machinalis

GPU Accelerated Gradient Boosting

Ariel Wolfmann

2018

About me

machinalis

- Computer Scientist from FaMAF, UNC. Cordoba, Argentina
- Machine Learning Developer at Machinalis
- Love to bridge the gap between Science and Business.
- Twitter: @osowolfmann14
- Mail: awolfmann@machinalis.com

Agenda

machinalis

- Supervised Learning
- Decision Trees
- Ensemble Methods
- Bagging
  - Random Forests
- Boosting
  - Gradient Boosting Trees
- **GPU Accelerated Gradient Boosting**

Supervised Learning

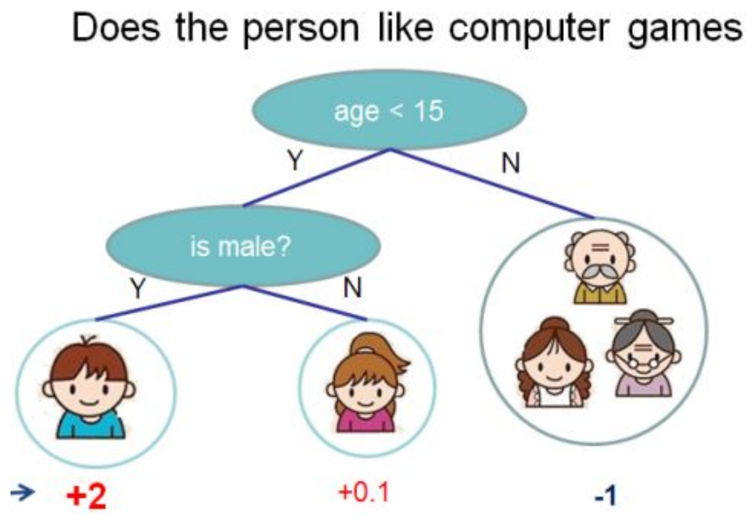
- Train a model with labelled data.
- Each data point is a pair (**xi**, yi).
    - **X**i is a feature vector -> attributes that represent an object.
    - yi is the label of the object.
- Learn to predict the label of a new data point based on what it learned from the train set.

Classification: predict a **discrete** variable or category.

Regression: predict the value of a **continuous** variable.

Decision Tree (CART)

- **Decision Rules:** sequence of binary selections
- Can be applied to both **classification** and **regression** problems.
- Rules based on variables' values are selected to get the **best split** to differentiate observations based on the dependent variable.
- Once a rule is selected and splits a node into two, the same process is applied to each "child" node (i.e. it is a **recursive procedure**).



Ensemble methods

- Combine models to improve performance.
- Mix weak learners to get a strong one.
    - **Bagging**
    - **Boosting**
    - **Stacking**

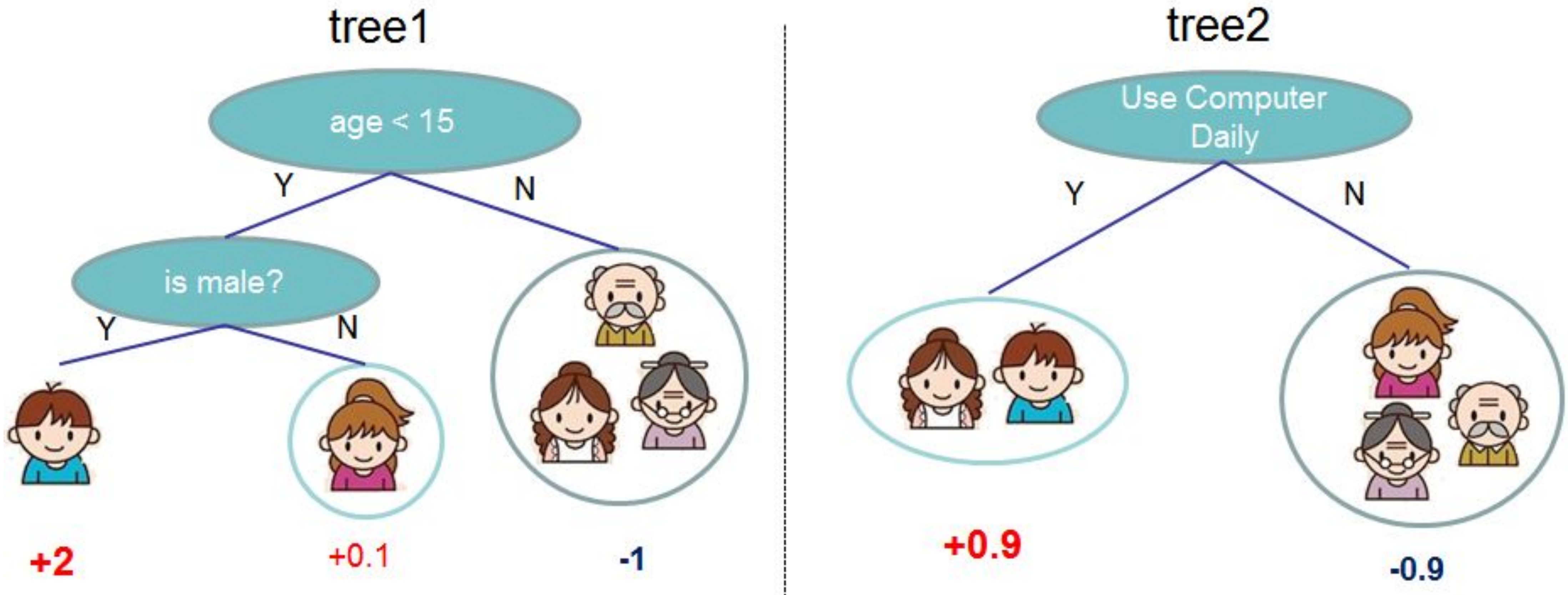
Ensemble methods -> Bagging

- Train each weak learner in a **parallel** fashion.
- Involves having each model in the ensemble vote with **equal weight** as a “committee” and calculate the average of the predictions.
- Trains each model in the ensemble using a **randomly drawn subset of the training set.**

Bagging -> Random Forests

- Bagging of decision trees
- **Random sample with replacement**.
- **Random subset of the features.**
- This bootstrapping procedure leads to **better model performance** because it decreases the variance of the model, without increasing the bias.

Random Forests



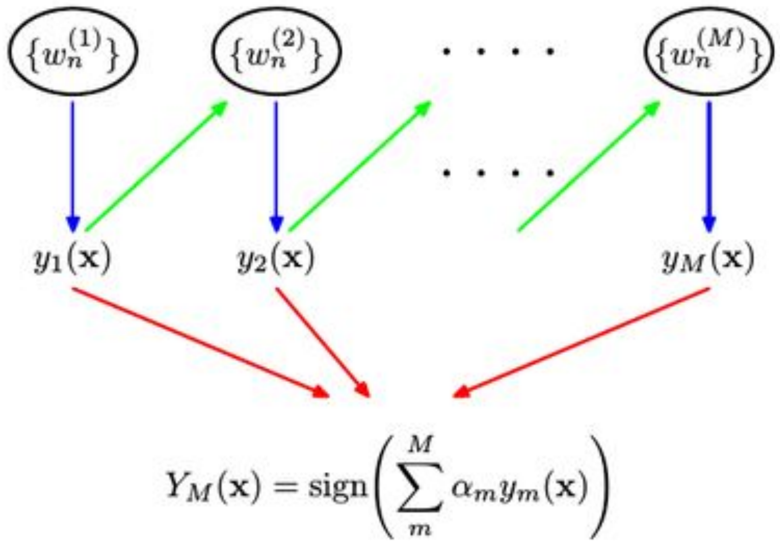
f() = 2 + 0.9= 2.9   f()= -1 - 0.9 = -1.9

Ensemble methods -> Boosting

- The base classifiers are trained in **sequence.**
- Each base classifier is trained using a weighted with a coefficient associated with each data point depending on the **performance of the previous classifiers**.
- **Misclassified points** by one of the base classifiers are given **greater weight** when used to train the next classifier in the sequence.

Ensemble methods -> Boosting

- Once all the classifiers have been trained, their predictions are then combined through a **weighted majority voting scheme**.
- The subset creation is not random and depends upon the performance of the previous models: **every new subsets** contains the elements that were **misclassified by previous models.**



Boosting -> Gradient Boosting Trees

- Boosting with CARTs as base model.
  - Add a new tree in each iteration.
  - Each tree uses information from the previous iterations.
- Represents the learning problem as gradient descent on some arbitrary differentiable loss function.

Implementation -> XGBoost

- Extreme Gradient Boosting
- Open Source
- Multiple Languages: Python, R, Julia, Scala, Spark, H20.
- Performance: Multiple CPU and GPU support
- Used in most of winner solutions for ML competitions.
- Scikit-learn interface and interaction.

```
from xgboost import XGBClassifier

xgb_clf = XGBClassifier(n_estimators=100, max_depth=7)
xgb_clf.fit(X_train, y_train)
xgb_clf.predict(X_test)
```

XGBoost: GPU Implementation

- Rory Mitchell's Master Thesis integrated to the library

- Install GPU builds and simply adding:

```
param['tree_method'] = 'gpu_hist'
```

- The tree construction algorithm is executed entirely on the GPU:
  - Advantage: Reduce the bottleneck of host/device memory transfers
  - Disadvantage: GPU has significantly lower memory than a CPU

XGBoost: GPU Implementation

machinalis

Parallel workaround:

- Parallelize approximated Split Finding at each level by features.
- Parallel Radix Sort based on Histograms and Prefix Sums
- Column Block for Parallel Learning.

Memory Efficiency:

- Bit Compression: Float64 to Float32
- Sparsity: Handle Sparse Matrix Optimizely cells containing 0 are not stored in memory.
    - The key improvement is to only visit the non-missing entries

XGBoost: GPU Implementation

machinalis

- **Benchmarks:**
  The GPU algorithm provides speedups: 3x to 6x over multicore CPUs on desktop machines.

- **Experimental improves:**
  - Multiple GPUs (experimental support).
  - DART: Dropouts meet Multiple Additive Regression Trees:
    - Trees added early are significant and trees added late are unimportant.
    - Add dropout technique from deep learning community

References

- Nvidia Parallel Forall: Gradient Boosting, Decision Trees and XGBoost with CUDA
- Mitchell R, Frank E. (2017) Accelerating the XGBoost algorithm using GPU computing.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system.
- Friedman, J. Greedy function approximation: a Gradient Boosting Machine.

¡Muchas gracias!

machinalis
Machine Learning Solutions Delivery