RF Research

Fabio Rougier

March 30, 2022

Contents

1 Master thesis from Eindhoven

This (Kuznetsova et al., 2014) is a master's thesis

1.1 Remarks

The author uses an entire chapter to describe in detail how a random forest works. It's definitely not a good thesis, judging by the wording. However, it might hold some value from its content...

1.2 Points to note

- Prototypes: On p 11 the author mentions the idea of prototypes for every class and attribute
- Different consumers: On p 14 the author emphasizes the different approach to ML, data and its different consumers. While the ML expert is more focused on improving his model performance (and it's indicating variables.), the Analyst/Domain Expert is more concerned about the insights that the model yields (attributes, instances, noise,).
- RAFT: Random Forest Tool by Breimann is introduced as an existing tool for RF visualisation, authors conclusion is that tool is best with small amount and exclusively numerical attributes
- Small multiples: Mentioned in her source n16 (Tufte, 1985)
- Visualization considerations: p20/21 lists concise considerations for the Visualization of the RF following the principle of small multiples
- ReFINE: Tool developed by the author, looks promising

1.3 Summary

Overall, a mediocre thesis, but all the more interesting and extensive software and approach. Note that it is from 2014 which is likely why the author uses java.

2 Breimann Implementation Paper

This (Livingston, 2005) is a paper on a particular implementation on the RAFT software of Breimann.

2.1 Remarks

The paper seems to be concerned about the specifics of how a RF is created. Using RFs for problems where a very rare occasion has to be trained often results in bad models, because the training set is already heavily scewed towards the "regular" case and so the detection of edge cases seems to be a problem for such models. They do use the RAFT software but the paper is likely not focusing on the specifics of RF Visualization. Additionally the paper is from 2005, so it is quite outdated. They use Fortran and Java ...

2.2 Points to note

- Weka: Some kind of java program
- Variable Importance: The authors implemented variable importance into Weka

2.3 Summary

Very old paper and the specifics will unlikely be relevant today, however it might contain very important references that I can use!

3 Explainable Matrix paper

This (Neto & Paulovich, 2020) is a journal article presenting ExMatrix - a visualization method trying to convey the configuration of a model in a matrix structure.

3.1 Remarks

This paper is from 2021 and therefore far more relevant than the others. It also specifically focuses on the problematics of RFs.

3.2 Points to note

- Model interpretability: The main problem statement is the lack of interpretability of models and their decisions despite being accurate. A 99% accuracy does not convince anyone, if there is no explanation for the decision.

- Global/Local approaches: Global explains the entire model, trying to improve the trust in the models' decision making. Local explains the decision behind a single instance.

- pre-/in-/post-model strategies: For which stage of the ML process is the visualization helpful?

- BaobabView: This is a node-link explanation technique (Van Den Elzen & Van Wijk, 2011), down below. But node-link visualization has scalability issues.

- RuleMatrix: The technique has been used before (Ming, Qu, & Bertini, 2018).

- Decision Paths: Focus of the visualization is on decision paths, rather than nodes.

- Surrogates: Sometimes RFs are used as a *surrogate* for a less interpretable model.

- iForest: Also closely related to ExMatrix (Zhao, Wu, Lee, & Cui, 2018), summarizes decision paths.

Trying to explain the visualization specifics, because I think, this is an incredibly valuable paper for me: Every path is translated into a rule vector, consisting of as many coordinates, as there are features along the path. Each rule consists of *predicates* which represent single decisions on attribute values. For each rule vector there is a rule certainty (for each class, adding up to 100%), which represents how accurately the respective path classifies the instances that came along its path. The rule vector class is the class that has the highest value in the rule certainty. Rule coverage is the percentage of instances of the training data of class c for which the rule *works*. The 2nd graphic is the LE (Local Explanation) / UR (Used Rules) visualization and focuses on a single instance. Each features decision boundaries are layed out, with a pointed line, representing the given instance. There is an additional column *cumulative voting* next to the rule certainty, which sums the certainties up to the respective row, starting from the top. The 3rd graphic is the LE / SC (smallest changes) visualization. It visualizes the smallest necessary rule change to change the classification of the displayed sample.

3.3 Summary

Extremely relevant paper with lots of potential references and ideas. This could even be a visualization to build my own visualization upon. The authors noted in their own conclusion, that especially the LE/SC visualization leaves quite some room for improvement. The general idea of breaking down the RF into single rules is also interesting. The code is on github and a forked version would be an option. However, the github was not updated since last year April. I could consider contacting the authors if I pursue any further ideas along their approach.

4 BaobabView

This (Van Den Elzen & Van Wijk, 2011) is a paper about the visualization of decision trees with the focus on giving domain experts the ability to bring in their specific knowledge.

4.1 Remarks

Since this is mostly about decision trees, it might not be very applicable to RFs. It is also quite old already (2011).

4.2 Points to note

- User requirements Users of the visualization usually only want 3 different things: EDIT the tree (grow, prune or optimize) USE it (classification) or ANALYSE it (data exploration)

- Node-Link diagram: Visualization uses a Node-Link diagram

- Streamgraphs: Figure 4, might be interesting for Node visualization

- Confusion Matrix: Should be considered to inspect misclassifications

4.3 Summary

"Different activities require different visualizations"(in 6, Conclusion) is a simple, but important lesson from this paper.

5 Matrix Visualization

This (Ming et al., 2018) is an entire paper on the idea of a rule matrix. This is used in the aforementioned "explainable Matrix" paper.

5.1 Remarks

There will likely be little that I can draw from this in regards to visualization of RFs, but it might still be a valuable source for citation of basic concepts if I do go further down along the explainable matrix approach. Since the paper was written in 2019 it is also one of the newer concepts.

5.2 Points to note

- Not specific to RFs: The concept of turning a ML model into a rule matrix works with all kinds of ML models, not just RFs

5.3 Summary

Mostly included this paper for completeness. It will only become relevant if I dive deeper into the ExMatrix topic.

6 iForest

This (Zhao et al., 2018) is a visualization approach mostly aiming for the interpretability of RFs.

6.1 Remarks

The version that I have seems to be the "unreleased" version, so be careful when citing this paper and look up the links at the footer.

6.2 Points to note

- Poor interpretability: Some domains would not use RFs simply because they are too difficult to interpret. Domains like healthcare or finance need the interpretability in order to be viable in day to day use.

- 3 different RF interpretation methods: feature analysis, model reduction and case.based reasoning

- Feature Analysis: Calculation of feature importance is easy and effective (Either Mean Decrease Accuracy ir Mean Decrease Impurity); However, MDI is specialised for tree-based models. Both can be calculated globally or for an individual feature.

- Partial Dependence Plots(PDP): Line chart where x-Axis is feature values and y-axis shows prediction probabilities. (Friedman, 2001), (Hastie, Tibshirani, Friedman, & Friedman, 2009) Usually good for trees, but not as good for forests. But since they included them anyways, there will probably follow a good reason for how and why they included them.

- Model Reduction: Mentions the idea of surrogates. Since they usually reduce too much and don't help with the actual understanding of the RF, they are not used here.

- Case-based Reasoning: Picking some kind of idealised sample in order to inspect the results of the RF based on the specific sample and user-configured permutations of that sample. This is connected to looking at distance measures between features and inspect correlations of distances and predictions.

- Design Goals: A section that I have seen in multiple papers now. Splitting up the different goals of the visualization and the necessary tasks to get there in G1,2,3... and T1,2,3...

- Implementation: This is one of the very few visualization where the implementation gets mentioned. Apparently they used Python (yay!). *Scikitlearn* is used for the entire RF part and *Flask* (Grinberg, 2018) for the Backend. The frontend is built using *D3* (Bostock, 2012).

6.3 Summary

Overall this visualization approach has better scalability than the matrix idea and the icicle representation from the master's thesis. The implementation approach of designing the entire visualization as a web-application also seems very promising.

7 Current Bottom Line

7.1 Technical stuff

The implementation from iForest seems very solid, but also a lot of work to set up initially. A web application would be a modern approach and an excellent showcase for any kind of visualization.

7.2 Target audience

Focusing on a specific target audience seems to be useful as this entire ordeal is a constant decision about trade-offs. Focusing on a target audience would reduce the amount of constant arbitrage between different kinds of graphics significantly.

7.3 Explicit goal

Tied in with the target audience are the goals that the visualization wants to achieve. They will be helpful to have an orientation during development but also give the thesis clearer path. Do I want DSs to precisely optimize their RF model or do I want a doctor to understand why the model chose a specific class for a sample?

7.4 Honest reflection on the necessity of the initial thesis statement

A significant amount of high quality work has been done on the topic of visualizing RFs. Is it really necessary to build something completely new from scratch or is there a smarter way to go about this? Maybe it is possible to improve on one of the existing approaches. This would reduce development time significantly (at least in theory, depending on how well documented the existing implementations are...) and enable more in-depth apporaches, that I would otherwise not be able to achieve if I was starting from nothing. In order to do that, I would need to formulate a strong argument for why it is necessary to improve the existing works and how this would be enough to warrant an entire thesis on it.

7.5 Discussion with Aleksandar

- Explainations: How much explanation of RFs/Decision Trees concepts and algorithms is enough/too much? —Depends on the rest of the content. Things that are a prerequisite should be explained. However, this will follow the development. Writing will happen at the end!

- Other ensemble methods: Should I only focus on RFs? —I can choose. Aleksandar's idea was to look at decision trees first and then scale it up. I can also ignore that and just focus on ensemble methods (which I think I will do...)

- Online vs. books: Is it considered a problem that most of the resources are journal articles instead of books? —No

- Research question: Is it still to be formulated or was there a clear idea present? —Not as important and it will likely change during the process.

- Target audience: Visualization needs will differ depending on who is supposed to use it. —I can decide, but it is well possible to alter that later on (with the right explanation).

- Evaluation: Is this an important part of the thesis or is a use case presentation sufficient? —The Heider Lab Team can do evaluation. It is likely necessary, but not a hugely important part.

- Prototypes: Is this a relevant concept? (Idealised samples) —Interesting idea, keep it in mind, but for later.

- Scope: Data preparation, Model creation, Data/Feature Analysis, Model performance... What to include? —I have to make a choice on this.

- *Bad* Sources: How to deal with sources that are obviously lackluster, but have important ideas? —Just use the ideas and don't focus too much on the rest of the source.

- Technical explanations: Many of the works do not even mention implementation details. Why? —That's just how it is...

- Breimann: I read this name a lot in connection with basic concepts. Did he 'invent' RFs? —Possible, I can and should find out.

- Build new or improve old: Is it really necessary to build something new from scratch? How about improving or extending existing work? —I could do that but it can be dangerous to do so. Depending on the documentation. Even *with* good documentation this can be challenging and I would probably spend around 30% of my time with understanding the implementation. So that is probably not a great option.

8 What Next?

8.1 30.03.2022

Folgende Punkte müssen entschieden werden:

- Focus: Do I focus on decision trees and RFs, or do I expand it to other ensemble methods? —I will not look too deep into decision trees, because I think that this idea is misleading at best. I should really focus on RFs for the first working example.

- Target audience: This is already decided —I will focus on domain experts as the target audience.

- Scope: For the first working example I will keep the scope as small as possible. Coming from a scikitlearn model should be the way.

- Build new: Yes, I will follow Aleksandar's implementation suggestion with streamlit

A meeting has to be set up with George sometime next week.

References

Bostock, M. (2012). *D3.js - data-driven documents.* Retrieved from `http://d3js.org/`

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

Grinberg, M. (2018). *Flask web development: developing web applications with python.* " O'Reilly Media, Inc.".

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

Kuznetsova, N., Westenberg, M., Buchin, K., Dinkla, K., & van den Elzen, S. (2014). Random forest visualization. *Eindhoven University of Technology*.

Livingston, F. (2005). Implementation of breiman's random forest machine learning algorithm. *ECE591Q Machine Learning Journal Paper*, 1–13.

Ming, Y., Qu, H., & Bertini, E. (2018). Rulematrix: Visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics*, *25*(1), 342–352.

Neto, M. P., & Paulovich, F. V. (2020). Explainable matrix-visualization for global and local interpretability of random forest classification ensembles. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 1427–1437.

Tufte, E. R. (1985). The visual display of quantitative information. *The Journal for Healthcare Quality (JHQ)*, *7*(3), 15.

Van Den Elzen, S., & Van Wijk, J. J. (2011). Baobabview: Interactive construction and analysis of decision trees. In *2011 ieee conference on visual analytics science and technology (vast)* (pp. 151–160).

Zhao, X., Wu, Y., Lee, D. L., & Cui, W. (2018). iforest: Interpreting random forests via visual analytics. *IEEE transactions on visualization and computer graphics*, *25*(1), 407–416.