Transparent Boosting Tree: An Interactive Machine Learning Framework

Tianyi Zhou
University of Washington
tianyizh@uw.edu

Tianqi Chen
University of Washington
tqchen@uw.edu

Luheng He
University of Washington
luheng@uw.edu

ABSTRACT

Machine learning (ML) is believed to be an effective and efficient tool to build reliable prediction model or extract useful structure from an avalanche of data. However, ML is also criticized by its difficulty in interpretation and complicated parameter tunning. In contrast, visualization is able to well organize and visually encode the entangled information in data and guild audiences to simpler perceptual inferences and analytic thinking. But large scale and high dimensional data will usually lead to the failure of many visualization methods. In this paper, we close a loop between ML and visualization via interaction between ML algorithm and users, so machine intelligence and human intelligence can cooperate and improve each other in a mutually rewarding way. In particular, we propose "transparent boosting tree", which visualizes the learning process and result of gradient boosting tree to user, and involve user's feedback operations to the structure of trees into the learning process. In this system, ML is in charge of updating weights in learning model and filtering information shown to user from the big data, while visualization is in charge of providing a visual understanding of ML model to user. It combines the advantages of both ML in big data statistics and human in decision making based on domain knowledge. We develop a user friendly interface for this novel learning method, and apply it to several datasets collected from real applications. Our study shows that making ML transparent by using interactive visualization can significantly improve the exploration of ML algorithms and integrates both machine and human intelligence.

Author Keywords

Ensemble boosting tree, decision tree, interactive visualization, interpretable machine learning, transparent machine learning

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous—*Optional sub-category*

INTRODUCTION

Machine learning (ML) plays an increasingly important role in data science today with great promises to improving our life. By fitting data to a flexible discriminative or generative model leveraging the relationship between variables and latent structures, ML is able to learn a prediction model or structured information that is consistent to the given data with statistical importance. The obtained prediction model or data structure can be applied to analysis and classification of new data instances. In this era, the avalanche of information gives rise to a growing interests in big data. While data collection becomes cheap, various current ML algorithms are designed for processing these large-scale and high dimensional data with noise with promising efficiency and accuracy.

However, non-experts often treat ML as a fancy "black box" with few convincing interpretation of the model structure and learning process. In addition, even for ML researchers, how does the ML mechanism well-defined on statistical assumptions work on practical data, and how is the ML model updated in each iteration according to input features, are not transparent and usually ignored. The understanding and further exploration of ML models are highly limited in this case. Especially when applied to specific applications, it is essential to study when and why the ML algorithms work better or worse than expected, and then adjust the model accordingly. Therefore, a transparent ML tool that can clearly show the learning process to users can significantly facilitate the exploration of ML models.

Moreover, parameter tuning and determination of model complexity are two tricky problems that are critical to the performance of ML algorithms. Although there exists some statistical methods for solving them, it turns out that human intelligence based on domain knowledge always performs better and is more reliable on these decision-making problems, e.g., determining the depth of a decision tree, feature selection, and reasoning of performance enhancement. In addition, the emergence of human intelligence often offer better interpretation than machine intelligence. However, in the previous ML systems, it is also hard to involve users wisdom into learning process to improve both the generalization ability and interpretability of the ML models.

In contrast, visualization is a popular and powerful tool to explore data by concentrating human intelligence on visually encoded informations extracted from data. It shows different types of features of data in separate and interpretable ways, and also takes the relationship between features and samples

Final Project of CSE 512 Data Visualization, Winter 2014.

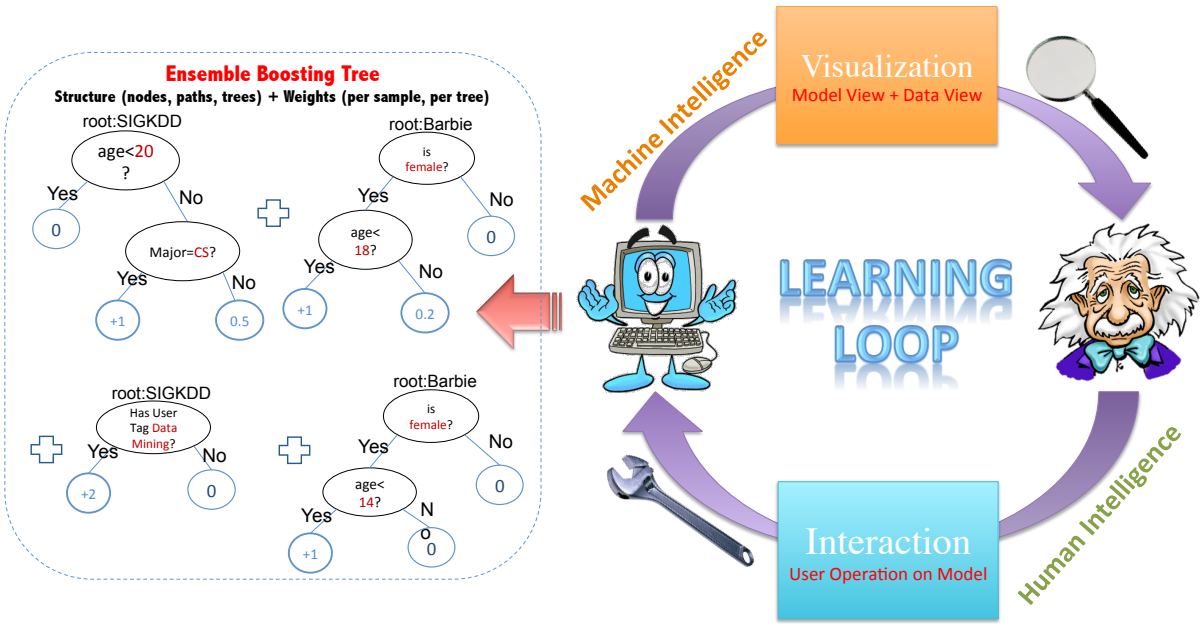

Figure 1. Interactive learning loop between machine learning and visualization by integrating both machine intelligence and human intelligence.

into account. The visual representation of data is expected to help, facilitate and improve the comprehension of audiences, and thus stimulate more effective analytic thinking and decision making. However, visualization usually suffers from high dimensions and large volume of data. In this situation, the audiences suffer from an information overload caused by the big data, and the cognitive calculation could be slowed.

Main Idea

In this paper, we are going to close a loop between machine learning and visualization so they can benefit each other in data analysis and complete an interactive learning system that makes ML transparent to users. An overview of this system is given in Figure 1. In each iteration of the loop, ML algorithm updates the prediction model according to given data and user feedback, then visualization provides a visual encoding to the model structures and the prediction results on training data by using the current model. In the visualization interface, user is able to track both the model structure and the prediction of training data, and operate on the model to change its structures based on domain knowledge. These user operations will be the input of the next run of the ML algorithm. ML algorithm will update the model by treating user's feedback as constraint in its learning. In this loop, ML is in charge of updating model, choosing useful information to visualize, and processing user's feedback, while user is in charge of modification to the updated model.

In particular, we choose gradient boosting tree (GBT) as our ML algorithm in the loop because of its appealing performance in lots of practical problems. GBT is an ensemble of decision trees that are established in a forward stagewise fashion similar to boosting. It learns an additive model defining decision rules for effective prediction from input features, such as classification and regression. GBT has been well known as the state-of-the-art method for many challenging real world problems. Each step of GBT learning can be treated as learning a decision tree based on all the previous trees. Each node of each decision tree corresponds to a selected feature and a decision rule.

In the interactive learning system called "transparent boosting tree (TBT)", visualization interface provides four views of model and data, i.e., feature view listing the feature groups and highlighting the selected features, forest view listing all previous trees, tree view showing the detailed structure of the current tree, path purity view capturing the prediction results of training set on each node along a selected path on the tree, and history view tracking both the training error and test error of each update in the learning process. In each step, users are accessible to these views to evaluate the current model, and our system allows the users to add/remove a tree, select feature group for building a new tree, remove a node on the current tree, expand a leaf node, and go back to any previous model in the learning loop. Then GBT algorithm will take these user operations as inputs to update the ensemble of decision trees accordingly.

In this paper, we implement TBT by building an interactive visualization system, and apply it to two practical datasets, mushroom and fusion health. The former aims at justifying if a given mushroom is poisonous or not, while the latter aims at predicting the healthy status of a patient. TBT is demonstrated to be a promising tool for exploration of GBT, visualization of ML model, and human-ML interaction.

RELATED WORKS

There are several, though not many, early efforts on improving the interpretation of ML models by using effective visualization. EnsembleMatrix [] enables users to choose

any combination of multiple classifiers to build a combination model, after showing users the confusion matrix of all classifiers. Thus it provides an interaction visualization tool emerging user intelligence in ML process. However, both the training and mode structure of each classifier are still invisible to users.

Some recent works like BigML [] focuses on the visualization of decision tree, which is related to ensemble boosting tree in our project. However, decision tree is less powerful and also simpler than ensemble boosting tree. In addition, they compute all possible paths before showing them to users rather than instantaneously update them, and allow fewer operations of users on the tree. So the loop has not been really built, the benefits of interaction is limited, and the computation could be expensive.

TRANSPARENT BOOSTING TREE

Learning Algorithm for Gradient Boosting Tree

One goal of TBT is to visualize the learning process and the model structure of GBT. GBT builds additive model $F(x)$ composed of M decision trees $h_m(x)$ with input features x, i.e.,

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x). \quad (1)$$

Each decision tree $h_m(x)$ is a weak learner that can handle data of mixed type and to model complex nonlinear function. Similar to boosting, GBT builds the additive model in a forward stagewise fashion such that

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x). \quad (2)$$

In each stage, fixing the previous ensemble of trees $F_{m-1}(x)$, the current decision tree $h_m(x)$ is chosen to minimize the loss function L.

$$h_m(x) = \arg\min_h \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) - h(x_i)). \quad (3)$$

This minimization problem can be solved by using a steepest gradient descent, which is

$$F_m(x) = F_{m-1}(x) + \gamma_m \sum_{i=1}^{n} \nabla L(y_i, F_{m-1}(x_i)), \quad (4)$$

where the step size γ_m is determined by line search. The final additive model is usually represented by the combination of all paths on all the trees with different weight per path.

In learning of each decision tree, the gradient $\nabla L(y_i, F_{m-1}(x_i))$ is firstly assigned to each training sample as its importance weight in learning the current tree. Then the nodes and their corresponding decision rules defining the tree are selected in a greedy manner from root to leaves. Specifically, for each node, learning algorithm chooses the best threshold for each available feature by evaluating all possible thresholds. The selected threshold defines the best decision rule associated to each feature. By using both the importance weights and Hessian matrix, we can thus compute the information gain caused by applying each feature and its decision rule on the node. The pair which results in the maximal information gain is selected to define the current node.

In TBT, we involve the user operations into the learning process of GBT. In particular, TBT allows users to operate on the features and trees that used to be selected by using greedy method, and to determine the number of trees and the depth of a path. Thus the user feedback can be treated as a human regularization or constraint to limit the model complexity of GBT, and to correct the possible mistakes made by pure greedy selection. This is expected to be effective in avoiding overfitting, simplifying model, and improving intepretability. Because human with domain knowledge is believed to be good at those decision making tasks.

We let the GBT learning algorithm in charge of the update of thresholds and importance weights, because computer is good at searching and optimizing these numerical variables. Therefore, the advantages of both human intelligence and machine intelligence can be combined in TBT.

Design of Interactive Visualization Interface

The visualization interface of TBT is composed of one initialization tool bar and five views, i.e., feature view listing the feature groups and highlighting the enabled features, forest view listing all previous trees, tree view showing the detailed structure of current tree, path purity view capturing the prediction results of training set on each node along a selected path on the tree, and history view tracking both the training error and test error of each update in the learning process.

In the initialization tool bar, user is able to choose the dataset to analyze, the method to group features in the feature view, the initial number of trees in GBT, the initial maximal depth of each tree. Clicking the "REBUILD!" button will provide an initial GBT model, which is learned by original GBT algorithm and visualized in the five views.

In the following tables, we list the major visually encoded information and interaction/user operation in each view.

Table 1. Visual encoding and user operation in feature view.

Visual Encoding
Feature group (text)
Feature name (text)
No. of features per group (integer)
User Operation
allow the feature (right click feature & select)
block the feature (right click feature & select)

Table 2. Visual encoding and user operation in forest view.

Visual Encoding
Tree index (integer)
Feature and decision rule for root node (text)
User Operation
Collapse the whole tree (left click)

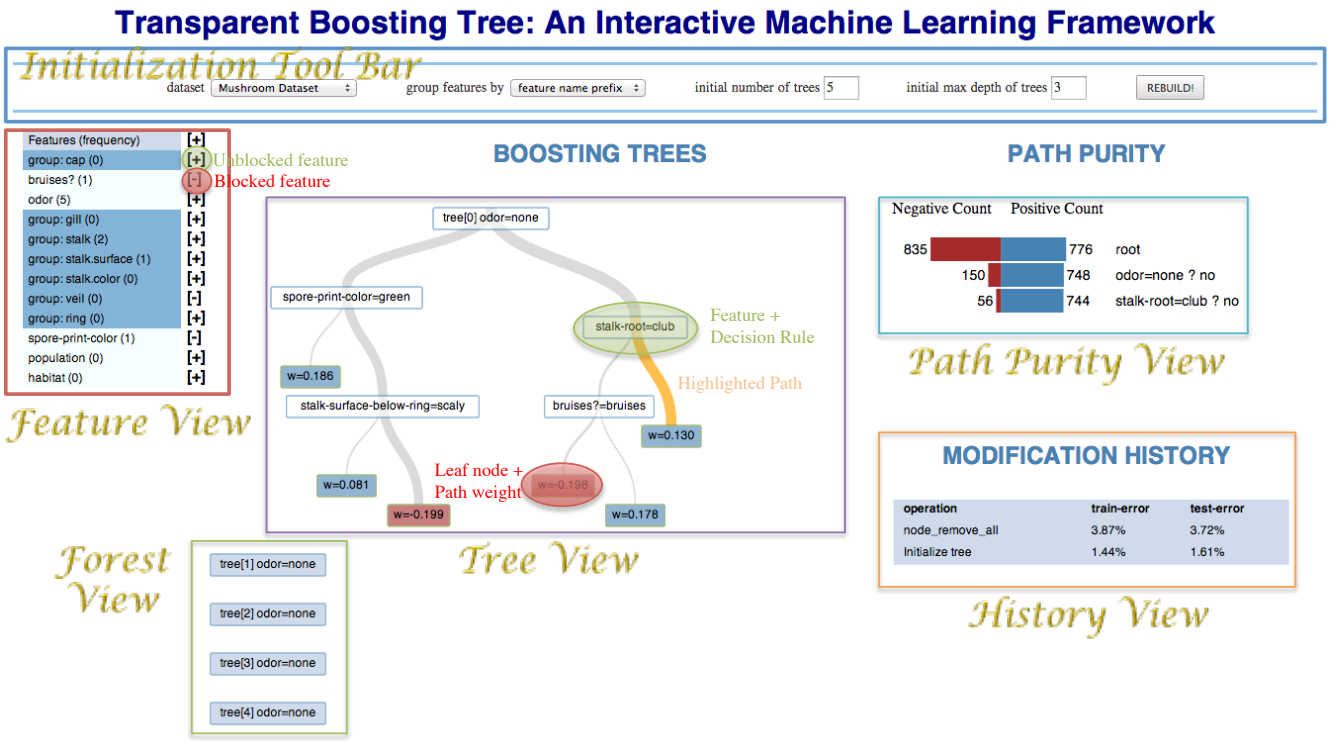

Figure 2. Views in visualization interface of transparent boosting tree.

Table 3. Visual encoding and user operation in tree view.

Visual Encoding
Feature (text)
Decision rule (text)
Path weight on leaf node (ratio)
Major class on leaf node (color)
No. of samples on each edge (width)
Highlight a path (brushing a path)
User Operation
Remove the current tree (right click root node & select)
Grow a new tree (right click root node & select)
Remove the node on the current tree (right click node & select)
Remove the node on all trees (right click node & select)
Expand the leaf node on the current tree (right click node & select)
Expand the leaf node on all trees (right click node & select)

Table 4. Visual encoding in path purity view.

Visual Encoding
Prediction results on a path (brushing a path in tree view)
Negative and positive counts on nodes (bar charts)
Decision rule (text)

In the developing visualization, we will provide two views. The model view shows the structure of model in learning process, i.e., several decision trees from root node to current leaf nodes, as well as the weights and the features corresponding to each node. The data view shows both the overview of the prediction results on all data points and the data points classified by the selected node. By assigning different feature to the selected node, users are able to track the

Table 5. Visual encoding and user operation in history view.

Visual Encoding
Historical operation (text)
Training error (ratio)
Test error (ratio)
User Operation
Go back to former model (right click row & select restore)

resulting changes in prediction performance.

The model view provides users a clear explanation of the current model, e.g., what features were selected to build decision rules, and how these features cooperate with each other. So users can justify which nodes on the current model are consistent with domain knowledge or common sense, and which are not. The data view provides users an evaluation evidence from the performance of current model on training data. In particular, the data view will includes bar charts showing the numbers of positive and negative samples after applying decision rule on each node or along each path.

By showing users both the model structure and performance on training data, we expect to collect the wisdom of users from both their domain knowledge and judgement based on the prediction result on data via their feedback. In our visualization tool, users have complete freedom to modify the structure of each tree, e.g., to choose feature for each node,

to prune an existing node/path, and to add a new node/path. The model will be updated according to these instantaneous feedbacks from users, and we will show the new views for the updated model.

In the learning process, ML algorithm is in charge of learning candidates of model, estimating the numerical weights of the model, and updating model according to user feedback, while users are in charge of modifying the ML model by their domain knowledge and feedback from the current learning results. Interactive visualization bridges these two components by conveying the feedback of one to the other. Therefore, we can close a loop between these two so that one helps enhance the performance of the other in a mutually rewarding way, until we achieve a promising ML model.

RESULTS

DISCUSSION

FUTURE WORK

REFERENCES

1. How to Classify Works Using ACM's Computing Classification System.
http://www.acm.org/class/how_to_use.html.