Machine Learning in Economics

Introduction

Alberto Cappello

Department of Economics, Boston College

Fal 2024

Lecture Goals

- What is Machine Learning

Lecture Goals

- What is Machine Learning
- Define basic language and concepts of SL/ML
  - supervised vs unsupervised learning
  - prediction vs inference
  - regression vs classification

Lecture Goals

- What is Machine Learning
- Define basic language and concepts of SL/ML
  - supervised vs unsupervised learning
  - prediction vs inference
  - regression vs classification
- Bias vs variance trade-off

What is Machine Learning?

- It is hard to come up with a working definition of Machine Learning
- It can refer to a collections of subfields of computer science
- But also it can refer to a set of topics that are developed and used across computer science, engineering, statistics, and increasingly the social sciences
- For our purposes, we will use a narrow definition of ML
- ***"Machine learning is a field that develops algorithms designed to be applied to data sets, with the main areas of focus being prediction (regression), classification, and clustering or grouping tasks"***[1]

[1] Athey, S., 2018. The impact of machine learning on economics

Supervised vs Unsupervised Learning

- These tasks can be generally divided into two main branches: **Supervised** and **Unsupervised**
- **Unsupervised Learning:** We will not be covering this type of learning in our course but it has broad applications
  - Defining feature: No outcome variable Y, just a set of predictors (features) X measured on a set of samples
  - Involves finding clusters of observations that are similar in terms of their covariates
  - These methods identify underlying patterns and predict the output
- Examples
  - Customer segmentation models group people that have similar traits for more efficient marketing and targeting campaigns
  - Recommender systems
- Methods: k-means clustering, neural networks, principal component analysis, matrix factorization

Supervised Learning

- **Supervised Learning** entails using a set of features or covariates (X) to predict an outcome (Y)
- In other words, the goal is to construct an estimator of Y as a function of X
  for eg, $E[Y|X = x] = f(x)$

Supervised Learning

- **Supervised Learning** entails using a set of features or covariates (X) to predict an outcome (Y)
- In other words, the goal is to construct an estimator of Y as a function of X
  for eg, $E[Y|X = x] = f(x)$
- Tradition ML techniques are not concerned with the form of $f(x)$. Their main aim is to achieve the best prediction of Y in an independent test set
  - Divide the whole data (X, Y) into the *training* and *test* data
  - Train an ML algorithm to form $E[Y|X = x]$ using the training data
  - Use the trained algorithm to predict $E[Y|X = x]$ in the *test* data
  - Compare the actual Y and predicted $E[Y|X = x]$ and check the accuracy of the algorithm
- No structure of the model, no interpretation and no inference

Supervised Learning

- In applied economics/econometrics, we often wish to understand an object like $f(x)$ in order to perform exercises like evaluating the impact of changing one covariate while holding others constant

Supervised Learning

- In applied economics/econometrics, we often wish to understand an object like $f(x)$ in order to perform exercises like evaluating the impact of changing one covariate while holding others constant
- In many cases the form of $f(x)$ will be informed by economic theory

Supervised Learning

- In applied economics/econometrics, we often wish to understand an object like $f(x)$ in order to perform exercises like evaluating the impact of changing one covariate while holding others constant
- In many cases the form of $f(x)$ will be informed by economic theory
- Our focus in this course will be
  - on problems which have a defined $f(x)$ and we will explore the ML methods that can be used to estimate it
  - on problems in which we are trying to figure out the "correct" $f(x)$

Supervised Learning

- Our objective is to come up with a procedure/algorithm to estimate a (population) relation f between Y and X:

$$Y = f(X) + \epsilon$$

where ϵ denotes all other factors that also determine Y, but are not available to us

Supervised Learning

- Our objective is to come up with a procedure/algorithm to estimate a (population) relation f between Y and X:

$$Y = f(X) + \epsilon$$

  where ϵ denotes all other factors that also determine Y, but are not available to us

- In *Regression problems*, Y is quantitative (wage, price, GPA, sales)
  - Examine the factors($X - age, educationlevel, experience$) and how they relate to wages(Y)

Supervised Learning

- Our objective is to come up with a procedure/algorithm to estimate a (population) relation f between Y and X:

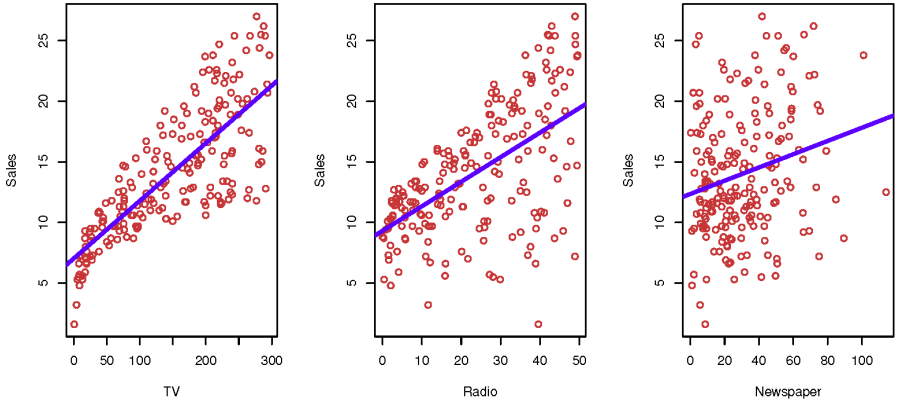
$$Y = f(X) + \epsilon$$

  where ϵ denotes all other factors that also determine Y, but are not available to us

- In *Regression problems*, Y is quantitative (wage, price, GPA, sales)
  - Examine the factors($X - age, educationlevel, experience$) and how they relate to wages(Y)
- In *Classification problems*, Y is either binary or takes a finite set of values
  - Will the stock market go up or down(Y) based on macro variables and past trends(X)?

Motivation

`Sales` vs advertising budgets on `TV`, `Radio` and `Newspaper`



How can we build a model of sales based on this data?

Notation

- `Sales` is our outcome variable Y
- `TV`, `Radio` and `Newspaper` form our vector of predictors $X = (X_1, X_2, X_3)$

Notation

- `Sales` is our outcome variable Y
- `TV`, `Radio` and `Newspaper` form our vector of predictors $X = (X_1, X_2, X_3)$
- We assume there exists a model that captures the relation between Y and X in a form of

$$Y = f(X) + \epsilon$$

- $f(X)$ is an unknown function of X and represents the information X provides about Y
- ϵ here captures everything else that affects `Sales`, but is not `TV`, `Radio` or `Newspaper` — other social and economic features, measurement errors, random deviations, etc.

Why do we need $f(X)$?

- **Prediction:** With a "good" $\hat{f}(X)$ we can make "ceteris paribus" predictions of $Y(\hat{Y})$ for a given value of $X = x$
  - Which value of `Sales` we can expect next year given planned budget allocations for `TV`, `Radio` or `Newspaper`).

Why do we need $f(X)$?

- **Prediction:** With a "good" $\hat{f}(X)$ we can make "ceteris paribus" predictions of $Y(\hat{Y})$ for a given value of $X = x$
  - Which value of `Sales` we can expect next year given planned budget allocations for `TV`, `Radio` or `Newspaper`).
- **Inference:** We can also assess which elements in X are important in explaining Y and which are irrelevant
  - `Newspaper` seems to have little impact

Why do we need $f(X)$?

- **Prediction:** With a "good" $\hat{f}(X)$ we can make "ceteris paribus" predictions of $Y(\hat{Y})$ for a given value of $X = x$
  - Which value of `Sales` we can expect next year given planned budget allocations for `TV`, `Radio` or `Newspaper`).
- **Inference:** We can also assess which elements in X are important in explaining Y and which are irrelevant
  - `Newspaper` seems to have little impact
- Depending on the complexity of $f(X)$, we may be able to evaluate the causal effect of each component of X on Y
  - Does \$1 spent on `TV` ads increase `Sales` as much as \$1 spent on `Radio` ads?

Examples

- College graduation rate
  - Prediction: graduation probability for a student based on application data
  - Inference: factors that affect graduation rate the most

Examples

- College graduation rate
  - Prediction: graduation probability for a student based on application data
  - Inference: factors that affect graduation rate the most
- Personal loans
  - Prediction: bankruptcy/default probability for a loan based on applicant's data
  - Inference: social and economics factors that have highest impact

Examples

- College graduation rate
  - Prediction: graduation probability for a student based on application data
  - Inference: factors that affect graduation rate the most
- Personal loans
  - Prediction: bankruptcy/default probability for a loan based on applicant's data
  - Inference: social and economics factors that have highest impact
- Online shopping
  - Prediction: show consumers "You may also like" items
  - Inference: which features make best selling products

How to estimate $f(X)$

- For any estimate $\hat{f}(X)$ of the true $f(X)$

$$\mathbb{E}\left[\left(Y - \hat{f}(X)\right)^2\right] = \underbrace{\left(f(x) - \hat{f}(x)\right)^2}_{\text{reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{irreducible}}$$

How to estimate $f(X)$

- For any estimate $\hat{f}(X)$ of the true $f(X)$

$$\mathbb{E}\left[\left(Y - \hat{f}(X)\right)^2\right] = \underbrace{\left(f(x) - \hat{f}(x)\right)^2}_{\text{reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{irreducible}}$$

- Our goal is to minimize the reducible error, and try to get as close as possible to the true data generating process

How to estimate $f(X)$

- For any estimate $\hat{f}(X)$ of the true $f(X)$

$$\mathbb{E}\left[\left(Y - \hat{f}(X)\right)^2\right] = \underbrace{\left(f(x) - \hat{f}(x)\right)^2}_{\text{reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{irreducible}}$$

- Our goal is to minimize the reducible error, and try to get as close as possible to the true data generating process
- The irreducible error arises becasue Y is also a function of ϵ which by definition cannot be predicted (if it is random) or cannot be predicted using the data at hand

Trade-offs in estimating $f(X)$

- Flexibility vs interpretability.
  - Linear functions are easy to interpret, splines (piecewise polynomials) or random forests are harder, neural networks are uninterpretable.

Trade-offs in estimating $f(X)$

- Flexibility vs interpretability.
  - Linear functions are easy to interpret, splines (piecewise polynomials) or random forests are harder, neural networks are uninterpretable.
- Good fit vs overfitting
  - In-sample vs out-of-sample accuracy.

Trade-offs in estimating $f(X)$

- Flexibility vs interpretability.
  - Linear functions are easy to interpret, splines (piecewise polynomials) or random forests are harder, neural networks are uninterpretable.
- Good fit vs overfitting
  - In-sample vs out-of-sample accuracy.
- Parsimony vs black-box
  - In Economics we often prefer a model that focuses on a few key variables (regression with a fixed set of variables) vs the one that throws in everything including the kitchen sink (neural network).

Overfitting

- When your model is flexible enough (e.g. higher order polynomials), it can follow the data very closely, too closely. When new data comes in, model's fit there maybe much less precise. This is known as *overfitting*.

Overfitting

- When your model is flexible enough (e.g. higher order polynomials), it can follow the data very closely, too closely. When new data comes in, model's fit there maybe much less precise. This is known as *overfitting*.
  - Intuitively this might be happening becasue if the model is trying to fit every data point, the model might be picking up associations due to random chance which don't exist in the new data

Overfitting

- When your model is flexible enough (e.g. higher order polynomials), it can follow the data very closely, too closely. When new data comes in, model's fit there maybe much less precise. This is known as *overfitting*.
  - Intuitively this might be happening becasue if the model is trying to fit every data point, the model might be picking up associations due to random chance which don't exist in the new data
- To avoid overfitting, we split our data into a *training* part `Tr`, which we use to estimate our model, and a *test* part `Te`, which we use to validate our estimate's performance.
- Ideally test data should be a new sample from the sample population, but more often we use independent draws from the same sample to split it between train and test.

Overfitting

- The basic idea is then to use the training data to estimate the model for various levels of flexibility, using only the training data:

$$MSE_{Tr} = \frac{1}{n_{Tr}} \sum_{i \in \text{Tr}} \left[y_i - \hat{f}(x_i) \right]^2$$

Overfitting

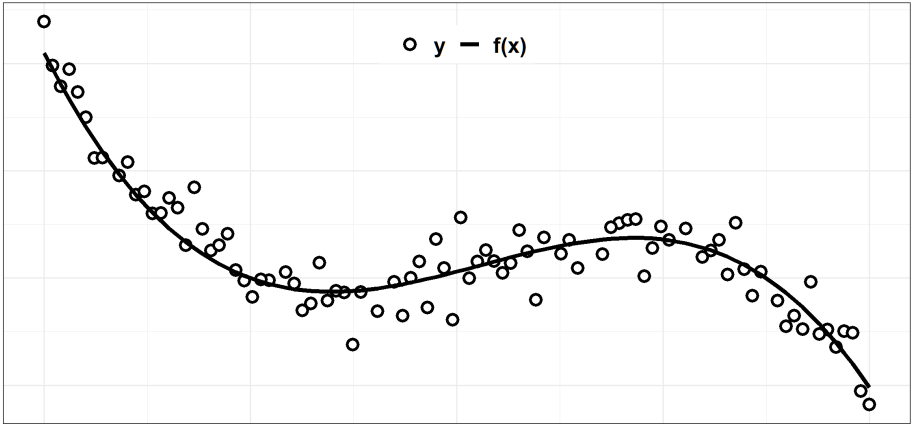
- The basic idea is then to use the training data to estimate the model for various levels of flexibility, using only the training data:

$$MSE_{Tr} = \frac{1}{n_{Tr}} \sum_{i \in \text{Tr}} \left[y_i - \hat{f}(x_i) \right]^2$$

- Then, in a second step, let the test data decide what the optimal model flexibility should be:

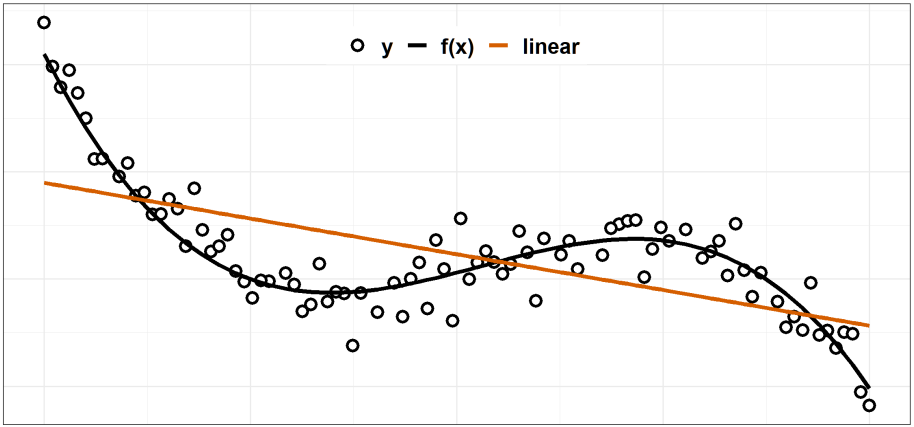
$$MSE_{Te} = \frac{1}{n_{Te}} \sum_{i \in \text{Te}} \left[y_i - \hat{f}(x_i) \right]^2$$

Overfitting



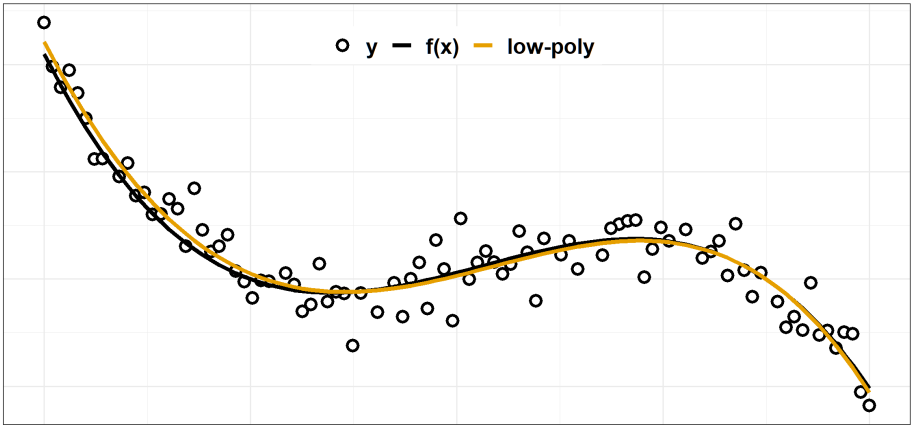
Black line is true $f(X)$, black circles are observed train data values of $y = f(x) + \epsilon$.

Overfitting



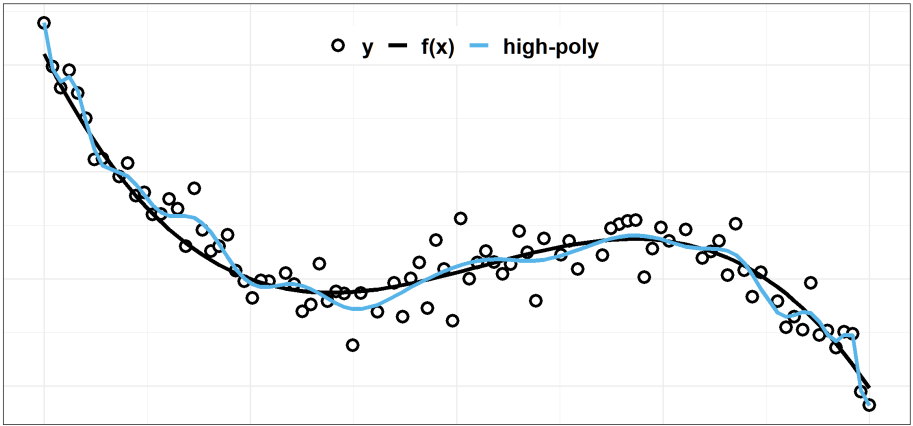
Even though true $f(X)$ is very smooth, a linear bit is clearly not flexible enough – it fails to follow both $f(X)$ and actual data points, but it does have a clear interpretation.

Overfitting



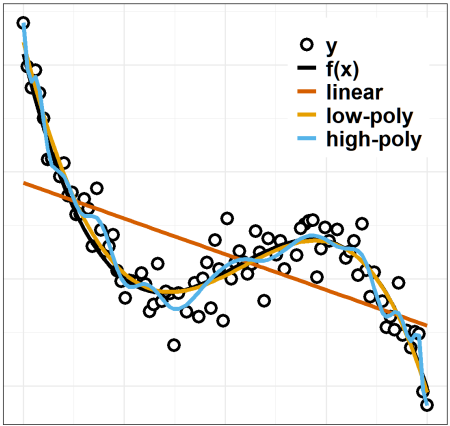
A more flexible low-order polynomial fit follows true $f(X)$ very closely, but not so much actual data points, yet providing somewhat clear interpretation.

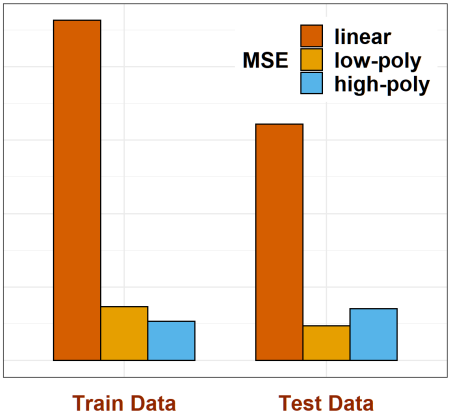
Overfitting



Very flexible high-order polynomial fit follows data points closely, but is too wiggly and thus deviates a lot from true $f(X)$. There is no clear interpretation for this fitted model.

Overfitting





Small training MSE + Large test MSE $\rightarrow$ Overfitting

Bias-variance trade-off

- Suppose our test data Te consists of a single data point (x_0, y_0). Then

$$\mathbb{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = \text{Var}\left(\hat{f}(x_0)\right) + \text{Bias}^2\left(\hat{f}(x_0)\right) + \text{Var}(\epsilon_0)$$

- The left hand side is the expected test MSE at x_0
- Variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set
- Bias refers to the error that is introduced by approximating a real-life problem by a much simpler model

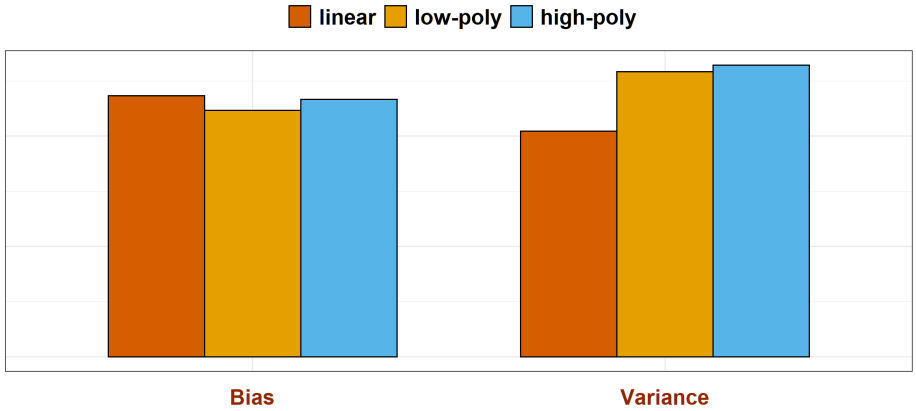
Bias-variance trade-off

- Suppose our test data Te consists of a single data point (x_0, y_0). Then

$$\mathbb{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] = \text{Var}\left(\hat{f}(x_0)\right) + \text{Bias}^2\left(\hat{f}(x_0)\right) + \text{Var}(\epsilon_0)$$

- The left hand side is the expected test MSE at x_0
- Variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set
- Bias refers to the error that is introduced by approximating a real-life problem by a much simpler model
- Typically as the flexibility of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the flexibility based on MSE amounts to a bias-variance trade-off.

Bias-variance trade-off



Linear fit has lowest variance, but highest bias. High-order polynomial fit has highest variance, lower-order polynomial has slightly lower variance and slightly lower bias, winning overall.

Bias-variance trade-off

- In the previous example low-order polynomial model won because it was close in nature to the the true $f(X)$ — flexible, but not too wiggly. That may change if the true nature of $f(X)$ changes to very flexible or very rigid.
- Since in real life scenarios we do not know true $f(X)$, using test/train data splits is the only way to avoid overfitting and find a model that is optimal in terms of bias-variance trade-off.
- Important: any optimal model will likely remain so only for a short period of time and/or for a similar dataset. And major changes in the environment will necessitate re-evaluation of all models (e.g. spike in online shopping following COVID-19 lockdowns).