Experiential Learning of Data Science in Engineering

Anna Hartwig[1], Shiwen Mao[2], Jin Wang[1], Q. Peter He[1,*]

[1]Dept. of Chemical Engineering, Auburn University, Auburn, AL 36849
2 Dept. of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849

Need

Data science is emerging as a field that is revolutionizing the world. Undergraduate education offers a critical link in providing more data science and engineering (DSE) exposure to students and expanding the supply of DSE talents. DSE education requires both appropriate classwork and hands-on experience with real data and real applications. While significant progress has been made in the former, one key aspect that yet to be addressed is hands-on experience incorporating real-world applications.

Guiding Question

Since there is a gap in "providing hands-on experience with real data and real applications" in DSE, while experiential learning theory (ELT) promoting "learning through experience", how can ELT guide the design and development of such learning materials?

Outcomes

- Have been developing data-enabled engineering project (DEEP) modules guided by ELT
- Course-based undergraduate research experience (CURE) provides excellent guidance for assembling DEEP modules into research projects
- DEEP modules are developed in the forms of interactive Matlab Live Scripts and Jupyter Notebooks
  - Hypothesis: these interactive development and learning environments (IDLE) will enable easy and wide adopted of the DEEP modules
- Testing DEEP modules in two courses at Auburn University in Spring of 2022
  - Chemical Engineering: CHEN 5970/6970/6976 Big Data Analytics and Machine Learning in Process Industry
  - Electrical and Computer Engineering: ELEC 5220/6220 – Information Networks and Technology
- Plan to expand the test to four courses in the Fall of 2022
- Metacognition Awareness Inventory (MAI) questionnaire is used to quantify students' metacognition awareness gains.
- After testing, plan to make DEEP modules publicly available through different channels

Sample Module

Bias Variance Trade-off

Overview

The U-shape of test MSE curves is the result of two competing properties of all statistical learning methods. For a given input x_0, the test MSE can always be decomposed into the sum of three fundamental quantities: the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$ (i.e., $\left(f(x_0)-\hat{f}(x_0)\right)^2$), and the variance of the error term ϵ. Note that $y_0 = f(x_0)+\epsilon$. The relationship is shown below.

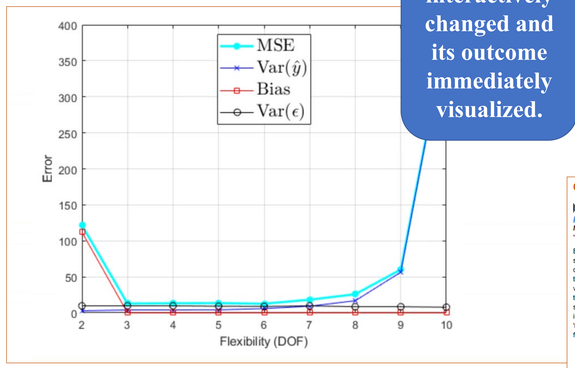
$$E\left[\left(y_0-\hat{f}(x_0)\right)^2\right] = Var\left(\hat{f}(x_0)\right) + \left[Bias\left(\hat{f}(x_0)\right)\right]^2 + Var(\epsilon)$$

Since all terms are nonnegative, test MSE can never be smaller than $Var(\epsilon)$, the irreducible error. This module is aimed to numerically demonstrate the above relationship.

Loading data

First load results from fittings using models of different flexibilities.

```
clc;
clearvars;
var_yt_all=zeros(9,50);
var_e_all=zeros(9,50);
bias2_all=zeros(9,50);
MSE_test_all=zeros(9,50);
for ii=1:9
    eval(['load poly',num2str(ii),'b.mat;']);
```



Easy integration of theory with numerical demonstration that can be interactively changed and its outcome immediately visualized.

Modules organized by Chapters and Sections, allow easy student navigation

Sample Student Project Reports

Predicting Cetane Number of Diesel Fuels

Introduction

Our group decided to use the NIR Spectra of Diesel Fuels data set for testing variables selection and calibration algorithms. We initially are using a simple linear regression in MATLAB (script attached). Our eight inputs are the boiling

Wastewater quality predictions using IR spectra from a wastewater facility

Team 1:

CHEN 6970

April 28, 2022

Literature Background and Introduction

Bearing Fault Classification

CHEN 6970

Project Report

Team Project: Analyzing the Heart Disease dataset

Class: CHEN 6970 Big Data Analytics & Machine Learning in Process Industry

A COMPARITIVE STUDY ON CLASSIFICATION MODELS FOR PREDICTION OF PARKINSON'S DISEASE BASED ON KNOWN FEATURES

CHEN6970 BIG DATA ANALYTICS - PROJECT

April 28, 2022

TABLE OF CONTENTS

S.No Content

Chapter 8 Tree-Based Methods

8.1 The Basics of Decision Trees

Example 1 construction of a regression tree using function `fitrtree`. Similar to other regression functions in *Statistics and Machine Learning Toolbox*, the "r" between "fit" and "tree" refers to "regression". For classification, we use function `fitctree`, where "c" stands for "classification".

By default, all predictors are treated as numerical variables unless it is a logical vector, unordered categorical vector, character array, string array, or cell array of character vectors. You can use `'CategoricalPredictors'` to specify categorical predictors using either column index or variable name if it's a table. You can also use `'PredictorNames'` to rename the predictors. In this example, we rename the predictors to make them short for easy visualization using function `view`. It can be seen that `view` is a very convenient tool to visualize the decision tree. After training a decision tree regression model, we can use function `predict` to predict responses for testinev samples. In this example, you can also learn how to use `'MinParentSize'` to control the depth of the tree. This parameter specifies the minimum number of observations must be included in each internal or branch node. The default value is 10. (Note that this is not the minimum number of observations in each leaf node, which can be smaller than this number.) Note that 'Parent' in `'MinParentSize'` can be understood as internal node, or brach node, or decision node, to differentiate from leaf node as shown in the following figure.

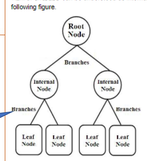


Fig. 1 Illustration of a binary decision tree. 'internal node' is also known as 'decision node', or 'parent node', or 'branch node', or 'split node'. 'leaf node' is also known as 'terminal node'.

Another note is that if you use only a subset of predictor variables in a table to fit the decision tree model, you need to provide a formula, similar to other regression models.

Example 2 Control the depth of a regression tree using `'MaxNumSplits'`.

Note that there are different ways to control the depth (complexity) of a decision tree. In Example 1, we learned `'MinParentSize'`. In this example, we learned `'MaxNumSplits'`. Another way to control tree depth is to specify `'MinLeafSize'` which determines the minimum number of observations for each leaf node. If you supply both `'MinParentSize'` and `'MinLeafSize'`, `fitrtree` uses the setting that gives larger leaves: MinParentSize = max(MinParentSize,2*MinLeafSize).

The function `fitrtree` splits branch nodes layer by layer until at least one of these events occurs:

Broader Impacts

- This project helps address a very important national need – preparing workforce talents with required data-skills to meet the demand of the current and future job market, which contribute to the NSF goal of "development of a diverse, globally competitive STEM workforce" and "increase economic competitiveness of the U.S.".
- We hope this project will serve as a model for other researchers to contribute to DEEPs development based on real data and applications from their lab experiments, research projects, and/or industrial projects.

AAAS Improving Undergraduate STEM Education Initiative

IUSE Project Title: Data-Enabled Engineering Projects for Undergraduate Data Science and Engineering Education
Award #: 1933873