New York University: Machine Learning in Economics

Lecture 1: Introduction to Machine Learning

Dr. George Lentzas

1. Introduction

2. Machine Learning

3. kNN

4. Notes

What is Machine Learning?

Key Concepts and Motivation

- Machine learning refers to a vast collection of statistical techniques for understanding and analyzing data.
- Difference between Machine Learning and Artificial Intelligence.
- The techniques come in two varieties; *supervised* and *unsupervised* learning.
- *Supervised* learning involves building a model that relates a set of inputs to an output. The familiar linear regression is an example of supervised learning.
- *Unsupervised* learning refers to techniques that deal with data that have no obvious output. Here we are interested in learning about the general structure of the inputs themselves.
- Supervised learning techniques can be further separated into two categories
  1. Regression problems: using inputs to predict a numerical output, much like in the classical linear regression case.
  2. Classification problems: using inputs to predict a categorical or qualitative output, for example a yes/no response.

Why Machine Learning in Economics?

Big Data and Motivation

- Big Data! That is data sets that are so big and/or complex that traditional statistical techniques (multiple regression) become problematic or suboptimal.
- This is both a problem and an opportunity. New datasets and new techniques allow us to re-examine some of the classic business problems.
- The combination of big data, new statistical techniques and increasing computing power has led to a recent explosion of potential interest in Machine Learning and its applications in economics. This is sill a relatively untapped area with great potential!
- Additionally recent focus on prediction; economics as a discipline under attack for not predicting the financial crisis. Can Machine Learning help?

Why Machine Learning in Economics?

What makes Machine Learning for Social Science Special?

- Classical versus Social Science Machine Learning.
- Some key differentiators:
  1. Small data? Particularly relevant for finance and macro.
  2. Time changing data generating process (non-stationarity). A problem in itself but also a driver of small data problem.
  3. Low Signal to Noise ratio (compare the picture of a cat to a stock chart). Makes classical Machine Learning methods not directly applicable.
  4. Reflexivity; cats don't know they are being classified but human behavior changes in the presence of successful Machine Learning.
  5. "Overfitting" is a major concern! Small data makes dealing with this really difficult!
- Where does this leave us? Need to understand and not apply blindly.
- Successful application of Machine Learning to Social Science is an open problem.

Introduction to R

The R Language and R Studio

- This class will make extensive use of the language R (https://www.r-project.org/). Unless you are an experienced user of R the recommended IDE is R Studio. You can install RStudio from https://www.rstudio.com/.
- A very detailed introduction to R can be found here: https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
- There are numerous great books on R, my personal favorite is "R Cookbook" by Paul Teetor which is part of the required reading in the first two weeks.
- A more advanced book is "Advanced R" by H. Wickham (http://adv-r.had.co.nz/).
- Another excellent resource is http://www.r-bloggers.com/ an aggregator of blogs about R.

A Gentle Introduction to Machine Learning

- Many ML problems can be written in the form

  $$Y = f(\mathbf{X}) + \epsilon$$

  where Y is a dependent/ response variable while $\mathbf{X}$ are the predictors/ features/ independent variables. Many ML techniques aim at estimating f, for prediction and/or inference.

- In the of prediction, the accuracy of our prediction $\hat{Y}$ depends on two quantities:

  $$\mathrm{E}\left(Y - \hat{Y}\right)^2 = \left(f(\mathbf{X}) - \hat{f}(\mathbf{X})\right)^2 + \mathrm{Var}(\epsilon)$$

- The first part is the *reducible* error; good use of ML techniques will reduce this. The second part is the *irreducible* error, essentially the variability that arises outside f and which will always provide an upper bound on the accuracy of our prediction. This bound will almost always be unknown in practical applications.

A Gentle Introduction to Machine Learning

- Inference refers to situations where we want to understand the relationship between the inputs and the outputs, how changes in in $\mathbf{X}$ change Y. Here f cannot be a black box as we need to know and understand its exact form to answer questions like, "which inputs are important" or "can f be accurately described by a linear model"? trade-off: interpretability and flexibility
- Depending on whether we are interested in prediction or inference different ML techniques might be preferable. In practice we face a trade-off between model interpretability (usually simple parametric) and flexibility (usually complex parametric or non parametric). Simple parametric models are easier to fit but heavily rely on a model assumption; complex parametric and non-parametric models can easily overfit the data and require a large number of observations for an accurate fit.
- The well known linear regression makes for good inference (simple and interpretable model) but may not produce as accurate predictions as non-linear approaches. Check out the graph below from ISLR to get an idea of where various techniques fall in this trade-off.

A Gentle Introduction to Machine Learning

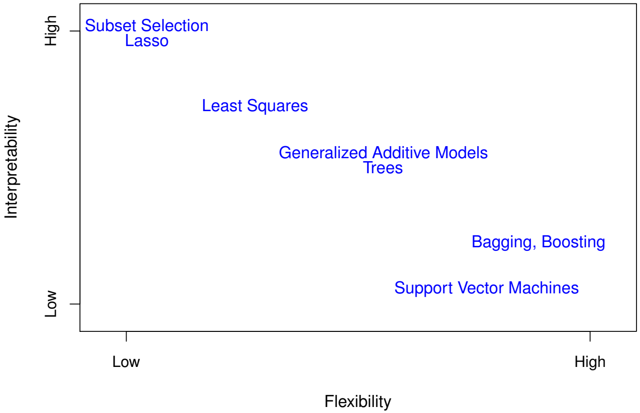

Figure: Prediction vs Inference

A Gentle Introduction to Machine Learning

- So far we have been mostly dealing with supervised learning ML. Recall that unsupervised learning deals with the less intuitive situation in which for every data point we observe a vector of measurements/ inputs x_i but no associated response/ output y_i. One ML technique appropriate for this situation is cluster analysis, which helps us figure out if the observations fall into distinct groups.

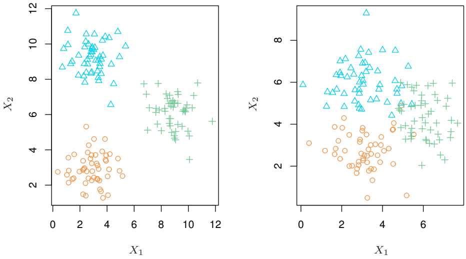

Figure: Unsupervised Learning: clustering with each group having a different color

A Gentle Introduction to Machine Learning

- In order to measure the performance of a ML method we will need to quantify how "close" the predictions are to the true outputs. The most commonly used such metric is *mean square error* (MSE), given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}\left(x_i\right)\right)^2$$

- We can calculate the MSE for our training data (training MSE) as well as for previously unseen test data (test MSE). Generally, we want to choose the ML technique that produces the lowest test MSE.
- Be careful, minimizing training MSE will not necessarily minimize test MSE as we can see below. When a chosen Machine Learning method gives a small training MSE but a large test MSE we are said to have "overfitted" the data.
- This monotone decrease in the training MSE and the U-shape in the test MSE is a fundamental property that holds across data sets and statistical methods. Estimating the minimum point in test MSE is a key issue in Machine Learning and we will come back to it when we discuss about cross-validation.

A Gentle Introduction to Machine Learning

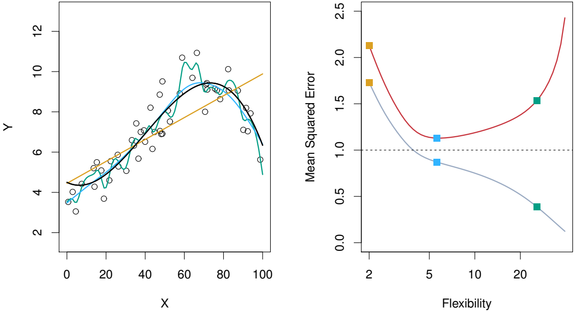

Figure: Simulated data in black with three estmates of f, orange, blue and green in increasing flexibility. Test MSE (red curve) and training MSE (grey curve). The horizontal grey line is $\mathrm{Var}(\epsilon)$ which corresponds to the lowest possible test MSE.

A Gentle Introduction to Machine Learning

- The U-shape of the test MSE is the result of two competing forces. To see this lets decompose the expected test MSE for a given value x_0 , that is the average test MSE that we could get if we estimated f over multiple training sets and each time calculated test MSE for x_0. This can be decomposed into three parts

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

  the variance of $\hat{f}\left(x_0\right)$, the squared bias of $\hat{f}\left(x_0\right)$ and the variance of the error ϵ.
- Variance captures how $\hat{f}$ would change if estimated over different training data sets. The higher the flexibility of the ML method the higher its variance,
- Bias captures the error that arises by approximating the real data generating process by a simpler model. The more flexible the ML method, the lower the bias.
- The trade-off between these two creates the familiar U-shape curve: initially, the gain from increased flexibility leading to lower bias dominates and test MSE drops but eventually the variance introduced becomes more important and test MSE starts to increase again.

A Gentle Introduction to Machine Learning

- We can see this so-called variance-bias trade-off (and how it differs depending on the data set) clearly below:

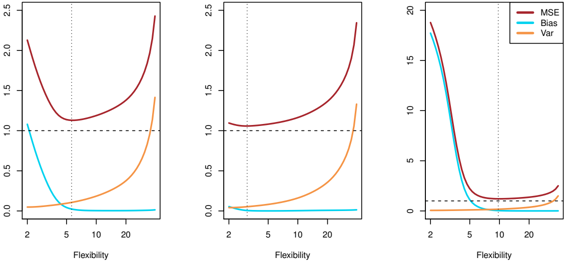

Figure: The blue curve shows the squared bias and the orrange curve the variance. Test MSE is given by the red curve and the variance of the error by the dashed horizontal curve. The three graphs correspond to three different data sets.

A Gentle Introduction to Machine Learning

Linear Regression Overview

- In this course will assume that you are familiar with multiple linear regression.
- Chapter 3 in ISLR is an excellent introduction/ overview of the topic.
- It is highly recommended that you read through Chapter 3 in ISLR as a refresher and also that you work through the R Lab section of ISLR.

Classification

Error Rates

- The most common deviation from the classical linear regression setting is the problem of classification of observations into classes. Here we re-cast the concept of MSE as the *error rate*, that is the proportion of mistaken classifications.
- As before this can be either training error rate given by

$$ER_{train} = \frac{1}{n} \sum_{i=1}^{n} I\left(y_i \quad = \hat{y}_i\right)$$

or the test error rate associated with a set of observations x_0, y_0 and given by

$$ER_{test} = Ave\left(I\left(y_0 \neq \hat{y}_0\right)\right)$$

where $\hat{y}_0$ is the predicted class label that results from applying the classifier to the test observation with predictor x_0. As before, a good classifier is the one for which the test error rate is small!

Classification

The Bayes Classifier

- We can show that the test error rate is minimized by a classifier that assigns each observation to the most likely class given its predictor values. In other words, one that assigns an observation with input values x_0 to the class j for which the conditional probability of being in a class given the input values

  $$\Pr\left(Y = j | X = x_0\right)$$

  is the largest. This is called the *Bayes classifier*.
- The Bayes classifier results in the lowest possible test error rate which is as expected called the *Bayes error rate* and which is similar to the irreducible error rate discussed earlier.

Classification

The Bayes Classifier

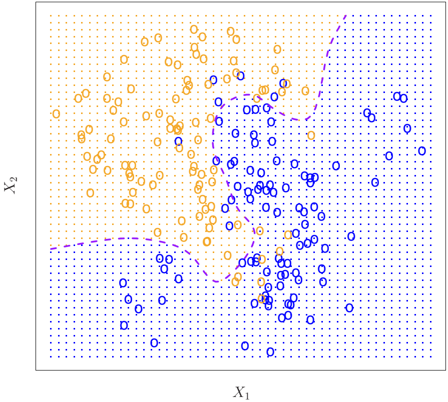

Figure: Simulated data for a two class problem. The dashed line illustrates the Bayes decision boundary.

Classification

k Nearest Neighbors

- In practice we do not know the conditional distribution of Y given X and so computing the Bayes classifier is not generally possible. Instead we use a number of ML techniques to estimate this conditional probability and then assign new observations to the class with the highest estimated probability.
- A first such ML method is the so-called k - Nearest Neighbors (kNN) classifier. The idea is simple: given a positive integer k and a test observation x_0 the kNN classifier finds the k points in the training sample that are closest to x_0 (lets call these N_0) and then estimates the conditional probability for class j as the fraction of points in N_0 whose output variable equals j. We can formalize this as:

$$\hat{\Pr}\left(Y = j | X = x_0\right) = \frac{1}{k} \sum_{i \in N_0} I\left(y_i = j\right)$$

- Then kNN applies Bayes rule and classifies each test observation x_0 to the class with the largest estimated probability.
- The word "closest" above implies a distance which is typically taken to be (standardized) Euclidean distance.

Classification

k Nearest Neighbors

- Even though this sounds simplistic, kNN often results in classifiers that are surprisingly good and close to the optimal Bayes classifier.

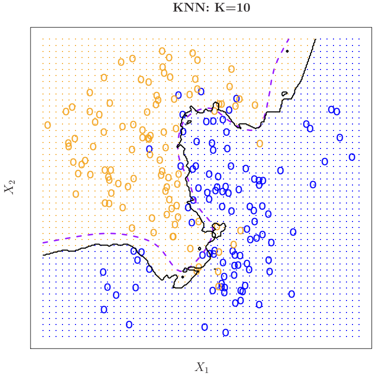

Figure: knn with k =10 on simulated data, black curve shows the kNN decision boundary and the purple line the Bayes decision boundary.

Classification

k Nearest Neighbors

- Obviously the choice of k is crucial: with $k = 1$ the decision boundary becomes noisy and overly flexible (low bias, high variance) while as k increases we reach the opposite point where the boundary becomes rigid and not flexible enough (low variance, high bias). We can see this below.

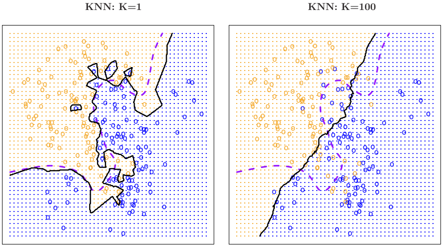

Figure: knn with k =1 (left) and k=100 (right)

Classification

k Nearest Neighbors

- In practice, *k* is usually chosen using a procedure called k-Fold Cross Validation. We will come back to this later in the seminar. For now, lets look at an example of how the choice of k affects the training and test error rates.

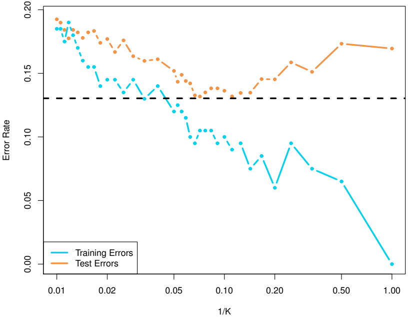

Figure: The kNN training error rate in blue and test error rate in orange.

Notes

- These notes follow closely ISLR. Also, some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani
- Some suggested homework; you do not need to submit, it is here to help you understand the material:
- ISLR Chapter 2: Exercise 8
- ISLR Chapter 5: Exercise 8
- It is also **imperative** that you work through the lab sections of the covered chapters in ISLR. This means reproducing what you see in the book on your own R session!