# NBTree implementation to link similar questions on stack overflow.

Pranita Sharma
Master of Computer Science
North Carolina State University
prsharma@ncsu.edu

Advait Trivedi
Master of Computer Science
North Carolina State University
astrived@ncsu.edu

Sudipto Biswas
Master of Computer Science
North Carolina State University
sbiswas4@ncsu.edu

**Abstract**

Critique the use of deep learning methods for not being reproduceable and state alternative learning models that are trained quicker.

***Keywords*** Deep Learning, NBTree, SVM, Differential Evolution

## 1 Criteria

### 1.1 Model readability

A readable model is one that can be explained easily to its users. A model is only as useful as it is understandable by the people who will use it. So it becomes our job to simplify complicated concepts so that business facing personnel can make the best of the model provided to them. To do that, the first step would be to understand the audience. Readability is also fundamental to maintenance of the code. If it is easily understandable it will also be easier to experiment with it and improve upon it.

### 1.2 Actionable conclusions

Actionable conclusions is the missing link between data and business value. Often what is really being offered by many analytics solutions is just more data or information—not useable insights. Again not all insights will be actionable. It is something that makes us rethink something and pushes us in a new direction. Whether the results are useable at all depends on whether it has the following properties: i. Alignment, ii. Context iii. Specificity, iv. Novelty, v. Clarity.

### 1.3 Learnability and Repeatability of the results

Recent developments in machine learning and analytics tools have come at the cost of massive computation cost. Most deep learning methods are CPU/GPU-intensive methods. For eg. Le [3] used a cluster with 1,000 machines (16,000 cores) for three days to train a deep learner. The problem is not solvable by waiting for faster CPUs/GPUs. We can no longer rely on Moore's Law to double our computational power every 18 months [2]. Power consumption and heat dissipation issues effect block further exponential increases to CPU clock frequencies. Cloud computing environments are extensively monetized so the total financial cost of training models can be prohibitive, particularly for long running tasks.

### 1.4 Multi-goal reasoning

Goals are a unifying structure across the variety of intelligent systems, and reasoning about goals takes many forms. Goal reasoning, which concerns the ability of an intelligent agent to reason about, formulate, select, and manage its goals/objectives [4]. Goal reasoning differs from frameworks in which agents are told what goals to achieve, and possibly how goals can be decomposed into sub goals. Existing goal modelling techniques, however, have shown limitations of expressiveness in coping with complex real-world problems where a system has to deal with multiple objectives and achieve a balance. So it becomes imperative for us to come up with algorithms that has the ability to manage all the different goals of a system.

### 1.5 Anomaly detection

Anomaly detection (also outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Removing the anomalous data from the dataset often results in a statistically significant increase in accuracy. So if the model is able to identify the outliers/anomalies in the dataset and treat them according to the problem at hand, the results will be a lot more relevant. Several Business use cases such as fraud detection, anomalous pharmacy detection are extremely valuable. To zero in on the anomaly would result in capturing the perpetrator or finding the fault in a newly manufactured drug.

### 1.6 Incremental

A good model should be constantly updateable. We should be able to add to its training data without causing it to start over as is the case with decision trees.

Some learners such as Naive Bayes are good at this. In the era of big data, this criteria has become almost a necessity for data miners. Incremental models gather new insight everyday as they keep up with the trend of data. Another value of incremental models is the generality of processing data. The model should adapt to extract information from any form of data source , be it text or multimedia.

### 1.7 Sharable

The exponential growth in the field of A.I. is a result of the open sourcing of various models. Thus an important objective at hand is to make sure that the learners trained can be implemented on other hardware and software platforms with ease. Enabling interoperability between different frameworks and streamlining the path from research to production will increase the speed of innovation in the AI community. A good example would be the ONNYX project [7] initiated by Microsoft.

### 1.8 Context aware

Context-aware data mining is an application able to sense and analyze the context from various sources and which takes actions suited to contexts for improving performance and efficacy of decision-making by identifying the unknown factors. Contexts consist of circumstantial aspects of the user and domain that may affect the data mining process. When the information has to be conveyed from one element to another we need to let the receiving element know the reference of our discussion. Chen and Kotz [5] defines context as the set of environmental states and rules that either determines an application behavior or describes where the event occurs. Lack of context-awareness leads to missing a lot of critical and useful in- formation that would affect the data mining process and thereby, affecting the data mining results. In real-world and live data sets, the context factors that constitute The context will make the system understand and adapt the data mine process and thereby providing the users with a time sensitive data accurately, efficiently and in a precise manner. [6]

### 1.9 Self-tuning

A self-tuning system optimizes its own internal running parameters in order to maximize or minimize the fulfilment of an objective function. A good data miner should be able to tweak itself to achieve better results. This also reduces the need for human oversight.

## 2 Key Criteria

**Learnability and repeatability of results:**
At the start of this course one of our teammates, Advait Trivedi, got into an extended discussion (almost a debate) with Professor Menzies as he was critical of Data Science being included as a part of science. He was taught that Data science had its basis in mathematics and statistics, and we don't question 1+1=2 right? He responded on how this debate is less of a scientific one and more of a philosophical one. He responded saying "Do you know what the Phd stands for? Doctor of philosophy, Einstein didn't just believe Newton was right, He challenged him. That is the nature of science, If you can't challenge a concept, it is not science"
An important part of knowledge is "no"; i.e. the ability to critically assess something, and to recognize when one idea is better, smarter, than another.
"Science" is the process of communities sharing and reviewing and improving each other's ideas. Sadly, most "data science" is not about "science". Rather its about vendors selling you stuff that does not work properly and does not ring an alarm when it starts failing.
So a million million people can run data miners. But how many now when those data miners start going wrong? And how to fix faulty models?
Do this subject, learn answers to those questions, become the data scientist everyone needs to consult with when things start going wrong.
This criteria is of imminence as big corporations fund research that require immense computational resources to train, and the sad part is they become state of the art benchmark models for new learners. For instance Google's GoogleLeNet is 22 layers deep[8] and uses exa-bites of data to be trained. How would we ever test for faults in a system this big and complex? How much time and resource would we need to comprehend, replicate the system? Is it even science? Our primary objective is to pick a problem used by a complex learner and try using a simpler learner to implement the task. We sincerely hope to use the most learnable and repeatable approaches to tackle the problem. Our Critique will assume the deficit of deep learning model on the key criteria discussed above.

## 3 Critique

4. **3.1** Any Deep learning method needs to be assessed with respect to its computational cost.Most deep learners need hours(or even weeks) to train a model.

Such long training time limits the repeatability and learnability of the results.

Here:

i. Repeatability refers to other researchers being able to repeat, improve, or even refute that original work.

ii. Learnability refers to the researcher being able to test the stability of the outputs via repeated runs with different parameters.

Any new Deep learning approach should be baselined against some simpler and faster alternatives.

In their paper [Xu], Xu et al. used a convolution neural network (CNN) to predict whether two questions on Stackoverflow are linkable. Such CNNs are highly computationally expensive, often requiring network composed of 10 to 20 layers, hundreds of millions of weights and billions of connections between units [9]. Even with advanced hardware and algorithm parallelization, training deep learning models still requires hours to weeks. XU report that their analysis required 14 hours of CPU.

In the same vein as we increase the number of neurons in every layer and number of layers in the network, there are limits on speed of computation and sometimes on the hardware needs and so on. This too has limits on devices they can run on and in scaling on the web and small devices. This makes it financially unviable. This makes it difficult for a large set of researchers to work with it.

Also as we scale and make applications on the web, the dependence on high performance devices will be a bottleneck.

**3.2** Deep Learning (and deep neural networks), are black-box approaches that work amazingly well to describe data but provide little to none understanding of generating mechanisms. As a consequence, they fail to be scalable to domains for which they were not trained for. Whereas we can easily explain a logistic/lineaer regression model or a simple decision tree.

**3.3** They require tons of data to be trained before doing anything interesting, and they need training every time they are presented with (even slightly) different data. It doesn't work on small data samples.

**3.4** It requires quite a lot of technical and mathematical expertise to set one up well from scratch (or correctly build and tune one in R/Python).

**3.5** As seen in [2], section 2.3.2 elaborates on how research in deep learning often ignore perks of several base line learning algorithm and only paint a broader "Accuracy" picture. For a lot of business would care for stuff other than accuracy which deep learning might underperform on ("No free lunch")

## 4 Review

To improve upon the results of the XU [9], There are the following learners for improving our objective: Naive Bayes and Fast Frugal Trees.

[A] Naive Bayes is often used as a baseline in classification because it is fast and easy to implement. Naive Bayes is fast. Fast because all it needs are the prior probability values that do not change and can be stored ahead of time.[17] The same probability values are reused in while calculating the posterior. During training the model parameters of Naive Bayes classifier i.e., apriori and conditional probabilities are determined using a deterministic set of steps:

1. Counting

2. Dividing,

two simple operations that actually make it so fast. Naive Bayes doesn't bother with iterations, epochs, nor optimization of a cost equations or error back-propagation. Also during inference the model parameters are stored for inference (on new data). During inference phase, it uses Bayes equation to compute the posterior probabilities [18]. This involves trivial arithmetic operations like addition and multiplication and further normalization is only a division by a scalar.

[B] A fast-and- frugal tree (FFT) [13] is a set of hierarchical rules for making decisions based on very little information (usually 4 or fewer). Specifically, it is a decision tree where each node has exactly two branches, where one (or in the cast of the final node, both) branches is an exit branch. FFTs are simple, transparent decision strategies that use minimal information to make decisions [14]. They are frequently preferable to more complex decision strategies (such as logistic regression) because they rarely over-fit data

[15] and are easy to interpret and implement in real-world decision tasks.

Fast-and- frugal trees can be used as decision-making tools which operate as lexicographic classifiers, and, if required, associate an action (decision) to each class or category. The basic elements on which to ground a binary classification are (sets of) cues. The fast-and-frugal tree establishes a ranking and, according to the ranking, a "topology" of the tree. Once the ranking is established, the fast-and- frugal tree checks one cue at a time, and at each step, one of the possible outcomes of the considered cue is an exit node which allows for a decision. [12] FFTs are extremely frugal (i.e., do not search for much cue information), highly robust, and well adapted to the payoff structure of a task.

## 5 Planning

The primary objective is to link questions on stack over flow on their relevance. Basically we would want to recognize if 2 questions are same, relatable or disjoint as proposed by Xu et. Al. [9]. However the deep learning method proposed requires 14 hours to train.

Fu et. al.[2] showed how by simply tuning the SVM, a model could be trained 84 times faster hours than a powerful Convolutional Neural Network[9]. Since this was a model trained in 14 minutes, our objective will be to replicate the findings and to test them with a baseline algorithm which can be easily repeatable.

The baseline algorithm will be a NBTree: Decision Tree Hybrid Naïve Bayes Classifier as proposed by [10]. The algorithm is similar to the classical recursive partitioning scheme with the leaves representing a naïve based probabilistic category. The utility of a node is computed by discretizing the data and computing the 5-fold cross validation accuracy estimate. Discretizing helps the validation operate in linear time. It would be interesting to implement a semLDA proposed by Ferrugento et. al.[11] to incorporate our classifier with a LDA-based topic model that would be sensitive to word sense. The implementation of this is present at :https://github.com/aferrugento/SemLDA.

The steps for implementing NBTree are:

**Figure 1. NBTree implementation.**

Input: a set $T$ of labelled instances.
Output: a decision-tree with naive-bayes categorizers at the leaves.

1. For each attribute $X_i$, evaluate the utility, $u(X_i)$, of a split on attribute $X_i$. For continuous attributes, a threshold is also found at this stage.
2. Let $j = \arg\max_i(u_i)$, *i.e.*, the attribute with the highest utility.
3. If $u_j$ is not significantly better than the utility of the current node, create a Naive-Bayes classifier for the current node and return.
4. Partition $T$ according to the test on $X_j$. If $X_j$ is continuous, a threshold split is used; if $X_j$ is discrete, a multi-way split is made for all possible values.
5. For each child, call the algorithm recursively on the portion of $T$ that matches the test leading to the child.

Figure 3: The NBTree algorithm. The utility $u(X_i)$ is described in the text.

## 6 References

[1]"A general software readability model." *MCS Thesis available from (http://web.eecs.umich.edu/~weimerw/students/dorn-mcs-paper.pdf )* (2012).

[2] Fu, Wei, and Tim Menzies. "Easy over Hard: A Case Study on Deep Learning." *arXiv preprint arXiv:1703.00133* (2017).

[3] V Le. 2013. Building high-level features using large scale unsupervised learning. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing . IEEE, 8595–8598.

[4] Nguyen, Chi Mai, et al. "Multi-objective reasoning with constrained goal models." *Requirements Engineering* (2016): 1-37.

[5] Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research.Dartmouth Computer Science Technical Report TR2000-381 (2000)

[6] Singh, Sachin, Pravin Vajirkar, and Yugyung Lee. "Context-based data mining using ontologies." *Conceptual Modeling-ER 2003* (2003): 405-418.

[7] https://github.com/onnx/onnx

[8] Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[9] Bowen Xu, Deheng Ye, Zhenchang Xing, Xin Xia, Guibin Chen, and Shanping Li.

2016. Predicting semantically linkable knowledge in developer online forums via
convolutional neural network. In Proceedings of the 31st IEEE/ACM International
Conference on Automated So.ware Engineering. ACM, 51–62.
[10] Kohavi, Ron. "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid." *KDD*. Vol. 96. 1996
[11] Ferrugento, Adriana, et al. "Can Topic Modelling benefit from Word Sense Information?." *LREC*. 2016.
[12] Martignon, Laura; Vitouch, Oliver; Takezawa, Masanori; Forster, Malcolm. "Naive and Yet Enlightened: From Natural Frequencies to Fast and Frugal Decision Trees", published in Thinking : Psychological perspectives on reasoning, judgement and decision making (David Hardman and Laura Macchi; editors), Chichester: John Wiley & Sons, 2003.
[13] Martignon, Vitouch, Takezawa, & Forster, 2003
[14]Gigerenzer & Todd, 1999; Gigerenzer, Czerlinski, & Martignon, 1999
[15]Gigerenzer & Brighton, 2009
[16]Jenny, Pachur, Williams, Becker, & Margraf, 2013
[17] Rennie, Jason D., et al. "Tackling the poor assumptions of naive bayes text classifiers." *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003.
[18] Bharadwaja, Sameera H, Shivkumar, Abhishek, Why is the Naive Bayes algorithm fast?, Quora, 2015