A Tutorial on XGBoost Machine Learning Algorithm

Jindal K. Shah (jindal.shah@okstate.edu)

School of Chemical Engineering

Oklahoma State University

September 20, 2024

About Myself

- Associate Professor in Chemical Engineering at Oklahoma State University
- Trained in MD/MC
- Started learning QM and machine learning a few years ago

i-comse.org

i-Co SE
Home About Announcements Team Workshops Resources Forums Contact
$H\Psi=E\Psi$
$S=k\ln\Omega$

The institute aims to provide basic and advanced training needs of computational molecular science and engineering community though in-person and virtual schools and well-curated training materials.

Upcoming Workshops

January 20, 2024

The 7th i-CoMSE workshop: Molecular Dynamics Summer School, Boise State University, July 8-12, 2024

http://www.i-comse.org/workshops/

6th iCoMSE workshop: Enhanced Sampling Virtual School 2024

Registration: Open

Location: Online

Workshop dates: Feb 12-16, 2024

Application deadline: Feb 2, 2024

Decision on application: Feb 7, 2024

This workshop is supported by funding from National Science Foundation Office of Advanced Cyberinfrastructure

Description: This workshop will provide an overview of modern enhanced sampling algorithms, including replica exchange, umbrella sampling, metadynamics, and path sampling methods. It will feature of mix of lectures and hands-on exercises running GROMACS on national supercomputing resources, and will run for 3 hours each day. The workshop will include a session on diversity, equity, and inclusion aspects of computational sciences. The session will run for five days, 3-6 EDT / 2-5 CDT / 1-4 MDT / 12-3 PDT.

Read more

The 5th i-CoMSE Workshop: Machine Learning for Molecular Science

Registration: Closed

Location: University of Minnesota Twin Cities

Workshop dates: July 10-14, 2023

Application deadline: May 29, 2023

Decision on application: June 10, 2023

This workshop is supported by funding from National Science Foundation Office of Advanced Cyberinfrastructure

Description: This workshop will provide an overview of machine learning tools applied to study molecular systems with a focus on computational molecular science. It will feature a mix of lectures and hands-on exercises running machine learning algorithms with molecular simulations on national supercomputing resources. The workshop will include a session on diversity, equity, and inclusion aspects of computational sciences. Sessions will be taught in Software Carpentry style, with approximately equal time divided between lectures and hands-on programming exercises.

Read more

DFT Summer School

Registration: Closed

Location: Missippi State University

Workshop dates: June 12-18, 2022

Application deadline: April 15, 2022

Decision on application: April 29, 2022"

This workshop is supported by funding from National Science Foundation

Density functional theory (DFT) has become an essential tool for modeling chemical reactions due to its relatively low computational cost and favorable scaling with the system size. This course will present a theoretical and practical introduction to computational techniques for studying chemical catalysis and kinetics. Participants will learn practical aspects of DFT calculations and advanced topics such as the effect of solvation on chemical reactions. The open-source software CP2K will be used for the hands-on tutorial sessions.

Read more

MDMC Summer School 2022

Registration: Closed

Location: Oklahoma State Universiy, Stillwater, OK

Workshop dates: July 10-15, 2022

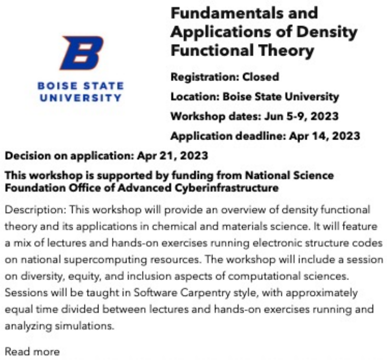
Application deadline: May 20, 2022

Decision on application: June 1, 2022

This workshop is supported by funding from National Science Foundation

Description: Monte Carlo (MC) and molecular dynamics (MD) simulation techniques have become essential tools in understanding thermophysical and phase equilibria properties of systems ranging from organic liquids to ionic liquids, polymers, biomolecules, solutions, zeolites, metal organic frameworks and covalent organic frameworks, etc. These techniques are based on statistical mechanics principles and enable one to access length scales spanning tens of nanometers and sample timescales up to hundreds of nanoseconds.

Read more



Fundamentals and Applications of Density Functional Theory

Registration: Closed

Location: Boise State University

Workshop dates: Jun 5-9, 2023

Application deadline: Apr 14, 2023

Decision on application: Apr 21, 2023

This workshop is supported by funding from National Science Foundation Office of Advanced Cyberinfrastructure

Description: This workshop will provide an overview of density functional theory and its applications in chemical and materials science. It will feature a mix of lectures and hands-on exercises running electronic structure codes on national supercomputing resources. The workshop will include a session on diversity, equity, and inclusion aspects of computational sciences. Sessions will be taught in Software Carpentry style, with approximately equal time divided between lectures and hands-on exercises running and analyzing simulations.

Read more

Enhanced Sampling Virtual School 2023

Registration: Closed

Location: Online

Workshop dates: Mar 20-24, 2023

Application deadline: Feb 24, 2023

Decision on application: Mar 6, 2023

This workshop is supported by funding from National Science Foundation Office of Advanced Cyberinfrastructure

Description: This workshop will provide an overview of modern enhanced sampling algorithms, including replica exchange, umbrella sampling, metadynamics, and path sampling methods. It will feature of mix of lectures and hands-on exercises running GROMACS on national supercomputing resources, and will run for 3 hours each day. The workshop will include a session on diversity, equity, and inclusion aspects of computational sciences. The session will run for five days, 3-6 EDT / 2-5 CDT / 1-4 MDT / 12-3 PDT.

Read more

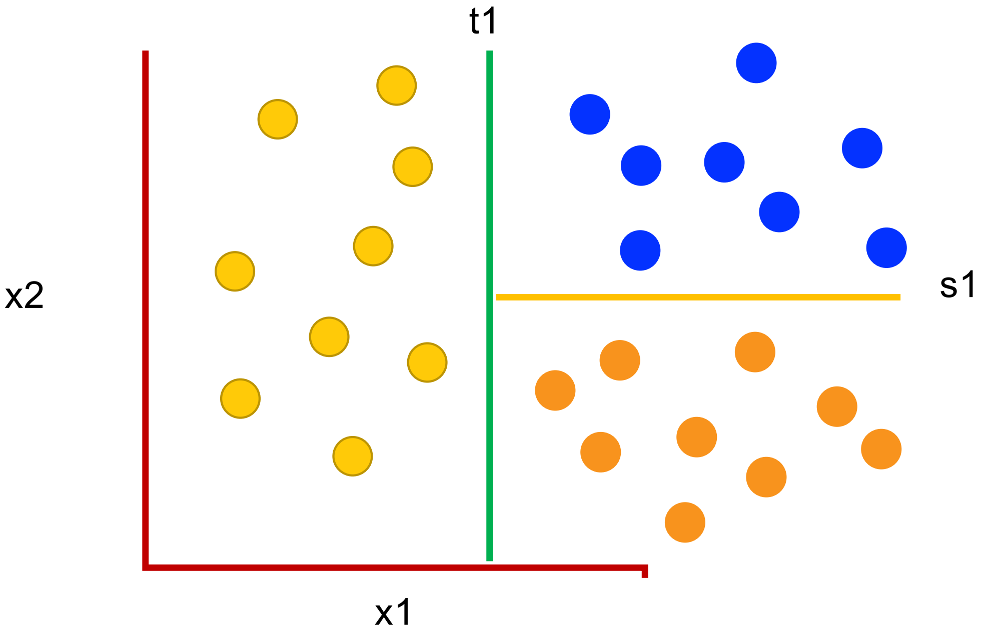
OAC-2118180 & CBET 1845143

Topics

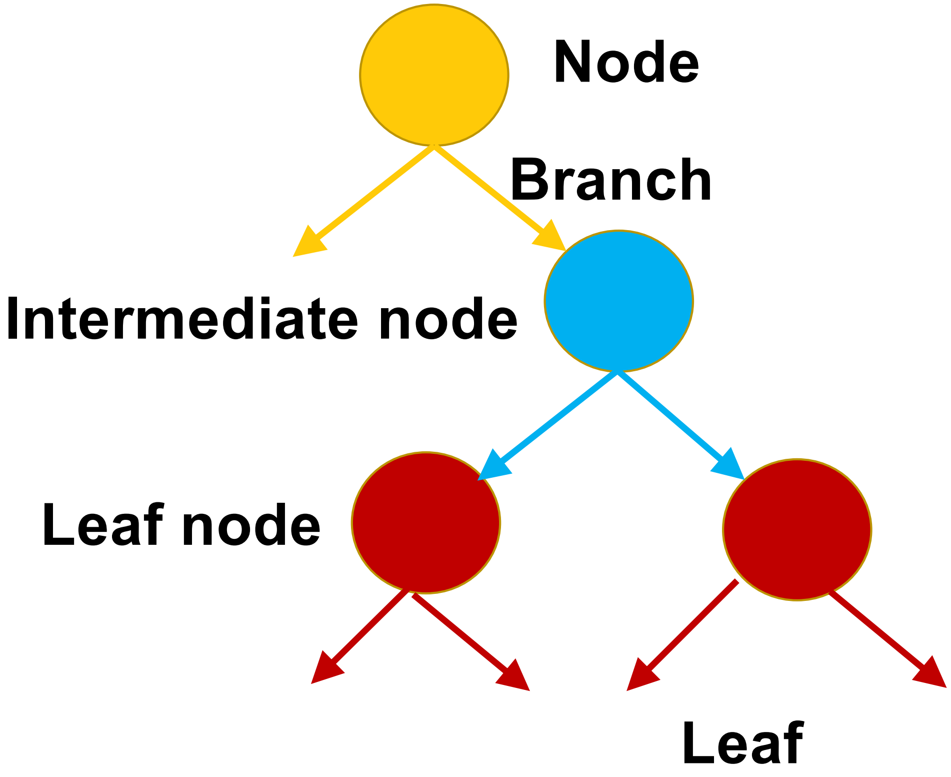
- Decision Tree
- Bagging (bootstrap + aggregating)
- Random forests
- Gradient Boosting and extreme Gradient Boosting (XGBoost)
- Hands-on Exercise

Decision Tree

- Supervised learning method
- Regression/classification
- Non-linear model
- If…then…else…
- Non-parametric model
- Piecewise continuous



Decision Tree



Depth of the tree = maximum number of branches to reach a leaf

Objective function

- The objective is to minimize the residual sum of squares

$$\sum_{j=1}^{J}\sum_{i\in R_j}(y_i-\hat{y}_{R_j})^2$$

- Here *J* represents the number of regions the feature space is partitioned into. The prediction in each of the regions is given by the average response.

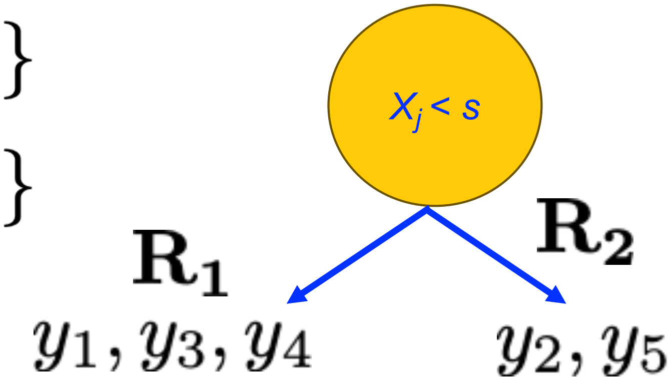
$$\hat{y}_{R_j}=\frac{\sum_{i\in R_j}y_i}{N_{R_j}}$$

Partitioning the Feature Space

- Top-down greedy approach known as *recursive binary splitting:*
  - Begins at the top of the tree and successively splits the feature space
  - Greedy because the split at a particular step minimizes the RSS at that step rather than splitting in such a way to achieve a better tree in a future step
- Consider a split over a feature *j* and the corresponding threshold value *s*, which divides the data such that

$$R_1(j, s) = \{X | X_j < s\}$$

$$R_2(j, s) = \{X | X_j \geq s\}$$



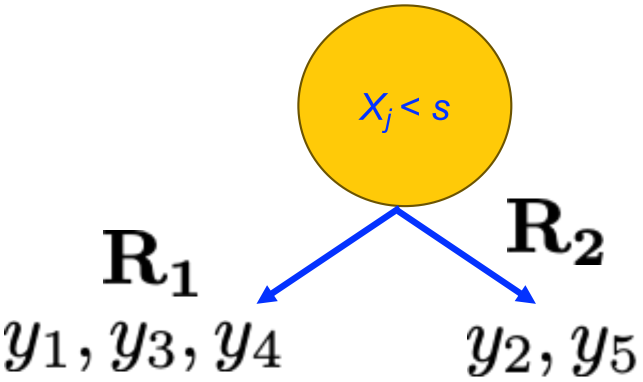
Selection of feature and its threshold

- Minimize the RSS

$$\sum_{i\in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i\in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

$$i = 1, 3, 4 \qquad i = 2, 5$$

$$\frac{y_1 + y_3 + y_4}{3} \qquad \frac{y_2 + y_5}{2}$$



(Dis)Advantages of Decision-Tree Models

- Advantages
  - Ease of interpretation
  - Graphical representation and understanding by a non-expert
  - Scaling of features is not required

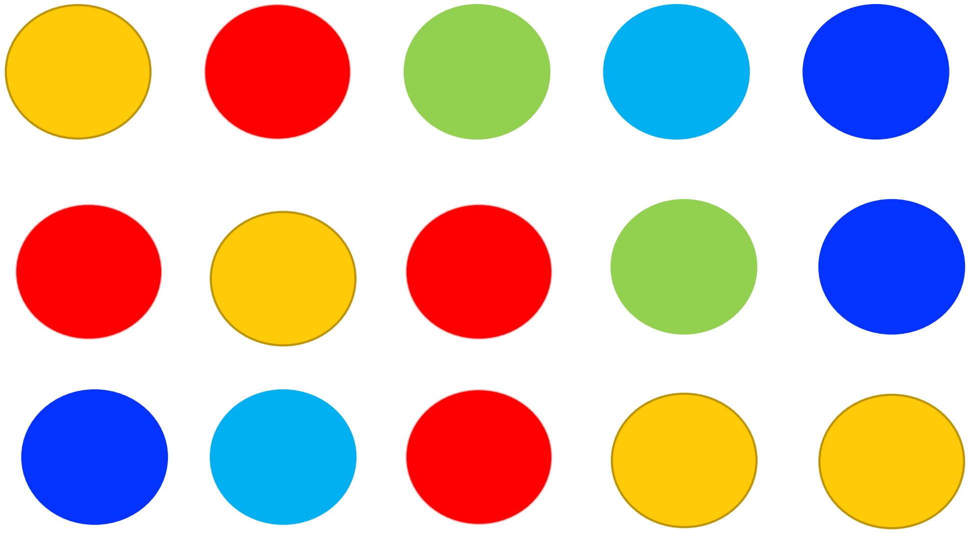
- Disadvantages
  - Accuracy is usually lower than other regression-based approaches
  - Small changes in the data can greatly impact the tree structure
  - As outputs are only piecewise continuous, multiple inputs can yield identical results.

Overcoming disadvantages

- Bagging (bootstrap + aggregating)
  - Using multiple decision-tree models
- Random forests

Bootstrap sampling

- Using the same data set, create multiple data sets by randomly drawing samples with replacement



Original Data

Bootstrapped Samples

Bootstrapped Samples

Aggregating

- For each of the bootstrapped data set *i*, develop a decision-tree model and predict a response $f_i(x)$

- Average each of the responses to obtain the response due to bagging.

$$f_{\text{bag}}(x) = \frac{\sum_i f_i(x)}{B}$$

Random Forests

- Multiple decision-tree models
- Bootstrapped data set
- Randomly selected subset of features at every split
- Achieves decorrelation of trees
- Hyperparameters:
  - Number of trees
  - Number of features to select at every split
  - Minimum number of samples required at an internal node
  - Minimum number of samples required at a leaf node

Gradient Boosting

- Borrow concept from RF but build trees sequentially
- Idea is to fit to residuals from the previous prediction
- Consider the following dataset

$$\{x_1, y_1\}, \{x_1, y_1\}, \{x_1, y_1\}, \ldots, \{x_n, y_n\}$$

- In the first step, response for each step is predicted to be the average response

$$y_i^0 = \frac{\sum y_i}{N}$$

- Residual for each of the data point is computed as

$$r_i^0 = y_i - y_i^0$$

Gradient Boosting

- A decision-tree is obtained for the residuals, which provides an estimate of the residual for the *i*th datapoint, say $\hat{r}_i^1$

- New prediction = old prediction + learning parameter * residual prediction

$$\hat{y}_i^1 = y_i^0 + \nu * \hat{r}_i^1$$

- New residual = Output – New prediction $r_i^1 = y_i - \hat{y}_i^1$

- Fit a decision-tree to r_i^1 and update predictions
- As one might imagine, the number of trees to becomes a hyperparameter

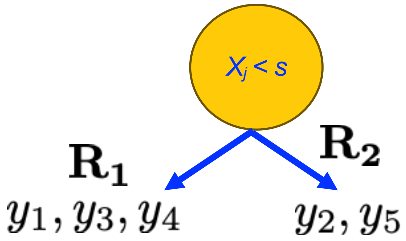
Extreme Gradient Boosting

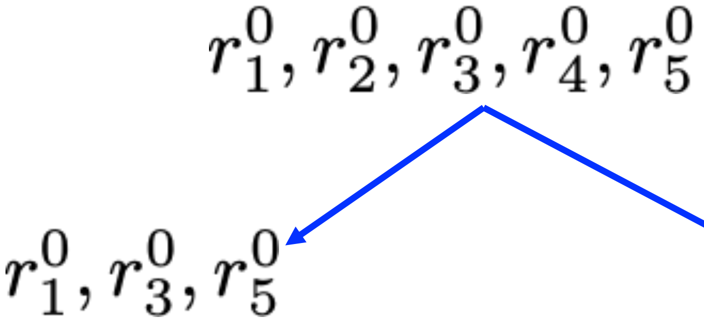
- Very similar to the gradient boosting but the split is based on similarity score and gain
- As before, compute residuals: $r_i^0 = y_i - y_i^0$
- Compute similarity score

$$\frac{\sum_i^{N_r} r_i^0}{\text{No. of residuals} + \lambda}$$

Regularization parameter

Splitting a Node in XGBoost





$$S_{\text{node}} = \frac{\left(r_1^0 + r_2^0 + r_3^0 + r_4^0 + r_5^0\right)^2}{5 + \lambda}$$

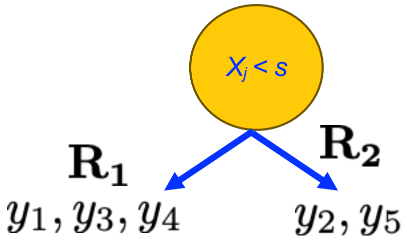
$$S_{\text{left}} = \frac{\left(r_1^0 + r_3^0 + r_5^0\right)^2}{3 + \lambda}$$

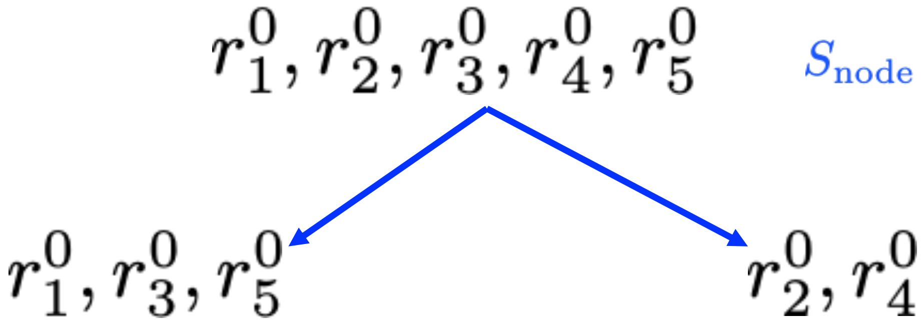
$$S_{\text{right}} = \frac{\left(r_2^0 + r_4^0\right)^2}{2 + \lambda}$$

$$\text{Gain} = S_{\text{left}} + S_{\text{right}} - S_{\text{node}}$$

Step through different values of the threshold and features; select the pair that maximizes Gain.

Output of a Leaf





$$S_{\text{node}} = \frac{\left(r_1^0 + r_2^0 + r_3^0 + r_4^0 + r_5^0\right)^2}{5+\lambda}$$

$$\text{Output}_{\text{left}} = \frac{\left(r_1^0 + r_3^0 + r_5^0\right)}{3+\lambda} \qquad \text{Output}_{\text{right}} = \frac{\left(r_2^0 + r_4^0\right)}{2+\lambda}$$

for $\lambda = 0$, output is average of the residuals

New predictions are obtained in a similar manner as that for the gradient boosting method – Slide 16

Topics Not Covered

- Mathematical formulation
- Pruning of trees
- Cross validation/hyperparameter tuning
- Classification problems
- Resources
  - https://xgboost.readthedocs.io/en/stable/index.html
  - https://youtu.be/OtD8wVaFm6E?si=541WWdCAKKrCtEIb
  - https://youtu.be/3CC4N4z3GJc?si=lj7GX4z_SAWrxqCT

Thank you!