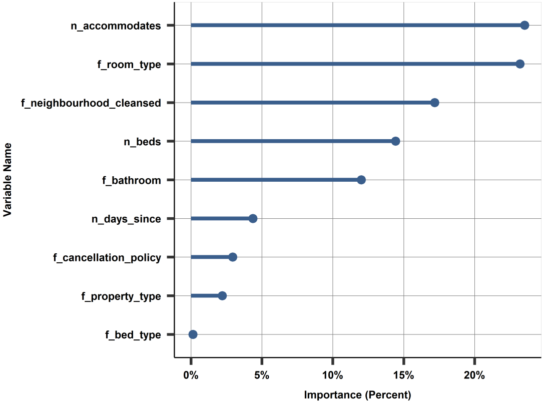
- Plot has small values for variables that are not in the tree shown.
- Caret uses the top competing variables, which are not chosen, are also tabulated at each split.
  - Not the default setting in python

Case study: Airbnb London data –VarImpPlot 2



- Random Forest Variable Importance Plot
  - Normalized by total improvement.
  - Top 10 variables - easier to read

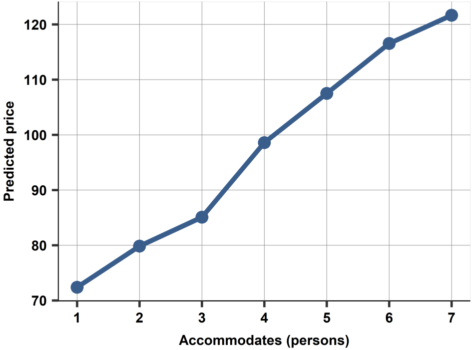
Case study: Airbnb London data –VarImpPlot 3



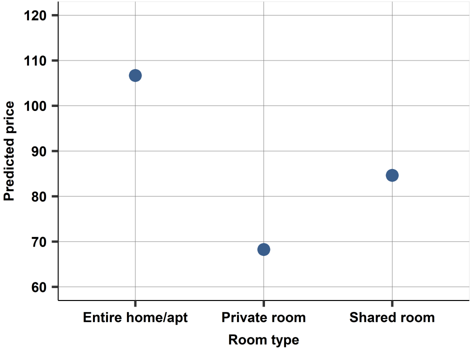
- Random Forest Variable Importance Plot
  - Normalized by total improvement.
  - Top 10 variables – With grouped factors
  - Grouping binary created for a factor as one variable

Case study: Airbnb London data - PDP plots

Numeric variable: looks linear



Categorical variable: there is variation



Performance across subsamples

	RMSE	Mean price	RMSE/price
Apartment size			
Large apt	62.11	144.6	0.43
Small apt	28.53	62.3	0.46
Type			
Apartment	42.32	92.8	0.46
House	42.47	76.3	0.56
Borough			
Kensington and Chelsea	65.11	146.3	0.45
Westminster	62.39	131.0	0.48
Camden	50.23	108.5	0.46
Hackney	33.99	78.2	0.43
Tower Hamlets	34.29	72.0	0.48
Newham	31.94	63.3	0.50
All	42.36	88.8	0.48

Boosting with GBM

Boosting

- Boosting is another ensemble method based on trees
- Different tree building and aggregation algorithm

Boosting

- Boosting is an alternative ensemble method.
- What's new?
- Bagging / random forest
  - grows independent trees,
- Boosting
  - Grows trees that build on each other.
  - Then, similarly to bagging, it combines all those trees to make a prediction.

Boosting idea (1)

- Boosting – grow trees *sequentially*, using information from the previous tree to grow a better tree the next time.
- The information used from the previous tree is which observations were harder to predict.
- The new tree then puts more emphasis on fitting those observations.
- Typically, this is done by taking the residuals from the previous prediction and fitting a model on those residuals instead of the original target variable.

Boosting idea (2)

- The prediction after having grown this next tree is not from the new tree only, but a combination of the new tree and the previous tree.
- Then, in the following step, the algorithm grows a yet newer tree building on that combined prediction, taking its residuals, and so on.
- The algorithm stops according to a stopping rule, such as the total number of trees grown.

Boosting idea (3)

- The final ingredient in boosting is aggregation: Take all previous trees as well to make the final prediction.
  - Rather then using the best tree built at the end
- –>Ensemble methods.
- Instead of using the results from one (maybe the best) tree,
- Combine results from many trees even if they are known to be imperfect.
- New vs RF
  - Trees gradually built
  - don't want those many trees to be independent of each other.

Gradient boosting machine (GBM)

- One boosting algorithm is Gradient boosting machine - GBM
- The **gradient** part of its name refers to a search algorithm that it uses to finds a better fit.
- At every step, the new model doesn't differ very much from the previous one.
- GBM has more tuning parameters than random forest.
    - determine the complexity of trees,
    - the number of trees,
    - how we combine the trees to form the new prediction at each step,
    - how large each tree should be.

Gradient boosting machine (GBM)

- Many boosting libraries
- We use Gradient Boosting Machines
- Alternatives: xgboost and many more

Case study: Airbnb London data - running the algo

Model	**RMSE**
Linear regression (OLS)	48.1
Linear regression (LASSO)	46.8
Regression Tree (CART)	50.4
Random forest (basic tuning)	44.5
Random forest (autotuned)	44.7
GBM (basic tuning)	44.6
GBM (broad tuning)	**44.4**

Machine learning in practice

- There are many other methods
- With similar idea
  - Would like to try out many models
  - Can't try out all
  - So have a smart shortcut
  - Try out many
  - Avoid overfitting

Why use random forest?

- There are many other ML method, but in our view, Random Forest and Boosting are great.
- It is based on a classic statistical approach with well known features
  - It is useful to know regression trees, sometimes they are really illustrative
- RF is based on a very important idea in machine learning (bootstrap aggregation)

Most importantly:

- For cases when target is number
- RF/GBM perform the best among key methods, or very close to the best.

Key advantages of ML

1. The most important advantage is that in terms of prediction, it performs better than regressions.
    - Some cases this is marginal, in other cases it is substantial.
2. Another advantage is the easy use
    - Once you have the features
    - Random Forest is easy to use
    - GBM is a bit harder but still easy to use
    - Get very good predictions right away
    - Easy to make the process automatic

Machine learning applications - a review

A key problem with machine learning

- It is a black box...
- Interpretations are difficult

- Can't do analysis like
- What would happen if there was a tax on dogs/cats

Some additional comments

- Random forest implementations are fairly fast, but much-much slower than a single tree / regression
- Fast implementation with h2o.ai
  - R, Python has super easy API for this, integrate seamlessly into code
  - With larger dataset, this is a good solution
- Advice for faster machine learning projects
  - use random subsample of data to keep calculations fast interactive
  - have a simple baseline (OLS, logit, CART)
  - Don't spend much time with fine tuning (tune hyper-parameters).

Outside validity and causality

- The role of causality in prediction
- Underestimated in data science
  - Maybe over-estimated among economists...
- Models with a theory are basically correlations.
- If you have no idea what is behind a correlation, you have no idea what might cause that correlation to break down.
- Having a structure (theory) in mind, may make you add variables despite poor fit, because in outside data it may matter.

Linear regression vs ML: Some trade-offs

- Prediction accuracy versus interpretability.
  - Linear models are easy to interpret;
  - Splines, polynomials are hard;
  - ML models are impossible.
- Parsimony versus black-box.
  - a simpler model involving fewer variables over
  - a black-box predictor involving many may perform better, but harder to operate, understand, interpret.

Predicting a quantitative target variable - overview

- Target variable y is numeric
- Predictors were born as text, number, categorical variable –> transformed to numbers
- Designed sample
- Feature, target engineering
- Cross-validated model selection

- Linear regression (OLS, LASSO)
- CART
- Random Forest
- Boosting (GBM)

Summary of methods

	OLS	**LASSO**	**CART**	**RF**	**GBM**
Performance (RMSE)	48.1	46.8	50.4	44.5	44.4
Speed (in min)	0.07	0.3	0.4	19	756
Solution	closed form	algo	algo	algo	algo
Choice of tuning parameters	n.a.	easy	easy	easy	hard
Interpretation	easy	easy	easy	difficult	difficult
FE: Variable selection	hand	algo	algo	algo	algo
FE: Non-linear patterns	hand	hand	algo	algo	algo
FE: Interactions	hand	hand	algo	algo	algo

Big Data

- What's different if dataset is very big
  1. Sheer size of the data require powerful new tools.
  2. Have more potential predictors than appropriate for estimation – need variable selection.
  3. Large datasets may allow for more flexible relationships than simple linear models.
- Machine learning techniques may allow for more effective ways to model complex relationships.
- Covered: Decision trees, random forest
  - Not covered: boosted trees, support vector machines, neural nets, deep learning

Prediction and big data: external validity!

- Overfitting aspect could be different
- In some cases you have
  - The whole dataset (ie population)
  - Very large random subsample of the population
- Overfitting may be less of a concern if you have very large data
- External validity concern remains!
  - No matter the size of the sample!
  - Must always think about potential differences between data at hand and data that is used at prediction

ML = Hype + Promise

- Machine learning is basically curve fitting
  - Often great to find patterns
- Regressions still useful

- External validity is not solved by large data and powerful methods
- Causality is relevant
- ML and causal tools may be combined

Main takeaways

- Random forest is a prediction method that uses several regression trees
  - Ensemble methods that combine predictions from many imperfect models can produce very good predictions
  - Random forest is the most widely used ensemble method based on regression trees; boosting is a more complicated but often better alternative
  - Both random forest and boosting are black box methods. We need to do additional diagnostics to uncover how the x variables contribute to our prediction.