Dynamic Visualization of Scikit Learn Random Forest Models

ADAM WOJTULEWSKI, University of Florida, United States

**ABSTRACT**

Comparing individual decision trees that make up Random Forest Models is a tedious task that is a largely difficult task using existing visualization libraries, which predominantly rely on static visualizations. Current approaches rely on largely static visualizations that have a difficult time adjusting to deep trees with a large number of nodes and provide minimally useful information in assessing the performance of testing data. To address the usability and flexibility problems of existing visualization libraries, this paper introduces a new type of dynamic visualization that affords both studying individual decision trees and comparing different decision trees generated by Random Forest Models. To study individual and groups of nodes of a single decision tree, this library allows users to zoom into nodes of interest using a bounded box technique and allows users to study datapoints contained in a set of nodes by remembering clicked nodes and subsequently generating tables and figures based on the nodes stored. To differentiate data between different trees, users may combine this functionality with tab switching functionality to instantly compare information between two different trees that make up the trained Random Forest classifier. Through this visualization application, it is easier to determine shared trends among different decision trees making up the same Random Forest Model, allowing to better generate rulesets for custom models and to eliminate irrelevant Decision Trees to prevent them from hindering the performance of the decision-making of the Random-Forest Classifier aggregator. [[1]](#footnote-2)

CCS Concepts: • **Artificial Intelligence → Machine Learning → Random Forest**; Visualization; Interactive Display; Data preprocessing

KEYWORDS: Random Forest Dynamic Visualization, Dynamic Visualization, Random Forest Visualizer

ACM Reference format:

Gorge Gubbiotti, Pam Malagò, Steve Fin, Steve Ram, Som Tacchi, Lamp Giovannini, David Bisero, Mike Madami and Glen Carlotti. 2017. Magnetic Normal Modes of Bi-Component Permalloy Structures. In *Proceedings of the ACM on Human-Computer Interaction,* Vol. 5, CSCW1, Article # (April 2021), XX pages, https://doi.org/10.1145/33XXXXX

1 INTRODUCTION

Traditional Computer Science approaches involves programmers developing a program with a specific end goal in mind that adheres to all possible combinations. In fact, it is often pertinent to provide both proofs of correctness and of time complexity to show that an algorithm will provide the expected output. In the past few years though, a subset of computer science, Artificial Intelligence, has become popular with a different approach, developing a solution that has the capabilities of self-teaching and adapting to situations that the programmers may not have originally anticipated. One type of artificial intelligence is machine learning, which has a general goal of trying to teach a computer how to make human-like decisions in order to solve prediction-oriented problems. Due to its frequently impressive predictive performance, the application of machine learning has pervaded the computer science world, with many companies eager to capitalize on its expected benefits. Unlike traditional programming approaches though, due to their tendency of relying on advanced mathematics techniques, such as linear algebra and calculus, where numerous calculations are performed behind the scenes within a program, there is typically a lack of transparency associated with these approaches.

While most machine learning approaches provide limited explanations surrounding their methodology for the average researcher, thus being labeled ‘black-box methodologies’, one example of a machine learning technique that is easier to interpret is the decision tree model. A decision tree is a hierarchical data structure that attempts to divide data with a set of rules, splitting into subtrees depending on whether certain conditions are met.

A diagram of a diagram

Description automatically generated with medium confidence

For a binary classification tree, shown in Figure 1, a node splits into a left subtree if a condition is met and to the right subtree if a condition is not met (or vice versa depending on the implementation). By observing the non-leaf nodes of a tree, it is possible to build a ruleset that is used to get to each leaf node at the bottom of the tree, helping to identify potential groups of interest, a task that could be of great interest in many fields, such as cancer research, where identifying patient subgroups could be of great interest. While a Decision Tree is intuitive, they are often subject to overfitting and consequently, the behavior of a single DT is best evaluated by comparing its result against that of several DTs. This combination of several decision trees, where a prediction y for some input X is based on the aggregated result of several decision trees is in itself a separate machine learning model known as a Random Forest Model.

Random Forest Models are frequently called upon and used because of their tendency to provide more generalizable behavior for testing datasets. For models with good performance, information like feature importance can be quite beneficial to isolate and identify factors that most influence predictions, which could point to areas of future research and discovery. Returning to the earlier example of cancer research, being able to identify that some feature is directly correlated to some desired output, like patient outcome, could help researchers to better treat patients, meaning more lives could be saved. However, sometimes, there can exist situations where certain factors cannot have an impact on the prediction of the model, or a model produces one or several poorly behaving decision trees as part of its aggregation. Whenever this is the case, researchers may opt to update the model by removing these trees, with the intent of producing a better behaving and hopefully better performing model.

With this information in mind, it becomes clear that high quality tools for determining the quality of decision trees could be quite useful. Unfortunately, the current state of visualization code for decision trees is quite lacking, making the process of analyzing decision trees largely tedious and inefficient. With the current visualization tools that exist for decision trees, being able to quickly tell how well a tree is performing from its visualization is not easy and this is especially true for large, dense, and deep trees, as a lot of the information of the nodes becomes quite cluttered and difficult to see. Moreover, the information portrayed can be quite confusing for some learning about this model.

A group of white rectangular objects

Description automatically generated

Taking a look at this example of a visualization for the dtreeviz package, the information portrayed is not immediately clear for most casual users and the text for this tree, which only has a maximum depth of 4, is quite small, meaning it could be difficult for some individuals to see clearly.

Now we observe a tree that has a maximum depth of 16.



Here, besides the general shape of the tree, which hardly reveals any interesting information, nothing significant can be learned from the tree. Hovering over the nodes provides no information and besides being able to zoom into the image by double clicking on the image (when generated in Jupyter Lab or Jupyter Notebooks), which is restricted in its scope, there is a limited interactivity with this image. Considering interactive and colorful libraries are increasingly used both in the workforce and in research environments, the narrow capabilities of this library and other similar decision tree libraries pales in comparison to visualization libraries like Seaborn, Plotly, and Dash, which can be used for other types of figures.

To address the usability and flexibility problems of existing visualization libraries, this paper introduces a new type of dynamic visualization that affords both studying individual decision trees and comparing different decision trees generated by Random Forest Models which was developed over the course of six months as part of an Undergraduate Honors Thesis at the University of Florida. In short, this library allows users to interact and view a set of nodes and it generates figures and tables dynamically based on clicked nodes to allow users to study a set of nodes and evaluate the performance of a decision tree with respect to this set of nodes; each of these will be thoroughly detailed in the paper below. Through this visualization application, it is easier to determine the characteristics and features of a single decision tree and the shared trends among different decision trees making up the same Random Forest Model, important outcomes that could allow researchers to make improved models in the future.

2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

2.1 Sample Fabrication

2.2 Quasi-Static Measurements: MOKE and MFM

*2.2.1 Component Structures.*

*2.2.2 Magnetization.*

*Eavesdropping.*



Fig. 1. MOKE hysteresis loop for the bi-component Py/Co dots array measured along the dots long axis.

2.3 Dynamic Measurements: BLS

*Definition 3.1.* A C-node is a set of live ranges (webs) in the AG or IG that are coalesced. Nodes within the same C-node cannot interfere with each other on the IG. Before any coalescing is done, each live range is a C-node by itself.

Lemma 3.4. *The solution to the C-MWPC problem is no worse than the solution to the MWPC.*

Proof. Simply, any solution to the MWPC is also a solution to the C-MWPC. But some solutions to C-MWPCmay not apply to the MWPC (if any coalescing were made).

2.4 Ground-State Magnetization Determination and DMM Micromagnetic Simulations

*2.4.1 Determined.*

*2.4.2 Micromagnetic.*

|  |  |
| --- | --- |
|  | (1) |

where is the azimuthal (polar) angle of the magnetization (the time dependence is omitted). The second derivatives of the energy density depend on the micromagnetic cell indexes, and through them on the material index corresponding either to Py or Co. The expressions of *E*ext, *E*exch*, E*dmg and *E*ani are the same as the ones of the single-component system.

“Instead of using pairwise RTS/CTS frequency negotiation we propose lightweight frequency assignments, which are good choices for many deployed comparatively static WSNs. We develop new toggle transmission and snooping techniques to enable a single radio transceiver in a sensor device to achieve scalable performance, avoiding the nonscalable “one control channel + multiple data channels” design.”

|  |  |
| --- | --- |
|  | (2) |

Therefore one can observe either an in-phase (acoustic) or an out-of-phase (optical) character of the modes, with respect to the precession of the in-plane magnetization components in adjacent Py and Co dots.

Table 1. Frequency of Special Characters

|  |  |  |
| --- | --- | --- |
| Non-English or Math | Frequency | Comments |
| Ø | 1 in 1,000 | For Swedish names |
| $ | 4 in 5 | Used in business[*a*](#tb1fn1) |

*Source:* This is a table source note

*a*This is table footnote

1. Instead of using pairwise RTS/CTS frequency negotiation we propose lightweight frequency assignments, which are good choices for many deployed comparatively static WSNs.

2. We develop new toggle transmission and snooping techniques to enable a single radio transceiver in a sensor device to achieve scalable performance, avoiding the nonscalable “one control channel + multiple data channels” design.

3 RESULTS AND DISCUSSION

3.1 Magnetization Curves and MFM Characterization



Fig. 2. MFM images of the bi-component Py/Co dots for different values of the applied magnetic field which are indicated by greek letters along both the major and minor hysteresis loop.

**ALGORITHM 1:** Iterative Algorithm

*current\_position ← center*

*current\_direction ← up*

*current\_position* is inside circle

**while** *current\_position* ***is inside circle***, **do**

*neighborhood*  ← all grid hexes within two hexes from *current\_position*

**for** ***each*** *hex* ***in*** *neighborhood*, **do**

**for** ***each*** *neuron* ***in*** *hex* **do**

convert *neuron\_orientation* to *vector*

scale *vector* by *neuron\_excitation*

*vector\_sum* ← *vector\_sum* + *vector*

**end**

**end**

normalize *vector\_sum*

*current\_position* ← *current\_position* + *vector\_sum*

*current\_direction* ← *vector\_sum*

return *current\_position*

**end**

3.2 Field Dependent BLS Measurements and DMM Calculations

[Fig. 3](#fig3) displays the frequencies of BLS peaks plotted as a function of the applied field magnitude starting from positive values. The field is then decreased and reversed following the upper branch of the hysteresis loop



Fig. 3. Dependence of the magnetic eigeinmode wave frequency on the applied field strength.



Fig. 4. Calculated spatial distribution of the in-plane dynamic magnetization.



Fig. 5. Full point are the frequencies measured along the minor hysteresis.

3.3 Analysis of the Dynamic Coupling as a Function of the Gap Size

Table 2. Comparison of Coefficients from Atomistic

|  |  |  |
| --- | --- | --- |
| Atm | MS-CG | MS-CG/DPD |
| 1.78 | 14.32 | 1.74 (-2%) |
| 0.43 | 31.00 | 0.40 (-7%) |
| 0.062 | 15.61 | 0.048 (-23%) |
| 0.032 | 9.76 | 0.024 (-24%) |
| 0.020 | 4.66 | 0.015 (-25%) |
| 0.012 | 2.32 | -”- |
| 0.0076 | 0.016 | -”- |



Fig. 6. Calculated frequency evolution of modes detected in the BLS spectra.

4 CONCLUSIONS

ACKNOWLEDGMENTS

This work was completed with the guidance and supervision of Professor Albert Ritzhaupt, Professor Emmanuel Dorley, and Professor Rong Zhang, professors at the University of Florida. Without their words of advice and pointers, this work would look very different than it does right now.

A HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are di.erent in the appendices. In the appendix environment, the command section is used to indicate the start of each Appendix, with alphabetic order designation (i.e., the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure within an Appendix, start with subsection as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

REFERENCES

[1] Patricia S. Abril and Robert Plant. 2007. The patent holder’s dilemma: Buy, sell, or troll? Commun. ACM 50, 1 (Jan. 2007), 36–44. DOI: http://dx.doi.org/10.1145/1188913.1188915

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless Sensor Networks: A Survey. Comm. ACM 38, 4 (2002), 393–422.

[3] David A. Anisi. 2003. Optimal Motion Control of a Ground Vehicle. Master’s thesis. Royal Institute of Technology (KTH), Stockholm, Sweden.

[4] P. Bahl, R. Chancre, and J. Dungeon. 2004. SSCH: Slo.ed Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In Proceeding of the 10th International Conference on Mobile Computing and Networking (MobiCom’04). ACM, New York, NY, 112–117.

[6] Jacques Cohen (Ed.). 1996. Special Issue: Digital Libraries. Commun. ACM 39, 11 (Nov. 1996).

[7] Bruce P. Douglass. 1998. Statecarts in use: structured analysis and object-orientation. In Lectures on Embedded Systems, Grzegorz Rozenberg and Frits W. Vaandrager (Eds.). Lecture Notes in Computer Science, Vol. 1494. Springer-Verlag, London, 368–394. DOI:http://dx.doi.org/10.1145/3-540-65193-429

[8] Ian Editor (Ed.). 2008. The title of book two (2nd. ed.). University of Chicago Press, Chicago, Chapter 100. DOI: <http://dx.doi.org/10.1145/3-540-09237-4>

Journal Name:ACM Trans. Graph.

Volume:4

Issue:2

Year:2017

Month:October

Article Seq:XXXX

DOI:124564

Copyright Year:2017

Copyright Statement:rightsretained

ISSN:0730-0301

Price:$15.00

1. [↑](#footnote-ref-2)