# Nonparametric Estimation: A Brief Overview

Alexander Torgovitsky

ECON 31720: Applied Microeconometrics
University of Chicago, Fall 2020

# Outline



1. **Motivation and Overview**
2. Saturated Linear Regressions
3. Kernel Smoothing, Local Polynomial and Nearest Neighbors
4. Sieves
5. Selecting Tuning Parameters
6. Machine Learning
7. Summary

# The Nonparametric Ideal

## Taxonomy

- Statistical models involve distributions, and thus functions
- A **parametric assumption** restricts the shape of a function
- For example, $f(x) = \alpha + \beta x$ restricts the shape to be linear
- Most models we use in economics are *semi*parametric

could be inf dimen

$Y = \alpha + \beta X + u$

→ e.g. linear regressions do not parameterize distribution of residual

logit is fully parametric

## Nonparametric identification

- We will often focus on **nonparametric identification** in this course

→ Identification analysis that does not involve parametric assumptions

- Popular because parametric assumptions can be hard to motivate/defend
- A bit constraining though — how do you extrapolate nonparametrically?
- Nonparametric identification leads to **nonparametric estimation**

$E[aY+bX|Z=z] = aE[Y|Z=z] + bE[X|Z=z]$

linear

Non-parametric is cleaner in a way

## The problem

- Want to estimate $\mu(x) \equiv \mathbb{E}[Y|X = x]$ with $Y$ and $X$ observed
- Nonparametric: without finitely-parameterizing $\mu$

## Different nonparametric estimation techniques

- Linear regression in certain cases (saturated regression)
- Kernel smoothing and nearest neighbors use "nearby" data
- Sieve approaches use increasingly flexible approximations
- "Machine learning" methods that automatically adjust to data (**adaptive**)

## Key concepts

- **Curse of dimensionality** — "hard" to estimate multivariate functions
- Most approaches require one or more **tuning parameters**

$\rightarrow$ These can sometimes be selected through **plug-in** or **cross-validation**

# Outline

1. Motivation and Overview
2. **Saturated Linear Regressions**
3. Kernel Smoothing, Local Polynomial and Nearest Neighbors
4. Sieves
5. Selecting Tuning Parameters
6. Machine Learning
7. Summary

## Binning estimator

- Suppose $X \in \{x_1, \ldots, x_K\}$ is discrete
- This covers cases where $X$ has multiple discrete components

  e.g. $X = \{(male, HS), (female, HS), (male, college), (female, college)\}$
- Then a nonparametric **binning estimator** is very natural: e.g.

$$\hat{\mu}(x) = \frac{1}{N_x} \sum_{i:X_i=x}^{N} Y_i \quad \text{where} \quad N_x \equiv \sum_{i=1}^{N} \mathbb{1}[X_i = x]$$

## Limitations and workarounds

- Only works if $X$ is completely discrete, otherwise $N_x = 0$ or 1!
- If $X$ is continuous (is anything continuous?), could discretely bin it

$\rightarrow$ This can be a bit arbitrary — views on this vary

- Poor finite sample performance if **small bins** ($K$ large relative to $N$)

## Regression implementation of a binning estimator

- Construct binary indicators: $W_k \equiv \mathbb{1}[X = x_k]$ for $k = 1, \ldots, K$
- Linear regression of $Y$ on $W_1, \ldots, W_K$ (with no constant)
- Coefficient on $W_k$ is *numerically* identical to $\hat{\mu}(x_k)$

## Alternative parameterization

- Can alternatively include a constant, but must drop a $W_k$ (why?)
- Suppose we drop $W_1$, then the coefficient on the constant is $\hat{\mu}(x_1)$
- Coefficients on $W_k$ is $\hat{\mu}(x_k) - \hat{\mu}(x_1)$ — sum with constant to get $\hat{\mu}(x_k)$

## Saturation

- Either of these specifications are called **saturated** (regressions)

$\rightarrow$ The specification has one parameter per unknown — can't take any more

- Useful as a conceptual baseline — parameterizations complicate things

# Outline

1. Motivation and Overview
2. Saturated Linear Regressions
3. **Kernel Smoothing, Local Polynomial and Nearest Neighbors**
4. Sieves
5. Selecting Tuning Parameters
6. Machine Learning
7. Summary

## Idea

- Suppose we want to estimate $\mu(x)$ at some fixed point $x \in \mathbb{R}$
- But there are few (or no) observations with $X = x$ in our data
- Idea is to instead use $X \approx x$ and **smooth** — makes sense if $\mu$ is smooth
- Many approaches do this with main difference being what $X \approx x$ means

## Uniform kernel smoothing

- Take a sample mean of $Y$ over observations with $x - h \leq X \leq x + h$

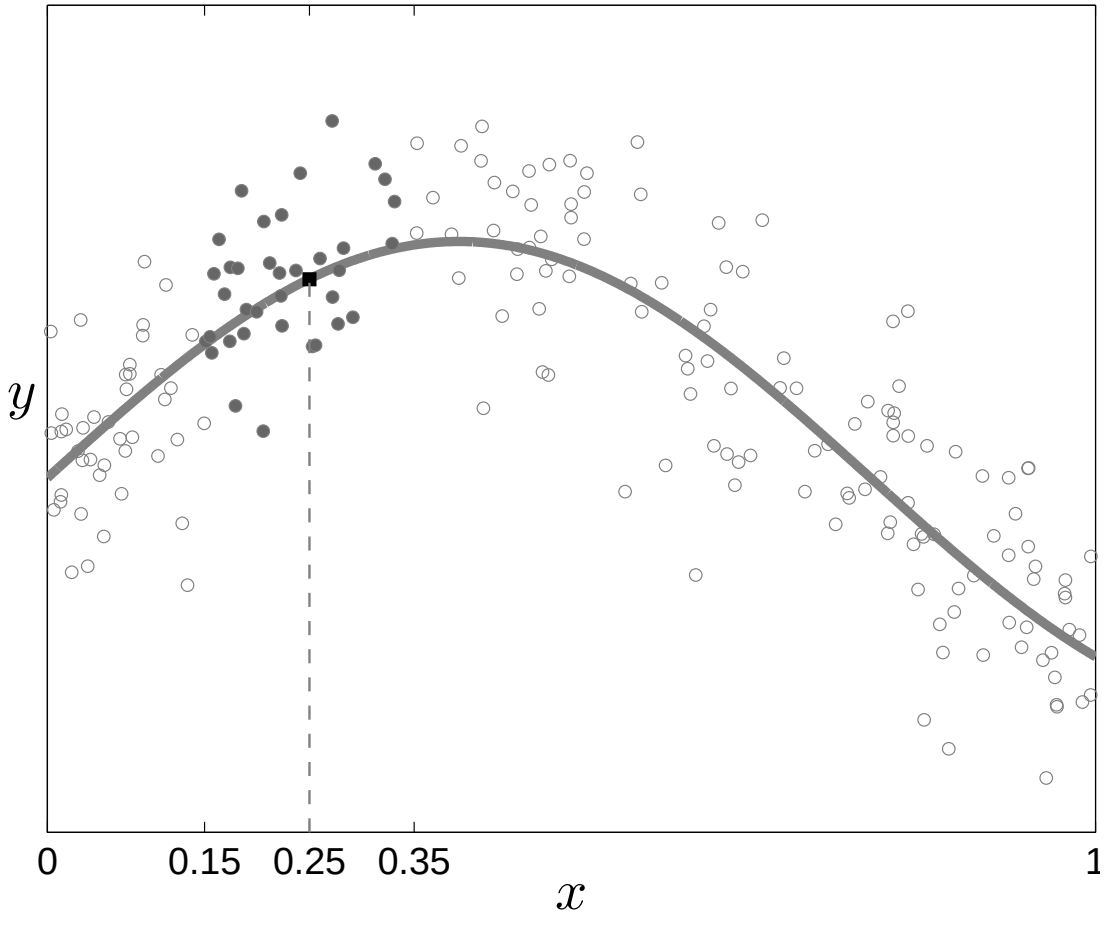
$$\hat{\mu}(x) \equiv \frac{\sum_{i=1}^{n} \mathbb{1}[x - h \leq X_i \leq x + h] Y_i}{\sum_{i=1}^{n} \mathbb{1}[x - h \leq X_i \leq x + h]} \approx \mathbb{E}[Y | x - h \leq X \leq x + h]$$

- $h > 0$ is a **bandwidth** parameter that needs to be chosen
- Smaller $h$ leads to a more local estimator with fewer observations

$\Rightarrow$ Smaller $h$ reduces bias, increases variance — the **bias-variance tradeoff**

- As $h \to \infty$, the estimator becomes the usual sample mean

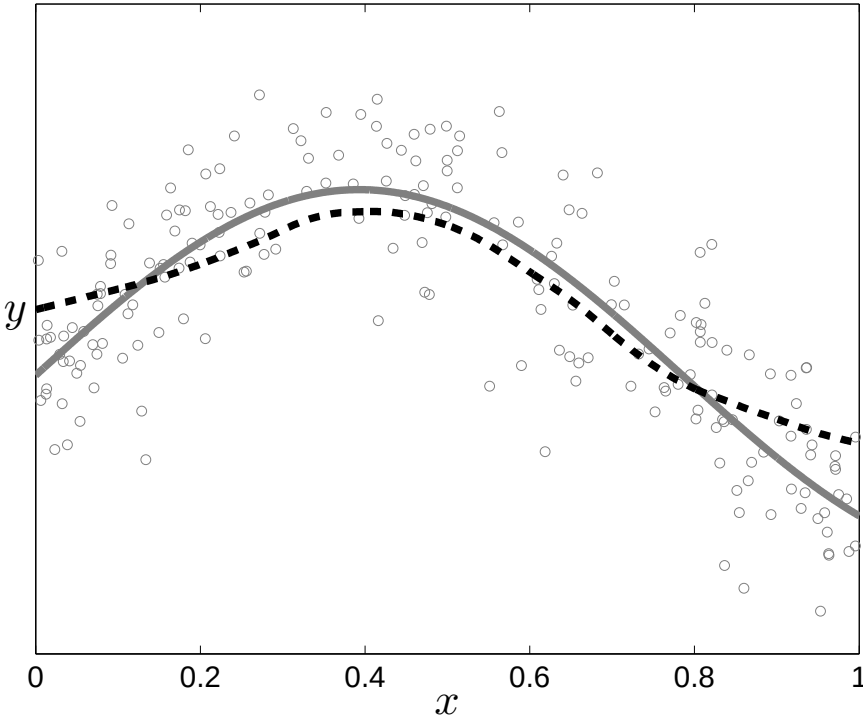
# Visualizing Uniform Kernel Smoothing



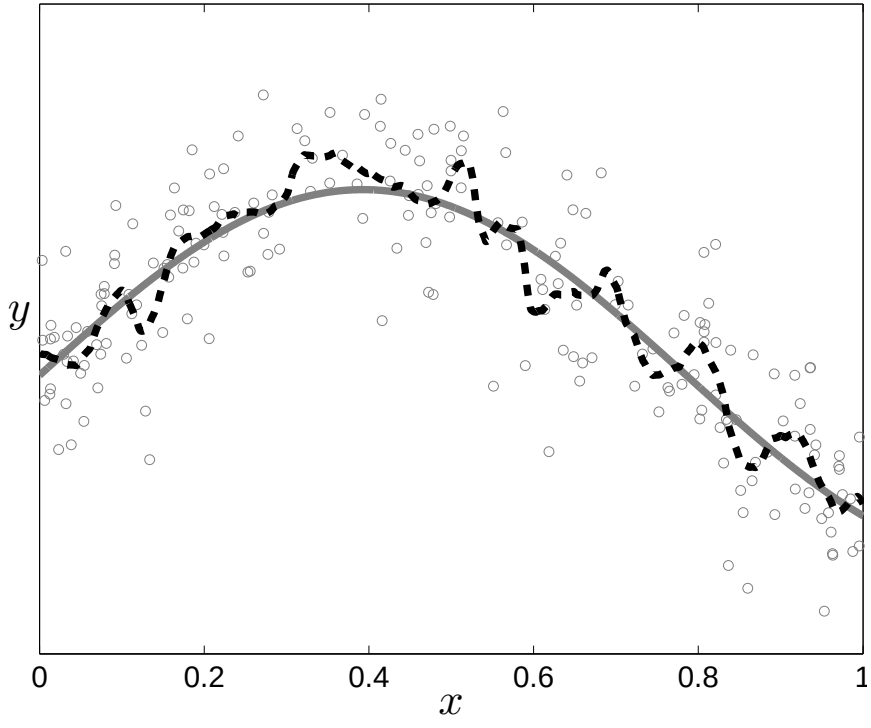

- The data was generated with $\mu(x)$ set to be the grey line
- Estimate $\mu(.25)$ with the simple mean of the shaded points $= \hat{\mu}(.25)$
- Bandwidth is $h = .1$

# Visualizing The Bias-Variance Trade-off

(a) $h = .3$




(b) $h = .03$




- The left is **oversmoothed** — high bias, low variance
- The right is **undersmoothed** — low bias, high variance
- (What does "too much/little" mean? Here we use "eyeball optimality")

## Definition

- Take weighted sample mean of $Y$ over *all* $X$

$$\hat{\mu}(x) \equiv \frac{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)}$$

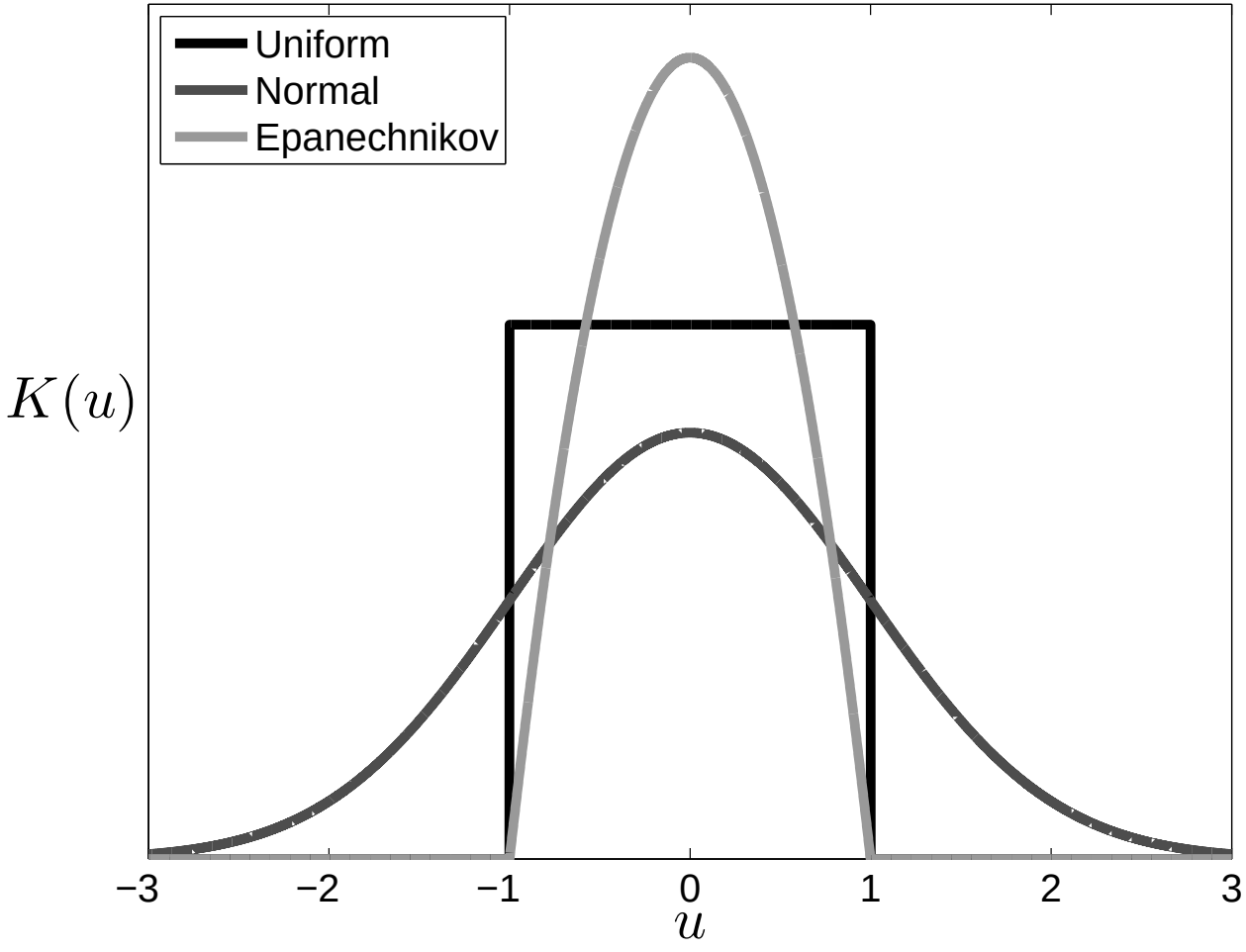
- Generalizes uniform case which had $K(u) \equiv \mathbb{1}[|u| \leq 1]$
- Common choice of $K$ is a standard normal pdf, but there are many others

$\rightarrow$ Any continuous density that is symmetric around 0 for example

## Choice of $K$

- Received wisdom is that $K$ doesn't matter much ($h$ is more important)
- So why generalize? Uniform kernel leads to $\hat{\mu}$ with kinks
- Smooth kernels produce more appealing (smooth) function estimators

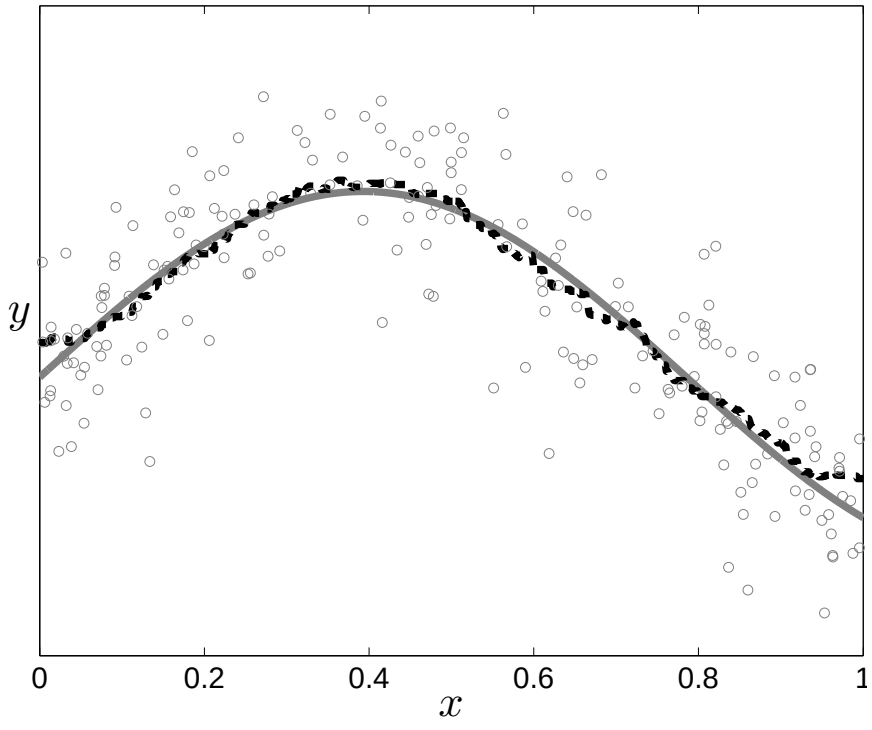
# Three Kernel Functions



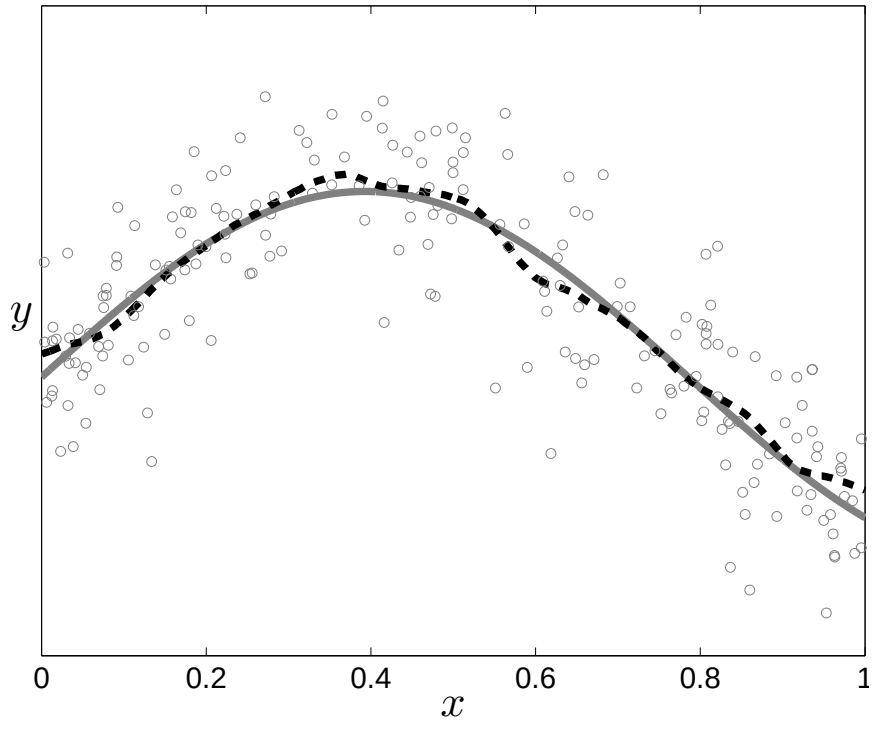

- The normal and Epanechnikov are smooth
- The uniform and Epanechnikov have bounded support on $[-1, 1]$

# Uniform vs. Smooth Kernels

(a) Uniform kernel




(b) Epanechnikov kernel




- If you look closely at the uniform will see abrupt jumps

$\rightarrow$ Happens for points $x$ at which $|X_i - x| = h$ for some $i$

- Usually not a big deal, but why have unnecessary visual distractions?

## Definition

- A uniform kernel is a very simple (local) regression: $Y$ on a constant
- A **local linear** estimator also adds $X$:

$$\underbrace{\min_{\mu_0} \sum_{i:X_i \approx x} (Y_i - \mu_0)^2}_{\text{uniform kernel (\textbf{local constant})}} \quad \text{vs.} \quad \underbrace{\min_{\mu_0,\mu_1} \sum_{i:X_i \approx x} (Y_i - \mu_0 - \mu_1 X_i)^2}_{\text{local linear}}$$

- If we add $X^2$ as well then have a **local quadratic** estimator

## Benefits of local linear estimators

- Easy to implement — regression with an "if" statement
- Can also change kernel — weighted regression with an "if" statement
- Same variance and "usually" lower bias than local constant estimators
- Especially at edges — local constant estimators have **boundary bias**

$\rightarrow$ We will return to this when discussing regression discontinuity designs

## Definition

- Simple idea: Average $Y$ for the $k$ observations with $X$ closest to $x$
- "Closest" here (as in kernel) is a tricky concept with multiple variables
- **Mahalonobis metric** — inverse-variance weighted Euclidean norm

$\rightarrow$ Doesn't help with discrete/categorical variables

## $k$–Nearest neighbors are adaptive kernels

- $k$–NN is a uniform local constant with $x$–varying bandwidth

$\rightarrow$ Same $k$, so $x$ in sparser areas use more distant observations

- This is an example of a locally **adaptive** estimator

$\rightarrow$ It adjusts the bandwidth to the data without our input

- Of course, we still need to choose $k$, so there is still a tuning parameter
- Unlike kernel, $k$–NN won't let you divide by zero

## A fundamental problem

- These approaches (and others) suffer from **the curse of dimensionality**

→ Estimator quality *rapidly* deteriorates with the dimension of $X$

- A statistical manifestation of a constant problem across science

## Statistical implications

- Formally, the **rate of convergence** of the estimators goes down

→ Adding a dimension requires much more data to maintain precision

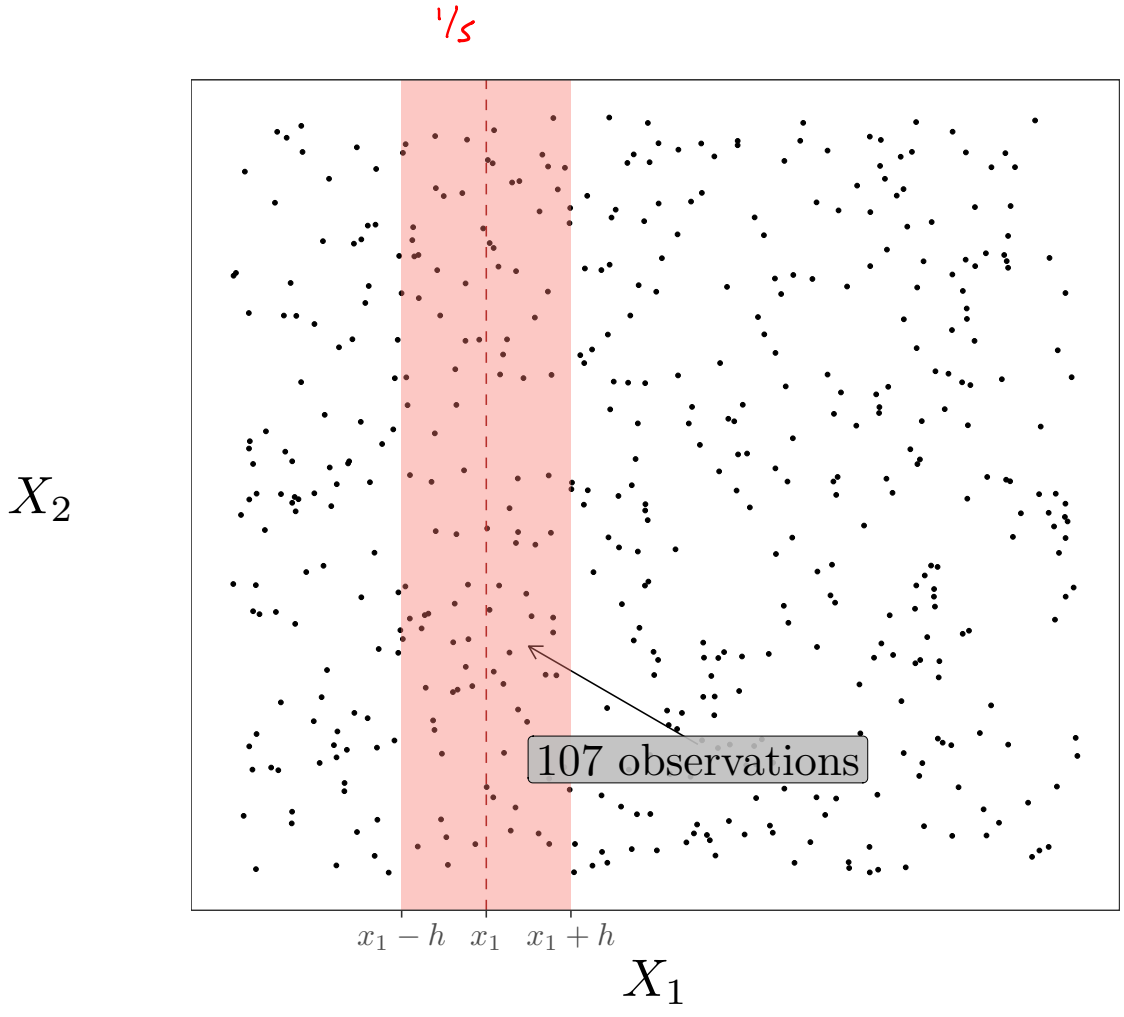
- If $X$ is scalar, number of observations around $\pm h$ of $x$ is roughly

$$N \times \mathbb{P}[x - h \leq X \leq x + h] = N \int_{x-h}^{x+h} f(u)\, du \approx N \times 2h \times f(x)$$

- If $X \in \mathbb{R}^2$, this drops to $4h^2 N \times f(x)$ — remember $h \to 0$ is "small"!

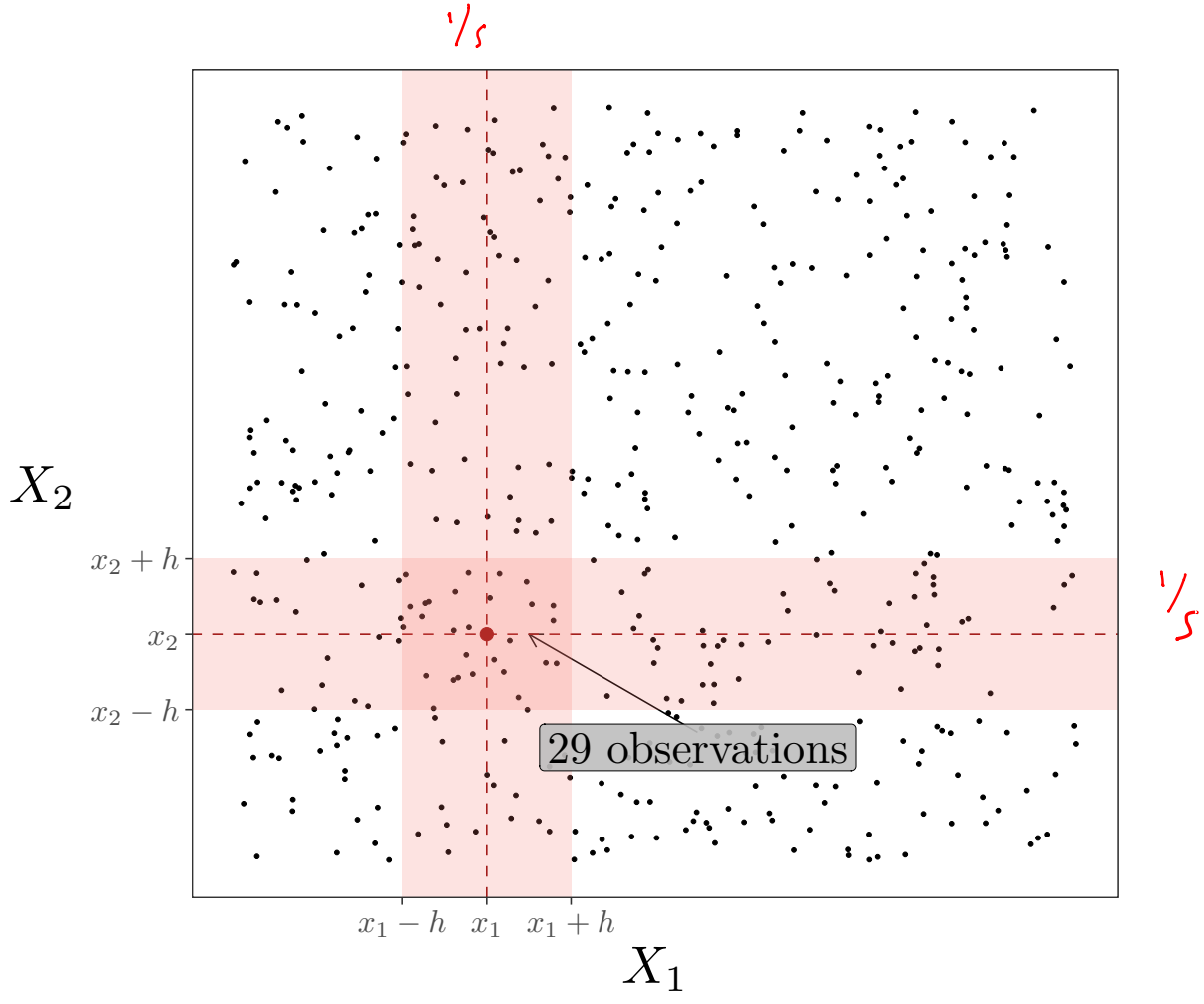
⇒ Need much larger $N$ (or $h$) to maintain the same "effective $N$"

# Visualizing The Curse of Dimensionality




- Kernel regression with 500 draws from a bivariate uniform $[0, 1]$
- Observations used to estimate $\mathbb{E}[Y|X = x_1]$ with bandwidth $h$

# Visualizing The Curve of Dimensionality




- Observations used to estimate $\mathbb{E}[Y|X = x_1, X = x_2]$ with bandwidth $h$
- Effective number of observations drops by an order of magnitude

# Outline

1. Motivation and Overview
2. Saturated Linear Regressions
3. Kernel Smoothing, Local Polynomial and Nearest Neighbors
4. **Sieves**
5. Selecting Tuning Parameters
6. Machine Learning
7. Summary

## Idea behind sieves/series (and basis expansions more generally)

- Suppose $\mu(x) \equiv \mathbb{E}[Y|X = x]$ is continuous with compact support
- Stone-Weierstrass Theorem: The **polynomial basis** can approximate $\mu$

$\rightarrow$ That is, arbitrarily well by increasing the order of the polynomial

- Many results like this in approximation theory and computer graphics

## Implementation

- Regress $Y$ on $1, X, X^2, \ldots, X^K$ for some "large" $K$
- Bias-variance trade-off: Larger $K$ is smaller bias, larger variance

$\rightarrow$ $K$ serves the same role as the bandwidth $h$ in kernel approaches

## Curse of dimensionality

- A $K$th order polynomial in one dimension has $K + 1$ terms
- A $K$th order polynomial in $J$ dimensions has $(K + 1)^J$ terms

$\Rightarrow$ Number of terms explodes with the dimension of $X$

## Flexibility

- Same idea can be applied to many estimators besides regression

$\rightarrow$ (Generalized) method of moments, maximum likelihood, etc.

- Extends easily to semiparametric models

$\rightarrow$ Replace unknown functions with a basis expansion

- Means sieves typically easier to work with for new/different models

## Shape constraints

- Suppose $\mu(x_1, x_2)$ is a third order polynomial (so 16 terms)
- Could assume no interaction effect: $\mu(x_1, x_2) = \mu(x_1) + \mu(x_2)$
- Breaks the curse of dimensionality with an interpretable assumption

$\rightarrow$ Can also be done with kernels, but much harder ("backfitting")

- More general shape constraints: monotonicity, concavity, etc.

$\rightarrow$ Ease of implementation for certain bases (e.g. B–splines, wavelets)

# Drawbacks of Sieve Approaches

## Basis choice

- With kernel estimation, the folklore is bandwidth matters, kernel doesn't
- Not true for sieves: Both the basis and the number of terms matter
- This gives the researcher more freedom, which might be a negative
- On the other hand, shape constraints often suggest certain bases
- Also much folklore — e.g. don't use standard polynomials!

## Local vs. global estimation

- Kernels are local because $\hat{\mu}(x)$ only uses data near $x$
- Some sieves are not, e.g. the standard polynomials

$\rightarrow$ Can lead to outliers having large influence and erratic estimates

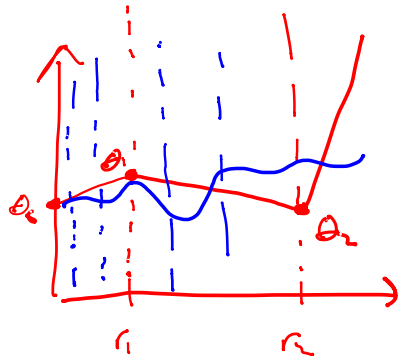
- Some merit to this view, as we will see in regression discontinuity design
- But really an issue of the basis, e.g. **polynomial splines** are local sieves

$\rightarrow$ Good in practice for a mix of local performance and ease of sieves

1. Motivation and Overview
2. Saturated Linear Regressions
3. Kernel Smoothing, Local Polynomial and Nearest Neighbors
4. Sieves
5. **Selecting Tuning Parameters**
6. Machine Learning
7. Summary


splines: $\mu(x) = \theta_0 + \theta_1 x + \sum_{k=2}^{K_n} \theta_k \mathbb{1}[x \geq r_{k-1}](x - r_{k-1})$

knot point




knots ↑ ⇒ more smooth

$\theta_0 + \theta_1 x + \theta_2 \mathbb{1}(x \geq r_1)(x - r_1) + \theta_2 \mathbb{1}[x \geq r_2)(x - r_2)$ } changes in slopes

## Idea behind bandwidth (or general tuning parameter) selection

- The bandwidth $h$ controls the bias-variance trade-off
- To determine the "right" $h$ we need to set up a criterion

$\rightarrow$ Usual choice is mean-squared error (MSE) or integrated (over $x$) MSE

- Want to choose $h$ to minimize $\mathrm{MSE}(h)$ — but we don't know $\mathrm{MSE}(h)$!

$\rightarrow$ It depends on the unknown mean function $\mu$, our object of estimation

## Plug-in methods

- Derive an approximation of $\mathrm{MSE}(h)$ (via Taylor expansion)
- Minimize the approximation yields a closed-form expression for $h^\star$

$\rightarrow$ This expression is still going to depend on $\mu$, $\mu'$, $\mu''$ and density of $X$ at $x$

- Estimate these objects with a **pilot bandwidth**, then plug-in to get $h^\star$
- Use the estimate of $h^\star$ in actual estimation of $\mu$
- Circular? Hope is that estimation is "less sensitive" to pilot bandwidth

## A more natural alternative

- Partition data into $K$ **folds**, and pick a tuning parameter
- Estimate using $K - 1$ folds, evaluate criterion on the $K$th fold
- Repeat for all $K$ folds, average the resulting $K$ criterion values
- Repeat and minimize average criterion with respect to tuning parameter

## Common implementations

- $K = N$ (sample size) is **leave-one-out** cross-validation
- $K = 10$ is common for more computationally intensive estimators

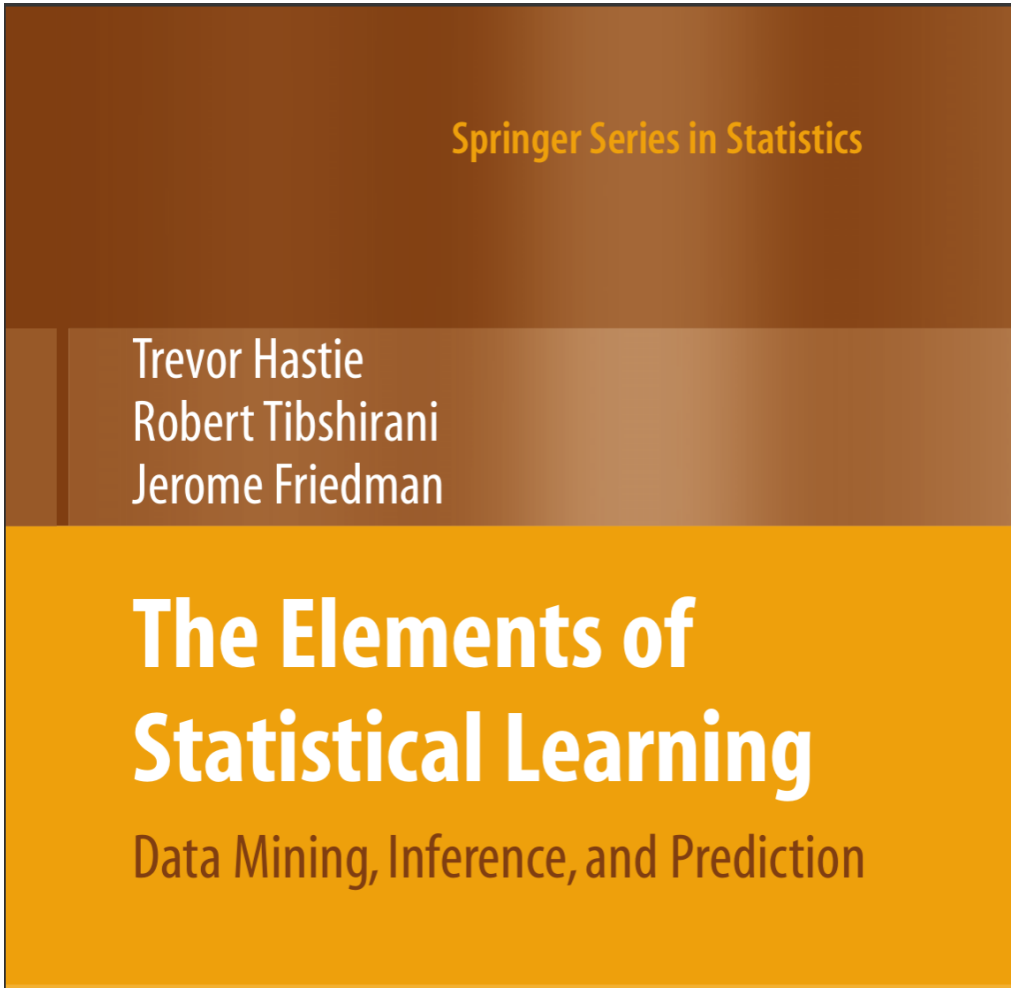
## Pros and cons of cross-validation

- Intuitive, easy to explain and motivate, but computationally intensive
- No pilot bandwidth, although there is the choice of folds
- Need to be able to construct a proper criterion — not always possible

# Outline


1. Motivation and Overview
2. Saturated Linear Regressions
3. Kernel Smoothing, Local Polynomial and Nearest Neighbors
4. Sieves
5. Selecting Tuning Parameters
6. **Machine Learning**
7. Summary

# "Supervised Learning" = Nonparametric Regression




- Hastie, Tibshirani and Friedman ("the Bible") covers all previous topics
- It also covers fancier topics (neural networks, random forests)
- Focus on richer models and model selection for **prediction**

**3 Linear Methods for Regression** **43**
3.1 Introduction . . . . . 43
3.2 Linear Regression Models and Least Squares . . . . . 44
3.2.1 Example: Prostate Cancer . . . . . 49
3.2.2 The Gauss–Markov Theorem . . . . . 51
3.2.3 Multiple Regression from Simple Univariate Regression . . . . . 52
3.2.4 Multiple Outputs . . . . . 56
3.3 Subset Selection . . . . . 57
3.3.1 Best-Subset Selection . . . . . 57
3.3.2 Forward- and Backward-Stepwise Selection . . . 58
3.3.3 Forward-Stagewise Regression . . . . . 60
3.3.4 Prostate Cancer Data Example (Continued) . . 61
3.4 Shrinkage Methods . . . . . 61
3.4.1 Ridge Regression . . . . . 61
3.4.2 The Lasso . . . . . 68
3.4.3 Discussion: Subset Selection, Ridge Regression and the Lasso . . . . . 69
3.4.4 Least Angle Regression . . . . . 73

- Hastie, Tibshirani and Friedman (“the Bible”) covers all previous topics
- It also covers fancier topics (neural networks, random forests)
- Focus on richer models and model selection for **prediction**

# "Supervised Learning" = Nonparametric Regression 


**5 Basis Expansions and Regularization 139**

5.1 Introduction . . . . . 139
5.2 Piecewise Polynomials and Splines . . . . . 141
5.2.1 Natural Cubic Splines . . . . . 144
5.2.2 Example: South African Heart Disease (Continued) 146
5.2.3 Example: Phoneme Recognition . . . . . 148
5.3 Filtering and Feature Extraction . . . . . 150
5.4 Smoothing Splines . . . . . 151
5.4.1 Degrees of Freedom and Smoother Matrices . . . 153
5.5 Automatic Selection of the Smoothing Parameters . . . . 156
5.5.1 Fixing the Degrees of Freedom . . . . . 158
5.5.2 The Bias–Variance Tradeoff . . . . . 158
5.6 Nonparametric Logistic Regression . . . . . 161
5.7 Multidimensional Splines . . . . . 162
5.8 Regularization and Reproducing Kernel Hilbert Spaces . 167
5.8.1 Spaces of Functions Generated by Kernels . . . 168
5.8.2 Examples of RKHS . . . . . 170
5.9 Wavelet Smoothing . . . . . 174
5.9.1 Wavelet Bases and the Wavelet Transform . . . 176
5.9.2 Adaptive Wavelet Filtering . . . . . 179


- Hastie, Tibshirani and Friedman ("the Bible") covers all previous topics
- It also covers fancier topics (neural networks, random forests)
- Focus on richer models and model selection for **prediction**

# "Supervised Learning" = Nonparametric Regression 


**6 Kernel Smoothing Methods** **191**

6.1 One-Dimensional Kernel Smoothers . . . . . 192
6.1.1 Local Linear Regression . . . . . 194
6.1.2 Local Polynomial Regression . . . . . 197
6.2 Selecting the Width of the Kernel . . . . . 198
6.3 Local Regression in $\mathbb{R}^p$ . . . . . 200
6.4 Structured Local Regression Models in $\mathbb{R}^p$ . . . . . 201
6.4.1 Structured Kernels . . . . . 203
6.4.2 Structured Regression Functions . . . . . 203
6.5 Local Likelihood and Other Models . . . . . 205
6.6 Kernel Density Estimation and Classification . . . . . 208
6.6.1 Kernel Density Estimation . . . . . 208
6.6.2 Kernel Density Classification . . . . . 210
6.6.3 The Naive Bayes Classifier . . . . . 210
6.7 Radial Basis Functions and Kernels . . . . . 212
6.8 Mixture Models for Density Estimation and Classification 214
6.9 Computational Considerations . . . . . 216
Bibliographic Notes . . . . . 216
Exercises . . . . . 216


- Hastie, Tibshirani and Friedman ("the Bible") covers all previous topics
- It also covers fancier topics (neural networks, random forests)
- Focus on richer models and model selection for **prediction**

**15 Random Forests** **587**
15.1 Introduction . . . . . 587
15.2 Definition of Random Forests . . . . . 587
15.3 Details of Random Forests . . . . . 592
15.3.1 Out of Bag Samples . . . . . 592
15.3.2 Variable Importance . . . . . 593
15.3.3 Proximity Plots . . . . . 595
15.3.4 Random Forests and Overfitting . . . . . 596
15.4 Analysis of Random Forests . . . . . 597
15.4.1 Variance and the De-Correlation Effect . . . . . 597
15.4.2 Bias . . . . . 600
15.4.3 Adaptive Nearest Neighbors . . . . . 601
Bibliographic Notes . . . . . 602
Exercises . . . . . 603

- Hastie, Tibshirani and Friedman (“the Bible”) covers all previous topics
- It also covers fancier topics (neural networks, random forests)
- Focus on richer models and model selection for **prediction**

## Machine learning methods are designed for prediction

- "Is this photo of a cat or a dog?"

$\rightarrow$ I know, you know it, but can we use a computer to automate it?

## Causal inference is about . . . inference

- "What is the average treatment effect of $D$ on $Y$?"

## Good prediction methods can be bad inference methods

$$Y = \beta_1 X_1 + \cdots + \beta_K X_K + U \quad \text{with} \quad \mathbb{E}[U|X] = 0$$

- Prediction: Given $x_1, \ldots, x_K$, what is $Y$?
- Inference: What is $\beta_1$?
- Suppose $X_2$ and $X_1$ are highly correlated
- Good prediction would omit $X_2$ — it contains similar information

$\Rightarrow$ Omitted variables bias for estimate of $\beta_1 \Rightarrow$ bad inference

**The quality of predictions can be more easily measured**

- With prediction problems you know **the ground truth**

→ We know if the photo is actually a cat or dog

- Common practice of using hold-out samples to evaluate performance
- Predicting the future is different of course, but we can still use the past

**The quality of inference depends on your assumptions**

- The data never "speaks for itself"

→ Causal inference always requires assumptions

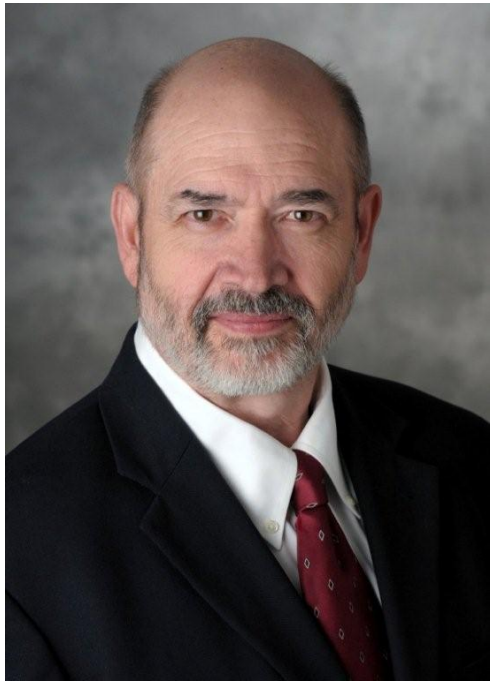
- No algorithm can tell you which assumptions are "credible"

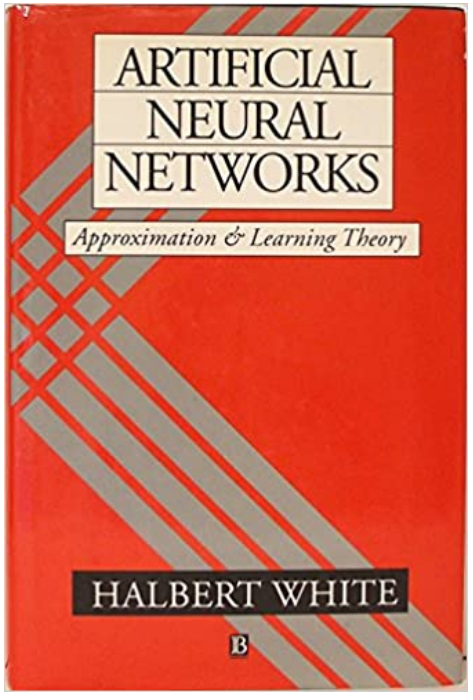
**Careful model selection**

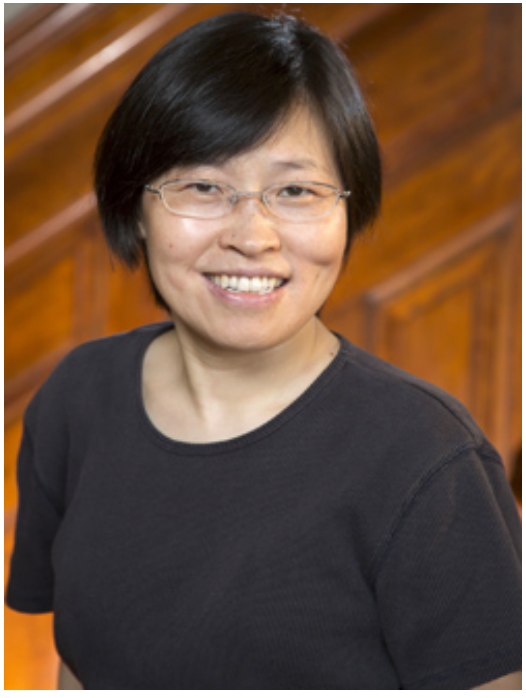
- However, *given* some assumptions, model selection could be useful
- We will see a couple of examples of this later in the course

# There is Nothing New Under the Sun







674 IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 12, NO. 4, JULY 2001

Semiparametric ARX Neural-Network Models with
an Application to Forecasting Inflation

Xiaohong Chen, Jeffrey Racine, and Norman R. Swanson

**What has been will be again; what has been done will be done again**

- Some econometricians are skeptical of machine learning trend

$\rightarrow$ Recent usage by empirical economists often smells like a fad

- Hostility to new ideas? Or "been there, done that"?

# Outline

1. Motivation and Overview
2. Saturated Linear Regressions
3. Kernel Smoothing, Local Polynomial and Nearest Neighbors
4. Sieves
5. Selecting Tuning Parameters
6. Machine Learning
7. **Summary**

## The nonparametric ideal

- Nonparametric identification and then estimation
- Useful paradigm for creating *clear* empirical arguments

→ Empirical work is an argument, so unclear ⇒ unconvincing

- Parameterizations often needed in practice for dimension reduction

→ But it's useful to separate the conceptual from the practical

## Nonparametric regression methods

- Local regression methods and $k$–nearest neighbors

→ Local is transparent, intuitive, not sensitive to outliers (by definition)

- Function approximation methods broadly called sieve methods

→ Flexible but more opaque, shape constraints, can be global or local

- Machine learning focused on prediction, and prediction $\neq$ inference