[W207] Intro to Statistical Learning Theory

Professor Jason Anastasopoulos
ljanastas@princeton.edu

Princeton University

January 17, 2018

For today...

- What is statistical learning?
- Statistics in social science – causality.
- Statistics in machine learning – prediction.
- Reducible and irreducible errors.
- Estimating f.
- Accuracy v. interpretability.
- Model accuracy.
- The bias-variance tradeoff.
- Classification.

Statistical learning theory

- *Input variables* – $\mathcal{X}$.
  - AKA features, independent variables, predictors, etc..
- *Output variables.* – $\mathcal{Y}$.
  - AKA dependent variables, outcomes, etc.
- Eg) Advertising expenditures.

Statistical learning theory

$$f : \mathcal{X} \to \mathcal{Y}$$
$$\mathcal{X} \in \mathbb{R}^{nxp}; \mathcal{Y} \in \mathbb{R}^{p}$$

- Abstractly is find a function f that accurately maps the inputs $\mathcal{X}$ to outputs $\mathcal{Y}$

Statistical learning theory

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

- More concretely, we are interested in finding a function $f(X)$ which can return values of an output Y.
- In introduction to regression courses, this is typically the equation you see.
- $f(X)$ is an unknown function of a matrix of predictors $X = (X_1, \cdots, X_p)$, an outcome Y and an error term ϵ.

Approach in social science

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

- While X and Y are known, $f(\cdot)$ is unknown.
- The goal of statistical learning, then, is to utilize a set of approaches to estimate the "best" $f(\cdot)$ for the problem at hand.

Statistical learning theory

$$f(X) = \sum_{i=1}^{p} \beta_i x_i$$

$$\epsilon \sim N(0, \sigma^2)$$

$$Y = \sum_{i=1}^{p} \beta_i x_i + \epsilon$$

- In social science, we often choose a linear function to estimate Y and assume that the error term is normally distributed with a zero mean.
- Parameters β are estimated by minimizing the sum of squared errors which form the normal equations $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$

Approach in social science: causality

$$Y = \beta_0 + \beta_1 T + \sum_{i=i}^{p-1} \beta_i x_i + \epsilon$$

- Often we are interested in the values of one or two parameters and whether they are *causal* or not.
- There are many interpretations of statistical causality (ie Pearl (2009), Rubin (1974)).
- The general idea is that β_1 measures the extent to which ΔX_t will affect ΔY_{t+1}.

Approach in social science: causality

$$Y = \beta_0 + \beta_1 T + \sum_{i=i}^{p-1} \beta_i x_i + \epsilon$$

- Causal inference requires that $T \perp \epsilon$ or $T|X \perp \epsilon$.
- This often requires *randomization* of T under most circumstances.
- This implies that we are not really all that interested in choosing an optimal $f(\cdot)$.

Approach in social science: causality

$$\text{Choose design: } \delta \subset \Delta$$
$$\text{s.t.: } \exists x_i \in \mathbf{X}$$
$$\text{satisfying: } x_i \perp \epsilon$$

- Choose a subset of research designs δ from all possible designs Δ so that you have at least one treatment (variable) that is randomized.

Approach in machine learning: prediction

$$\hat{Y} = \hat{f}(X)$$

- Machine learning is primarily concerned with prediction.
- We are interested in finding the "best" $f(\cdot)$ and the "best" set of X's which give the best predictions, $\hat{Y}$.
- We want to find the function that minimize the difference between the *predicted* values and the *observed* values.

Reducible and irreducible error

$$\hat{f}(X) = \hat{Y} \qquad \text{estimated function}$$
$$f(X) + \epsilon = Y \qquad \text{true function}$$

- Prediction of Y with $\hat{Y}$ can be broken down into two components: reducible and irreducible error.
- **reducible error** – $\hat{f}$ is used to estimate f but is not perfect. Improving the accuracy of $\hat{f}$ can be accomplished by adding more *observed* features (variables) to the model.
- **irreducible error** – ϵ represents all other features that can be used predict f. These are unobserved and thus are irreducible.

Reducible and irreducible error

$$\begin{aligned}\mathbb{E}(Y - \hat{Y})^2 &= \mathbb{E}[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \mathbb{E}[(f(X) + \epsilon - \hat{f}(X))(f(X) + \epsilon - \hat{f}(X))] \\ &= [f(X) - \hat{f}(X)]^2 + Var(\epsilon)\end{aligned}$$

Estimating *f*

- **Training data** – is required to "teach" our machine learning algorithm to predict outcomes.
- *Predicting presidential elections*
  - **outcome**/**Response**- presidential candidate vote share in each state for the Republican candidate.
  - **features** – state Republican vote share in last election, ?.

Estimating f – example 1 – predicting elections

$$\text{Training data: } \{(x_1, y_1),(x_2, y_2), \cdots, (x_n, y_n)\}$$
$$x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^T$$

- $n = 50$ states.
- $i = 1, \cdots, n$ observations (states), $j = 1, \cdots, p$ features (state-level variables).
- Training data: feature (or feature set) x_{ip} and outcome y_i (election results).

Estimating f – example 2 – political sentiment in Tweets

$$\text{Training data: } \{(x_1, y_1),(x_2, y_2), \cdots, (x_n, y_n)\}$$
$$x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^T$$

- $n = 1,000$ Tweets.
- $i = 1, \cdots, n$ observations (Tweets), $j = 1, \cdots, p$ features (words, tweet length, etc).
- Training data: feature (or feature set) x_{ip} and outcome y_i (pro/anti Trump).

Estimating f – parametric methods

$$\text{Step 1 – Functional form: } f(X) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

$$\text{Step 2 – Training: } Y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

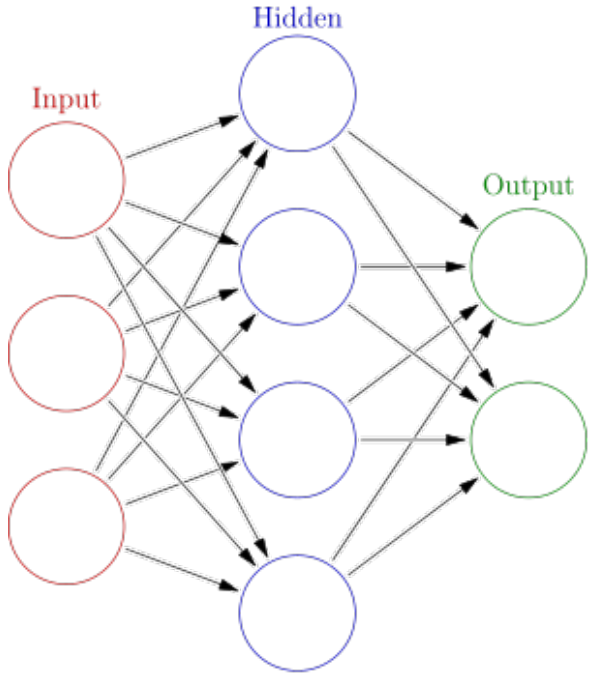
- *parametric methods* are model-based approaches that involve two steps.
- **step 1** involves choosing a predefined functional form. Linear, quadratic, etc.
- **step 2** involves *training* or fitting the model using the training data.

Estimating f – parametric methods – issues

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i=1}^{p} + \beta_i x_i^2 + \sum_{i=1}^{p} \beta_i x_i^3 + \cdots$$

- Rigid models such as a strictly linear model may not fit the data well.
- More flexible models require more parameter estimation and may result in **overfitting** – a model that is only useful for the training data at hand.

Estimating f – parametric methods – examples

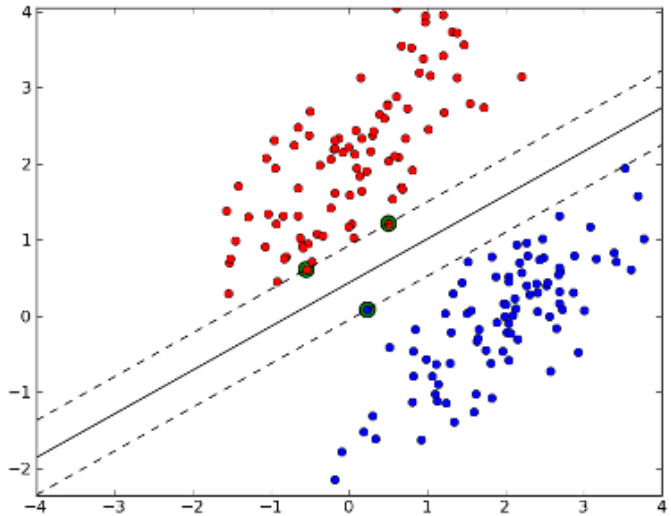


- Linear regression.
- Logistic regression.
- Naive bayes.
- Neural networks.

Estimating f – non-parametric methods

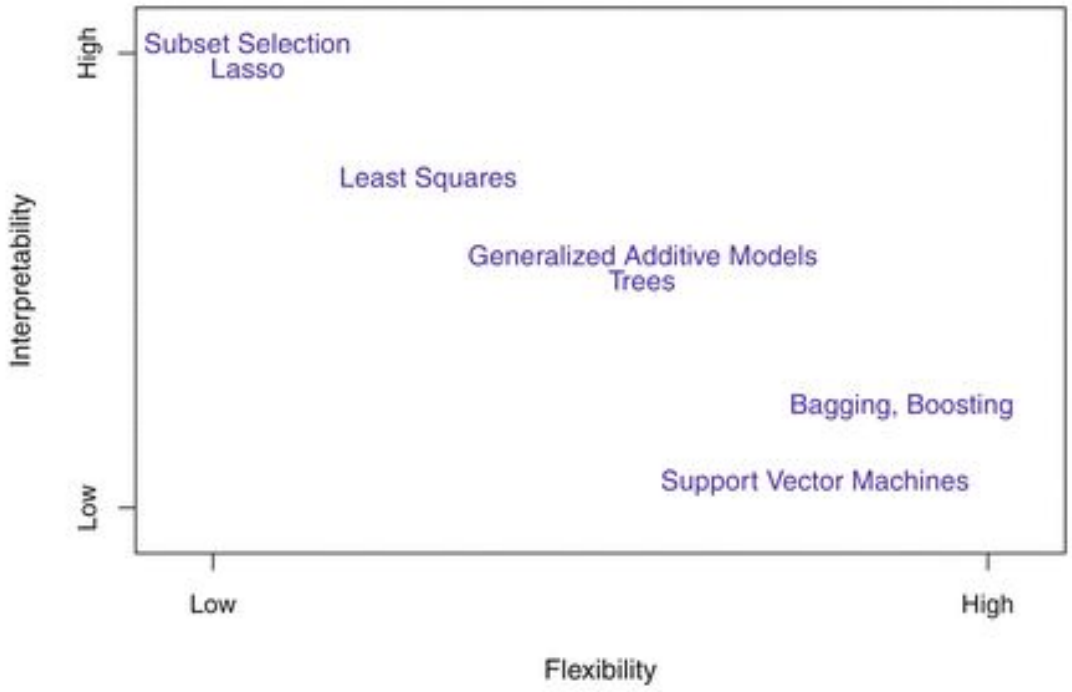
- **non-parametric** methods do not assume anything about the functional form of f.
- Estimates a function only based on the data itself.

Estimating f – non-parametric methods – examples



- K-Nearest Neighbors.
- Support vector machines.
- Decision trees.

Accuracy and interpretability tradeoffs



- More accurate models often require estimating more parameters and/or having more flexible models.
- More models that are better at prediction generally are less interpretable.

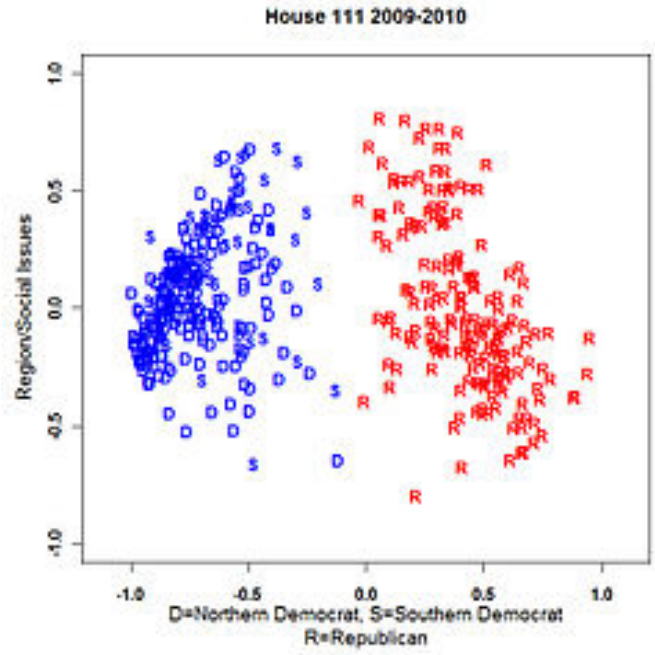
Supervised v. unsupervised learning

- **Supervised learning** involves estimating functions with known observation and outcome data.
- **Unsupervised learning** involves estimating functions without the aid of outcome data.

Supervised learning – examples

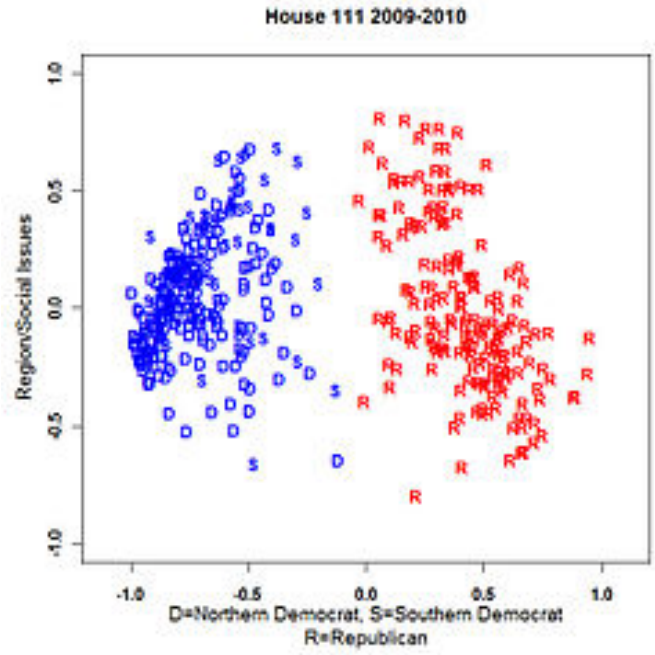
- Naive bayes.
- Support vector machines.
- Neural networks.
- Linear regression.

Unsupervised learning – examples



- Topic models.
- K-Means clustering.
- Multidimensional scaling.
- Pagerank.

Unsupervised learning – examples



- Topic models.
- K-Means clustering.
- Multidimensional scaling.
- Pagerank.

Assessing model accuracy

- Machine learning is as much an art as it is a science.
- There is not best method, only a method that best fits a problem.

Measuring fit

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

- In the regression setting, the mean squared error is a metric of how well a model fits the data.
- To estimate model fit we need to partition the data:
  1. Training set – data that we will use to fit the model.
  2. Test set – data that we will use to test the fit of the model.

Measuring fit

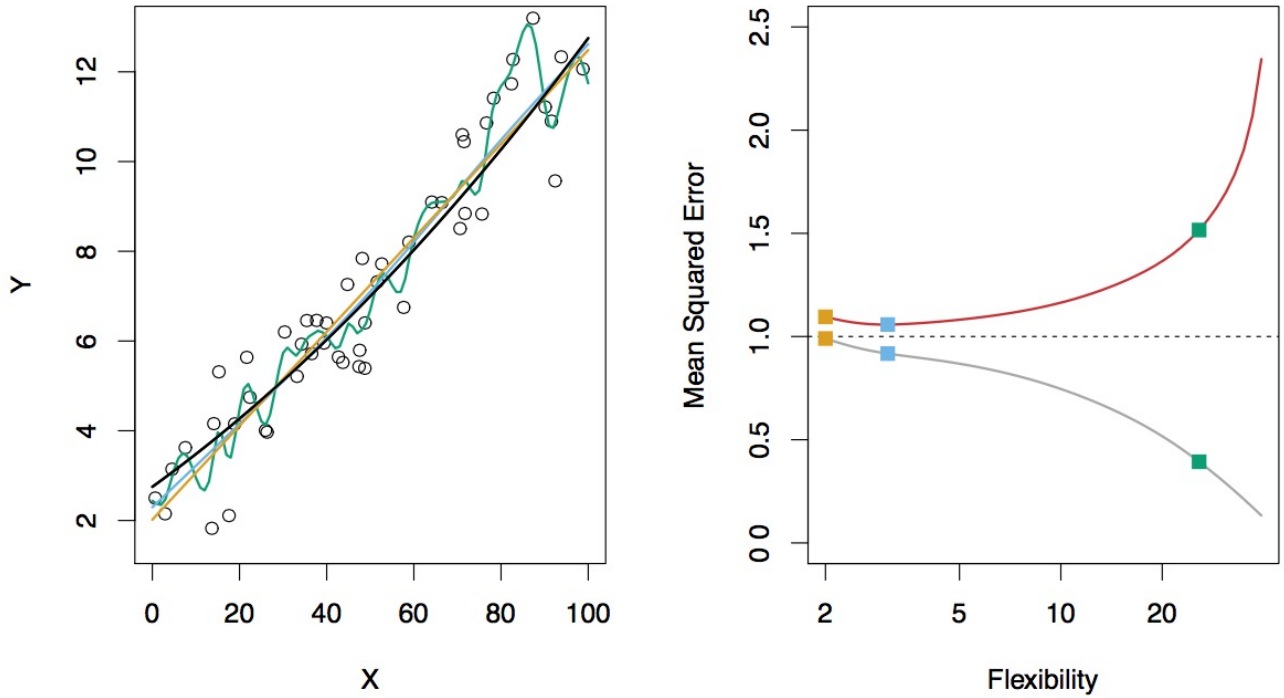
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

- **Training MSE** tells us how well our model fits the training data.
- **Test MSE** tells us how well our model fits new data.
- We are most concerned in minimizing *test MSE*.

How to choose training and test set?

- Divide labeled data randomly into two parts: training and test sets.
- **Cross-validation** involves randomly dividing the data into training and test sets several times and assessing the *average* model fit across each test set.

Training MSE, test MSE and model flexibility



- Increasing model flexibility tends to *decrease* training MSE but will eventually *increase* test MSE.

The bias-variance tradeoff

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

- It can be shown that the expected value for the test MSE can be decomposed into 3 components:
  1. $Var(\hat{f}(x_0))$ – Variance of the predictions.
  2. $[\text{Bias}(\hat{f}(x_0)]^2$ – Bias of the predictions.
  3. $Var(\epsilon)$ – Variance of the error terms.

The bias-variance tradeoff

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

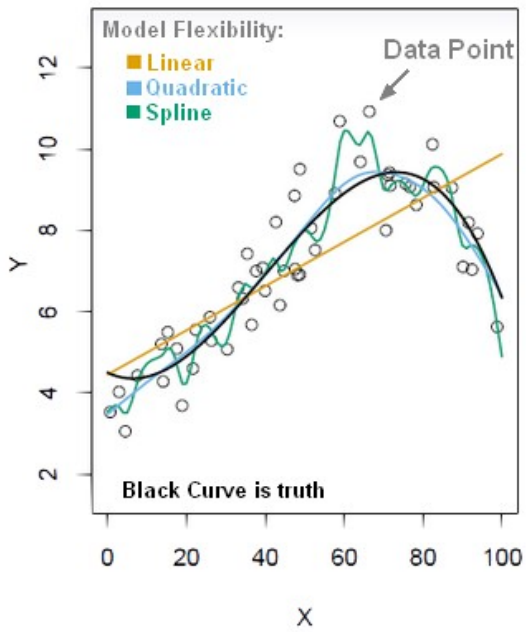
- It can be shown that the expected value for the test MSE can be decomposed into 3 components:
  1. $Var(\hat{f}(x_0))$ – how much would $\hat{f}$ change if we applied it to a different data set.
  2. $[\text{Bias}(\hat{f}(x_0)]^2$ – how well does the model fit the data?

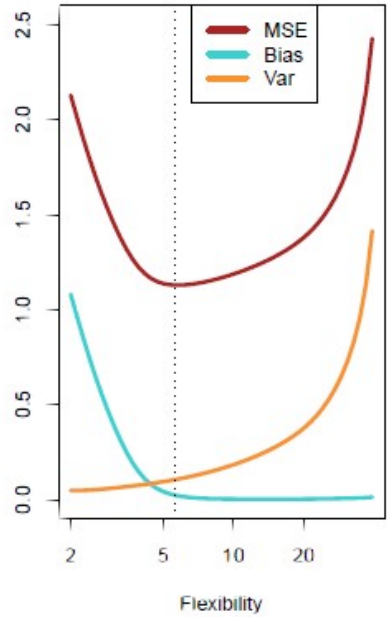
The bias-variance tradeoff

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

- It can be shown that the expected value for the test MSE can be decomposed into 3 components:
  1. $Var(\hat{f}(x_0))$ – how much would $\hat{f}$ change if we applied it to a different data set.
  2. $[\text{Bias}(\hat{f}(x_0)]^2$ – how well does the model fit the data?

The bias-variance tradeoff





- Simple models give consistent results across test sets (low variance) but don't predict well. (high bias).
- Very flexible (complex) models give inconsistent results across test sets (high variance), but do well at prediction (low bias).

Classification

$$\text{Error rate: } \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_i \neq \hat{y}_i)$$

- Our discussion of MSE previously was in the context of *regression* in which the outcome was a continuous predictor.
- There are some slight modifications that can be made in the setting in which we're interested in prediction *classes:*.
- $\{Democrat, Republican\}$, $\{Violent, Nonviolent\}$, $\{Protest, Non-protest\}$

Classification

$$\text{Error rate: } \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_i \neq \hat{y}_i)$$

- We are essentially interested in what % of classifications are correct.

Bayes classifier

$$\mathbb{P}(Y = j | X = x_0) \tag{1}$$

- It can be shown that the error rate is minimized by a classifier that assigns each observation a classification based on its predictor value.

Bayes classifier – text analysis context

$$\mathbb{P}(Y = \textit{Happy} | X = \{\textit{depressed}, \textit{miserable}\}) = 0.1 \qquad (2)$$

- Bayes classifiers are used very frequently in text analysis to predict the class of a document given words and other features.

Bayes classifier – text analysis context – classification

$$\mathbb{P}(Y = \textit{Happy} | X = \{\textit{depressed}, \textit{miserable}\}) = 0.1$$
$$\mathbb{P}(Y = \textit{Sad} | X = \{\textit{depressed}, \textit{miserable}\}) = 0.9$$
$$\arg\max_j \ \mathbb{P}(Y = j | X = x_0)$$

- Classification proceeds by choosing the class with the highest probability.
- In this case *Sad*.

Bayes classifier – text analysis context – Bayes error

$$1 - \mathbb{E}\left(\arg\max_j \ \mathbb{P}(Y = j|X = x_0)\right)$$

- If we had two observations in which $\mathbb{P}(Y = \textit{Sad}|X) = .9)$ and $\mathbb{P}(Y = \textit{Happy}|X) = .6)$.
- The Bayes error rate is: $\frac{0.1+0.4}{2} = 0.25$.