Machine Learning and AI in Economics

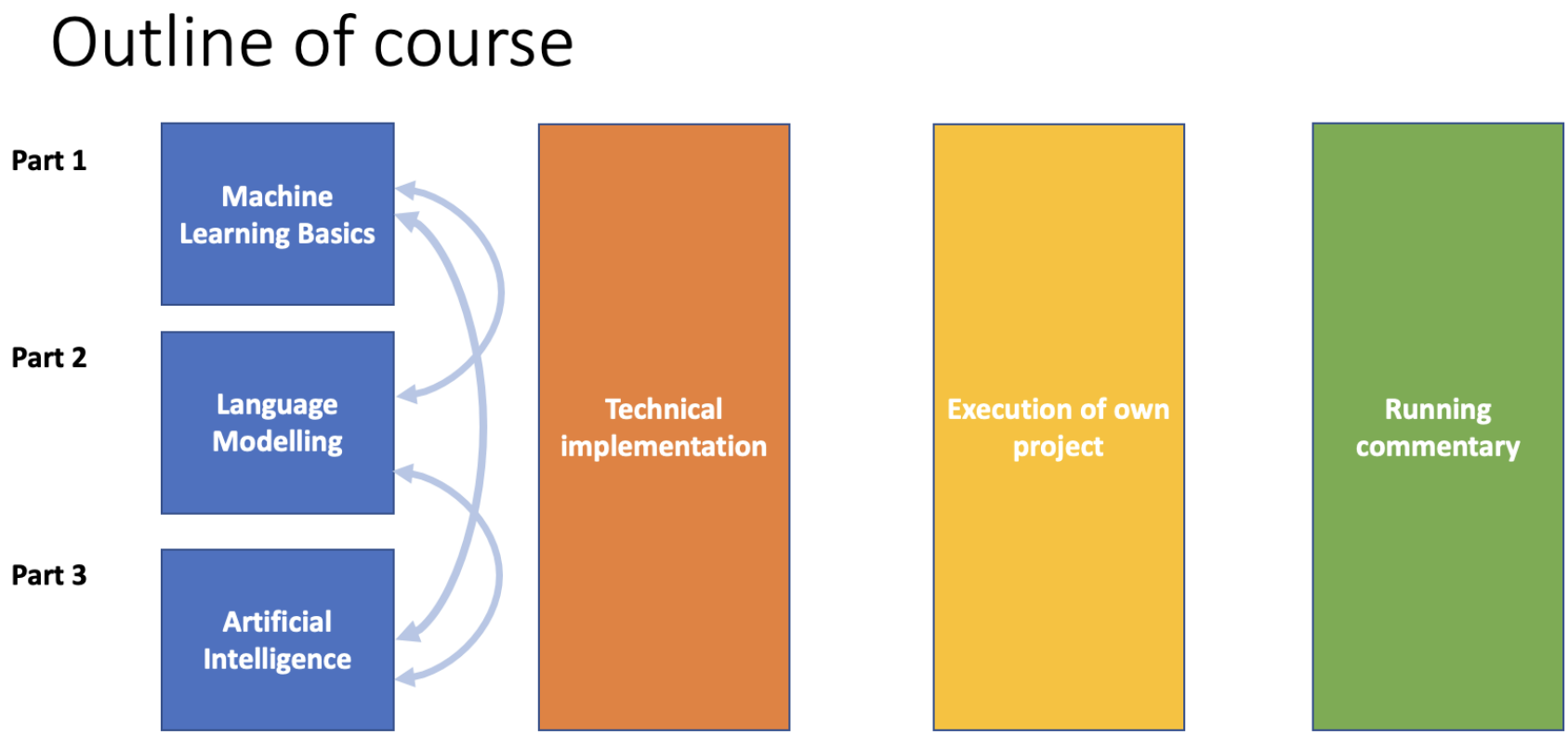
Thiemo Fetzer

University of Warwick & Bonn & CEPR & LSE & NIESR & AEAI

March 12, 2026

This course

Aims to fuse three main areas of work



Technical implementation

We will be experimenting

- Develop our own classifiers or solutions to common ML problems and think about their applications in economics
- We will deploy large language models locally to work "off the grid" and try to embed them into workflows
- We will aim to understand the interlinkages with concrete examples

This course

Aims to fuse three main areas of work

Items covered

Machine Learning Basics

Model fitting
BSS, Lasso, GLM, GAM
Model evaluation
Dimensionality Reduction
Classification
Clustering
Neural Networks

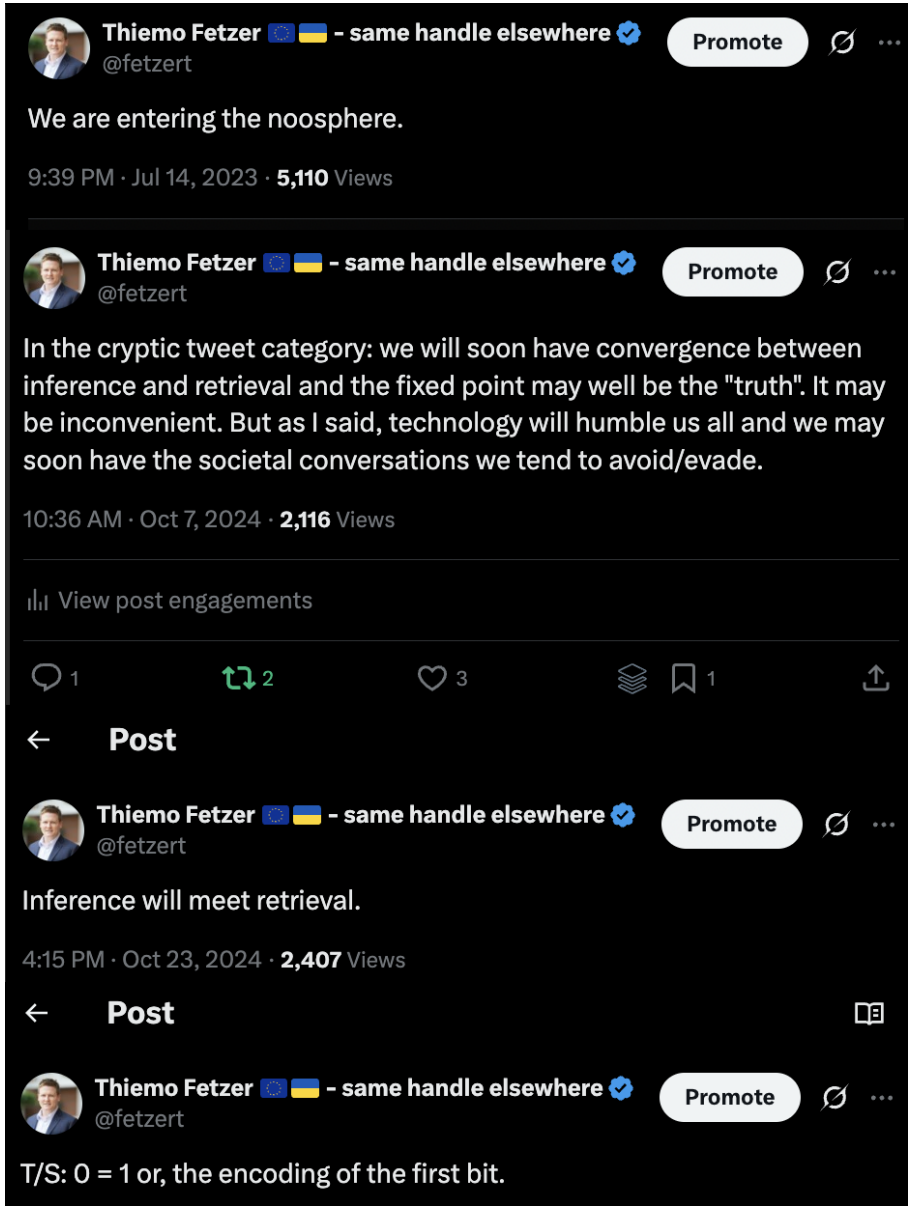
Language Modelling

Text as sequences
POS, NER, ...
Conventional processing
Tagging & classification
Ngram language model
Vector representation
Embeddings
Topic modelling

Artificial Intelligence

Neural networks
Transformers
Large language models
Large language model architecture

Running commentary



Thiemo Fetzer – same handle elsewhere
@fetzert
Promote

We are entering the noosphere.

9:39 PM · Jul 14, 2023 · **5,110** Views

Thiemo Fetzer – same handle elsewhere
@fetzert
Promote

In the cryptic tweet category: we will soon have convergence between inference and retrieval and the fixed point may well be the "truth". It may be inconvenient. But as I said, technology will humble us all and we may soon have the societal conversations we tend to avoid/evade.

10:36 AM · Oct 7, 2024 · **2,116** Views

View post engagements

1 2 3 1

← Post

Thiemo Fetzer – same handle elsewhere
@fetzert
Promote

Inference will meet retrieval.

4:15 PM · Oct 23, 2024 · **2,407** Views

← Post

Thiemo Fetzer – same handle elsewhere
@fetzert
Promote

T/S: 0 = 1 or, the encoding of the first bit.

Plan

Illustrating some concepts

1. **Listen to some natural text**

   Who can still remember the first words I used at the start of this lecture?
2. **What did you have for breakfast**

   Distinguishing between what is relevant and not.
3. **What is learning?**

   Connecting dots.
4. **What do humans need look like in the post materialistic society?**

   Attention is all you need or alternatives to human flourishing.
5. **What are we actually doing?**

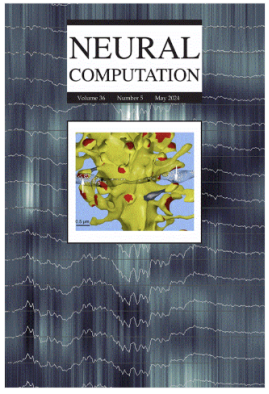
   Connecting dots.

What a byte?

object size	legacy
1	1 bytes
1024	1 Kb
1024^2	1 Mb
1024^3	1 Gb
1024^4	1 Tb
1024^5	1 Pb

How about the human brain?

Volume 36, Issue 5
May 2024



‹ Previous Article Next Article ›

Article Contents

April 23 2024

Synaptic Information Storage Capacity Measured With Information Theory

In Special Collection: CogNet

Mohammad Samavat, Thomas M. Bartol, Kristen M. Harris, Terrence J. Sejnowski

Neural Computation (2024) 36 (5): 781–802.
https://doi.org/10.1162/neco_a_01659 **Article history**

Abstract

Variation in the strength of synapses can be quantified by measuring the anatomical properties of synapses. Quantifying precision of synaptic plasticity is fundamental to understanding information storage and retrieval in neural circuits. Synapses from the same axon onto the same dendrite have a common history of coactivation, making them ideal candidates for determining the precision of synaptic plasticity based on the similarity of their physical dimensions. Here, the precision and amount of information stored in synapse dimensions were quantified with Shannon information theory, expanding prior analysis that used signal detection theory (Bartol et al., 2015). The two methods were compared using dendritic spine head volumes in the middle of the stratum radiatum of hippocampal area CA1 as well-defined measures of synaptic strength. Information theory delineated the number of distinguishable synaptic strengths based on nonoverlapping bins of dendritic spine head volumes. Shannon entropy was applied to measure synaptic information storage capacity (SISC) and resulted in a lower bound of 4.1 bits and upper bound of 4.59 bits of information based on 24 distinguishable sizes. We further compared the distribution of distinguishable sizes and a uniform distribution using Kullback-Leibler divergence and discovered that there was a nearly uniform distribution of spine head volumes across the sizes, suggesting optimal use of the distinguishable values. Thus, SISC provides a new analytical measure that can be generalized to probe synaptic strengths and capacity for plasticity in different brain regions of different species and among animals raised in different conditions or during learning. How brain diseases and disorders affect the precision of synaptic plasticity can also be probed.

Estimated Synaptic Storage Capacity

- Number of synapses in the human brain:
  $\sim 10^{14}$ (100 trillion)
- Synaptic information storage capacity (SISC) per synapse:
  4.1 to 4.6 bits (Samavat et al., 2024)
- **Total brain capacity:**

$$(4.1 \text{ to } 4.6) \times 10^{14} \text{ bits} \approx 51 - 57 \text{ TB}$$

- *Assuming 8 bits = 1 byte and converting to terabytes*

Comparison With Other Data Sources

Data Source	**Size Estimate**
Human brain (estimated)	51–57 TB
All of Wikipedia (text only)	$\sim$ 20 GB
Entire English Wikipedia (with media)	$\sim$ 200 GB
Netflix's daily data usage	$\sim$ 1000 TB (1 PB)
YouTube (new uploads per day)	$\sim$ 720 TB
All digitized books (Google estimate)	$\sim$ 15 TB
Entire web (text content)	$\sim$ 50–100 TB
Internet Archive (Wayback Machine)	$>$ 70 PB

Table: Human brain vs. digital data scales (approximate).

Key Takeaways

- The human brain likely stores ∼50–60 TB of data using synaptic strength encoding.
- That's **more than all of Wikipedia and most of the web's text content**.
- But it's still **tiny compared to the size of the entire Internet**.
- Biological information storage is efficient, flexible, and dynamic—not just about raw capacity.

How Much Information Does the Brain Process Daily?

- **Visual input alone:** $\sim$74 GB/day (Koch et al., 2006)
- **Full sensory input:** Estimated at $>$**100 GB/day**
- **Total (incl. unconscious processing):** Up to **10+ TB/day**
- Most of this information is *not* retained in long-term memory

Takeaway

The brain filters massive streams of data, encoding only what's salient.

What Does This Imply About Forgetting?

- If the brain stored all incoming data, it would run out of space in:
  - $\sim$1.5 years (100 GB/day)
  - $\sim$2 months (1 TB/day)
  - $<$1 week (10 TB/day)
- But that doesn't happen.

Key Insight

Forgetfulness isn't a flaw — it's a feature. The brain filters, compresses, and abstracts, retaining only what serves future decisions.

Processing vs. Storage: The Brain's Efficiency

Information Processed Daily:

- ∼70–100 GB/day (conscious perception)
- Up to **10+ TB/day** (all brain activity)

Long-Term Storage Capacity:

- ∼51–57 TB total (Samavat et al., 2024)
- Based on $\sim 10^{14}$ synapses $\times$ 4.1–4.6 bits/synapse

Key Insight

The brain compresses, abstracts, and forgets — storing only what matters. It's not a camera, it's a meaning-maker.

How People Process Information

- Information is not passively received — it is actively **filtered and interpreted**.
- Two key dimensions of interpretation:
  1. **Personal filters**
     Beliefs, values, lived experience, cognitive biases
  2. **Institutional or mechanical filters**
     Media framing, educational systems, algorithmic curation

Interpretive Divergence

Even when exposed to the same facts or evidence, individuals and groups may arrive at **radically different conclusions**.

How People Process Information

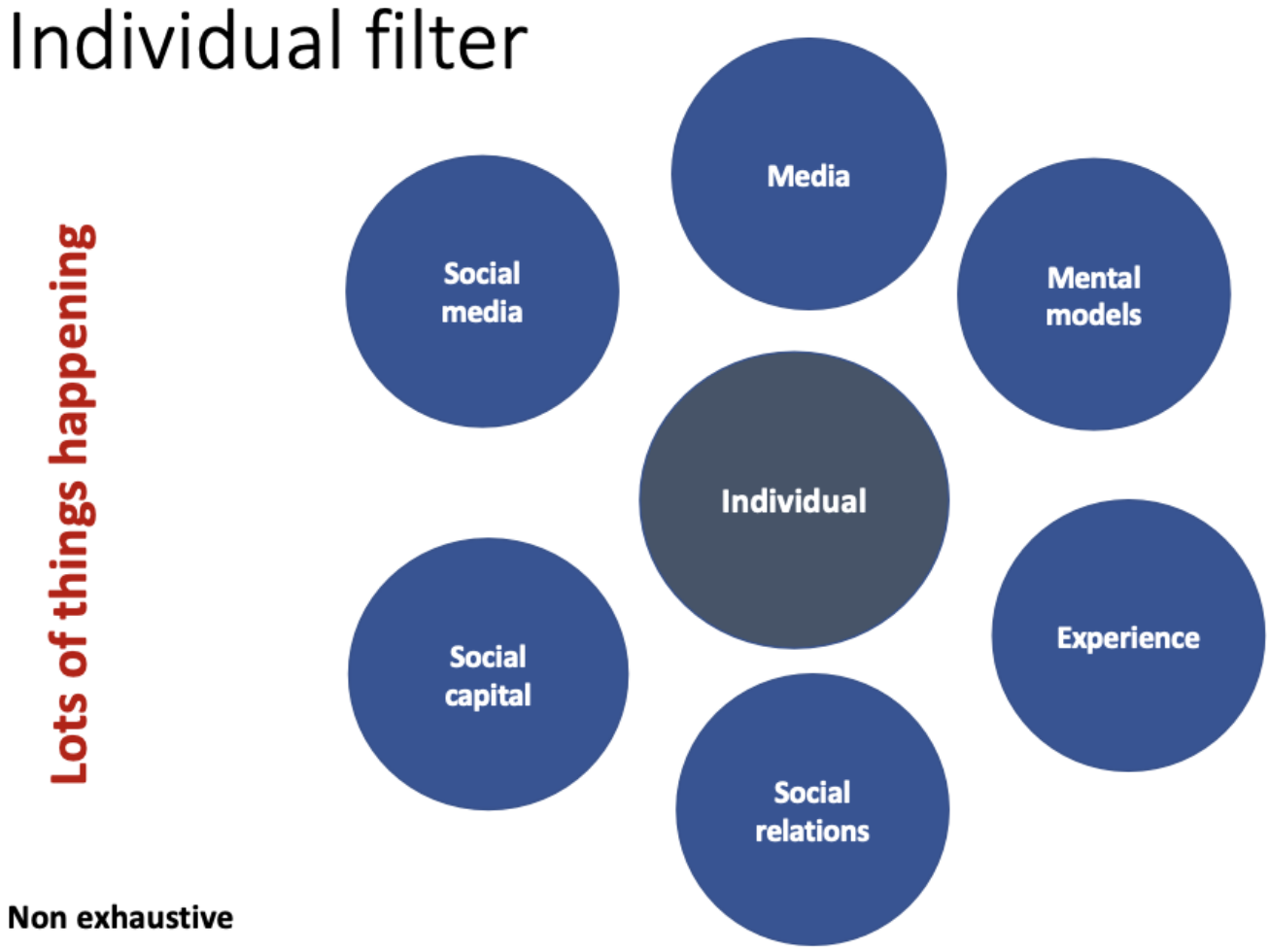
- Information is not passively received — it is actively **filtered and interpreted**.
- Two key dimensions of interpretation:
  1. **Personal filters**
     Beliefs, values, lived experience, cognitive biases
  2. **Institutional or mechanical filters**
     Media framing, educational systems, algorithmic curation

Interpretive Divergence

Even when exposed to the same facts or evidence, individuals and groups may arrive at **radically different conclusions**.

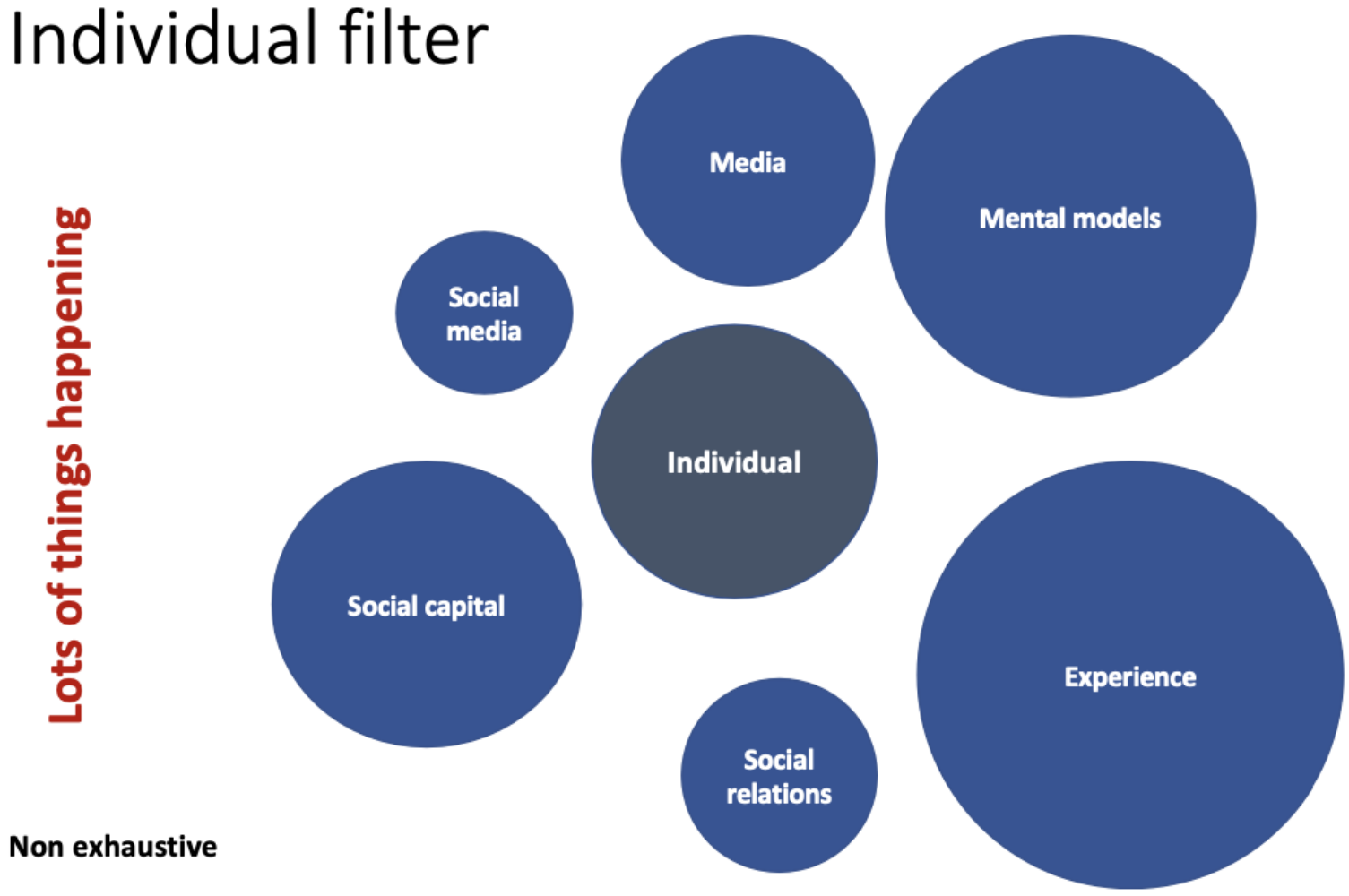
These filters play a crucial role in shaping narratives — especially during transitions, crises, or social change.

Processing information



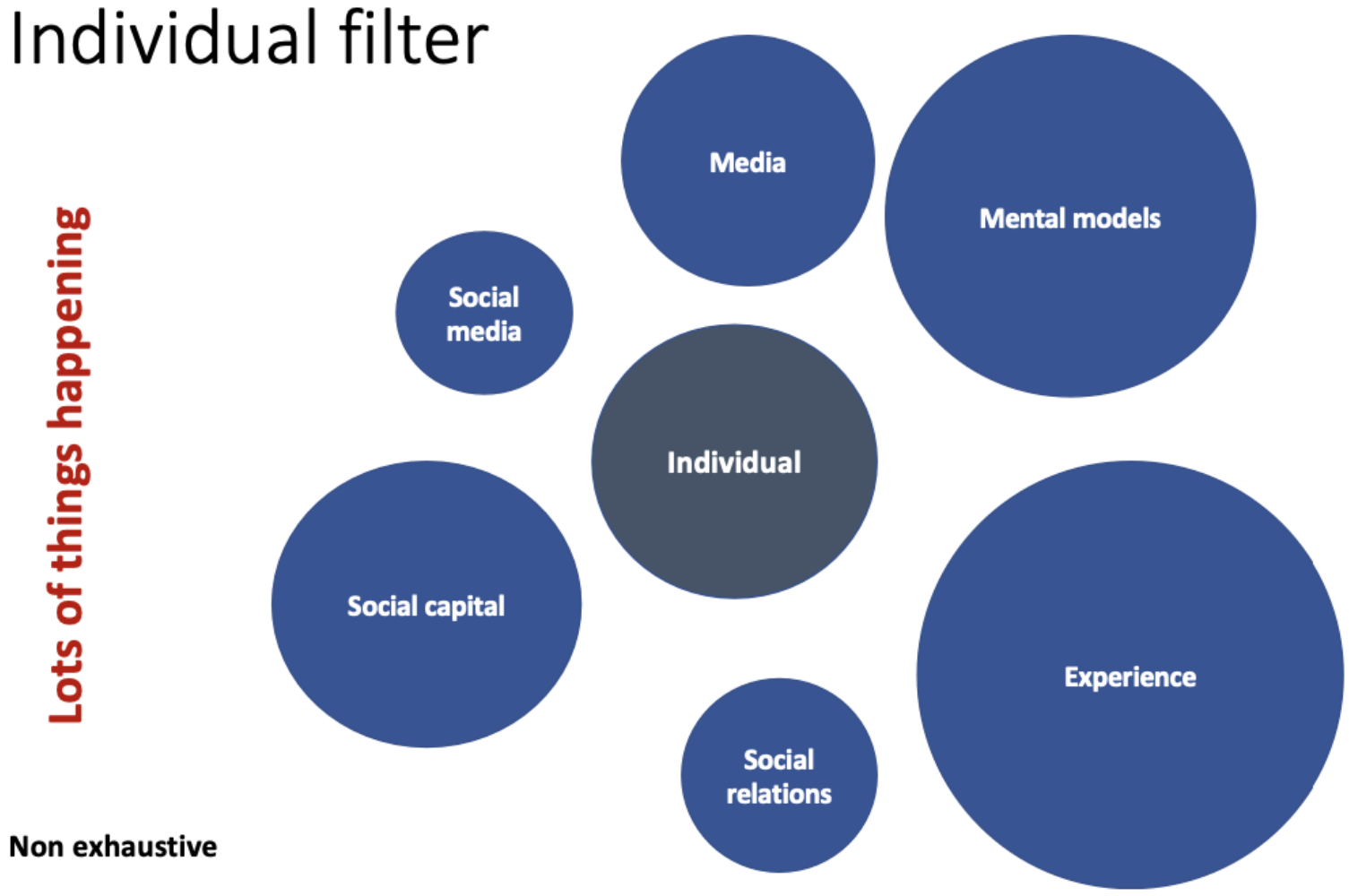
from Keynote at SHAPER Workshop 2023 on the *Political Economy of Climate (In)Action*.

Processing information



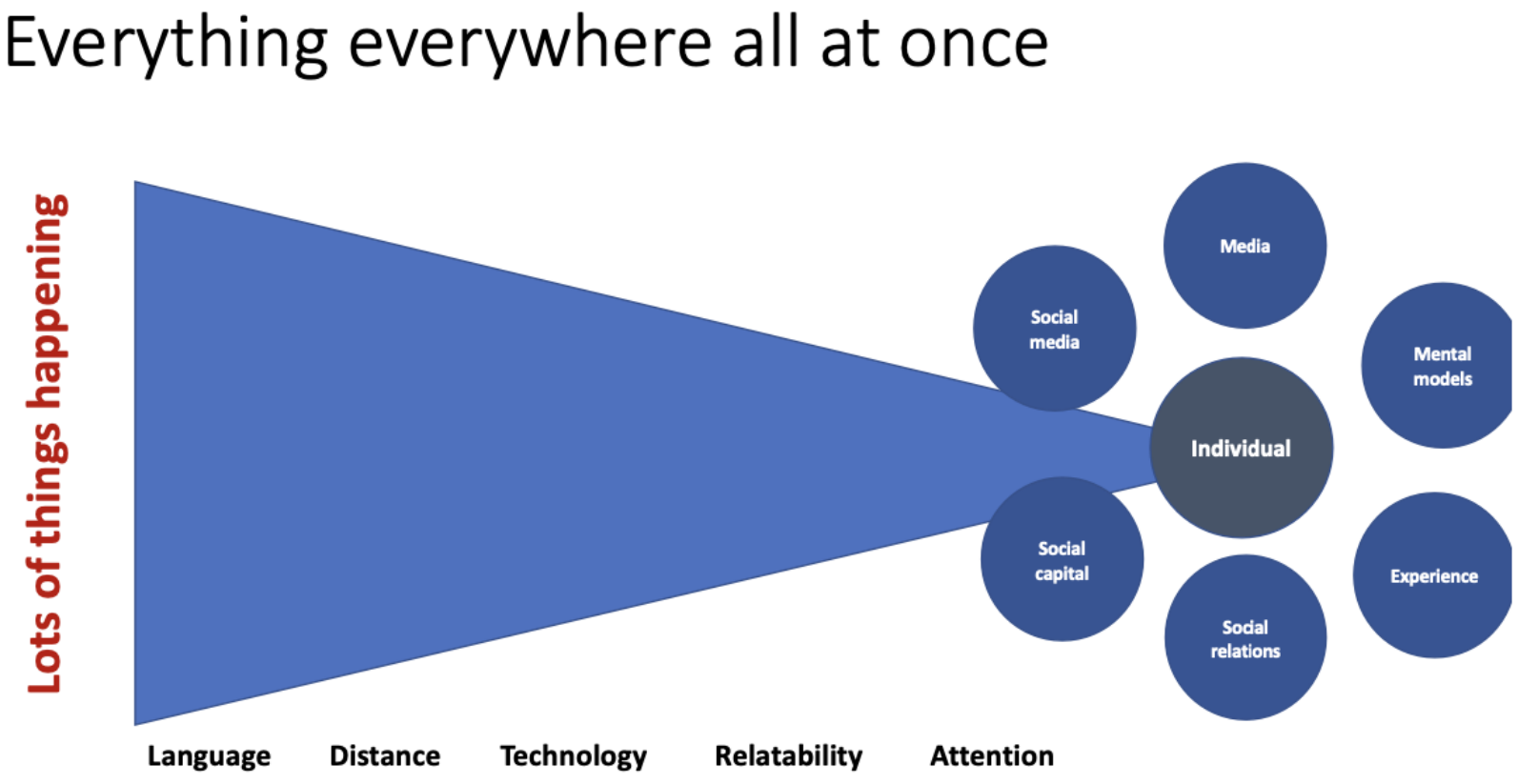
from Keynote at SHAPER Workshop 2023 on the *Political Economy of Climate (In)Action*.

Processing information



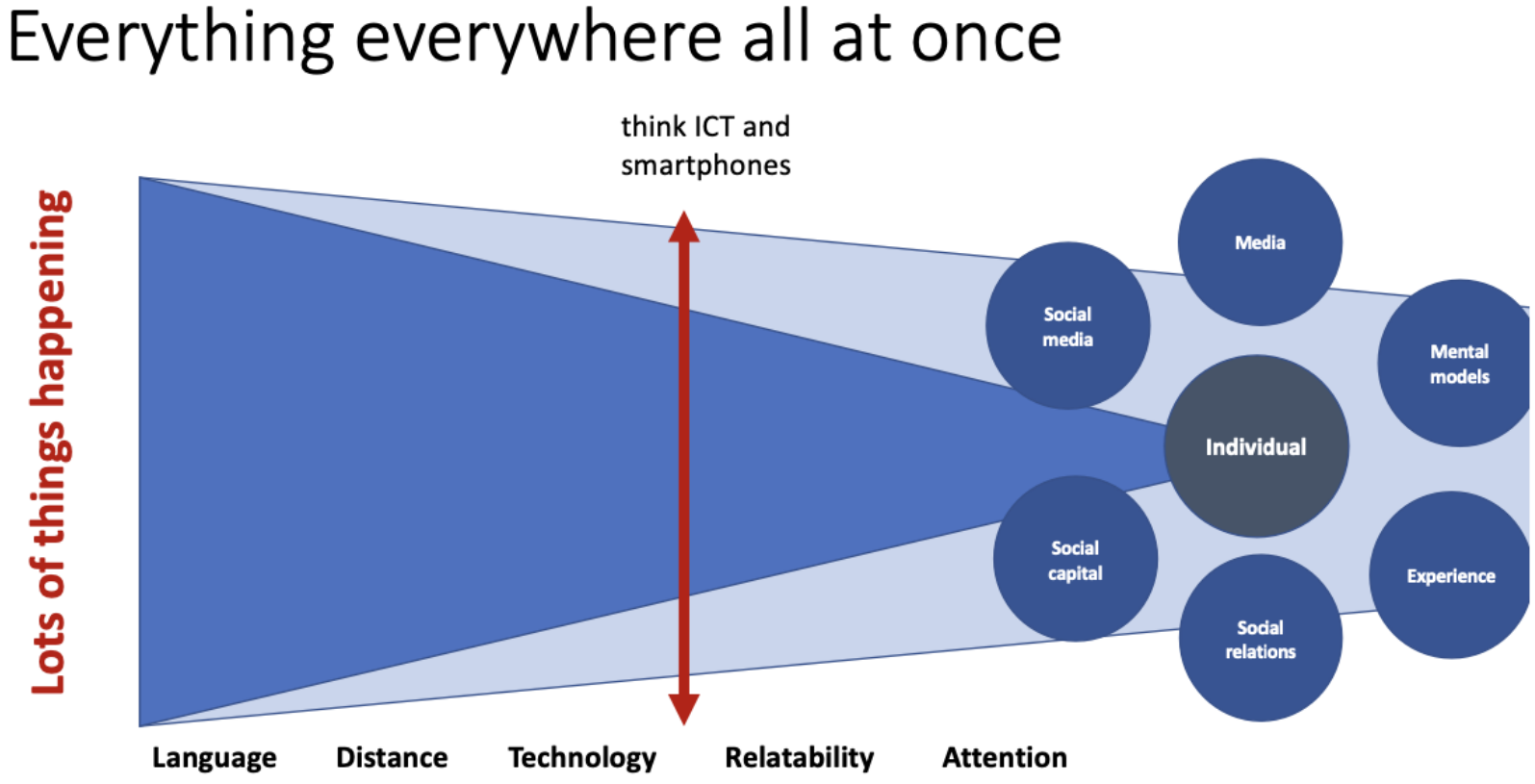
from Keynote at SHAPER Workshop 2023 on the *Political Economy of Climate (In)Action*.

Information flow



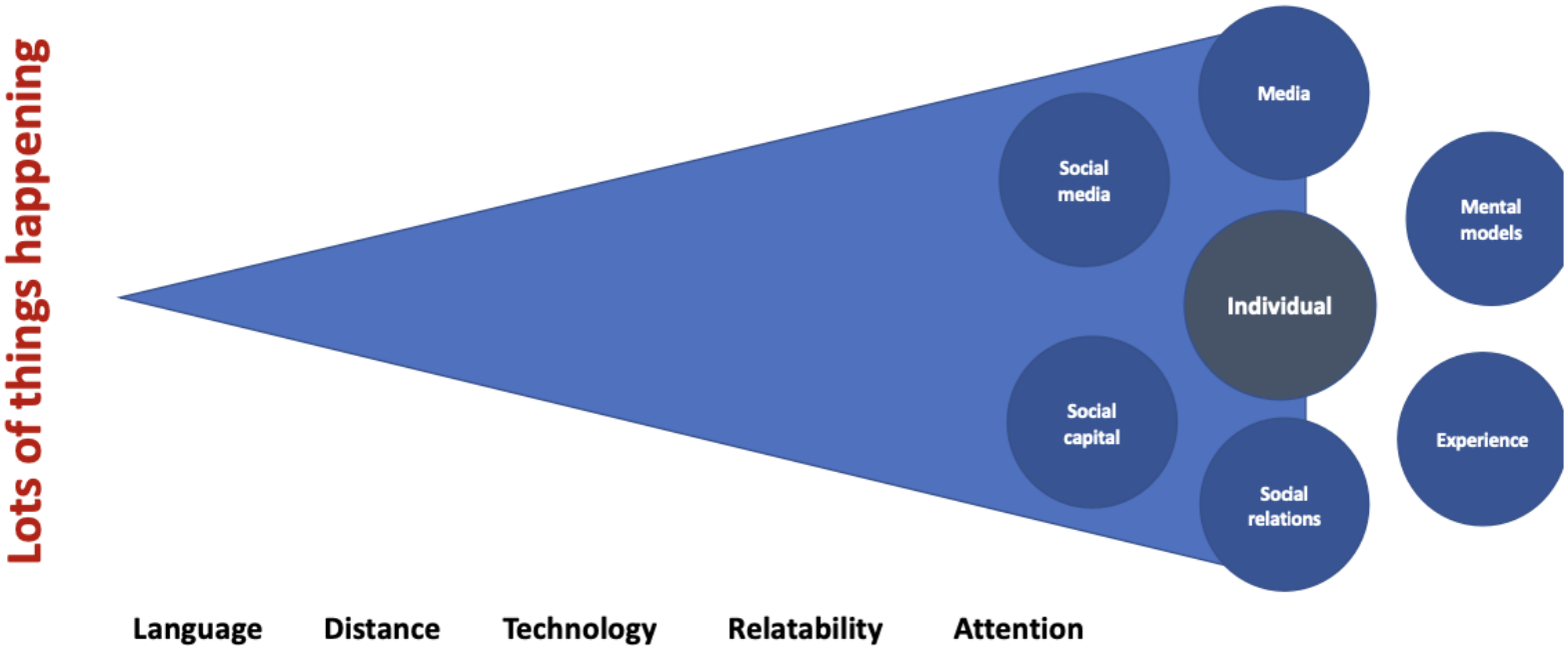
from Keynote at SHAPER Workshop 2023 on the *Political Economy of Climate (In)Action*.

Technology shocks



Fetzer & Garg (2025) Network Determinants of Cross-Border Media Coverage of Natural Disasters.

Incredibly close and extremely loud



from Keynote at SHAPER Workshop 2023 on the *Political Economy of Climate (In)Action*.

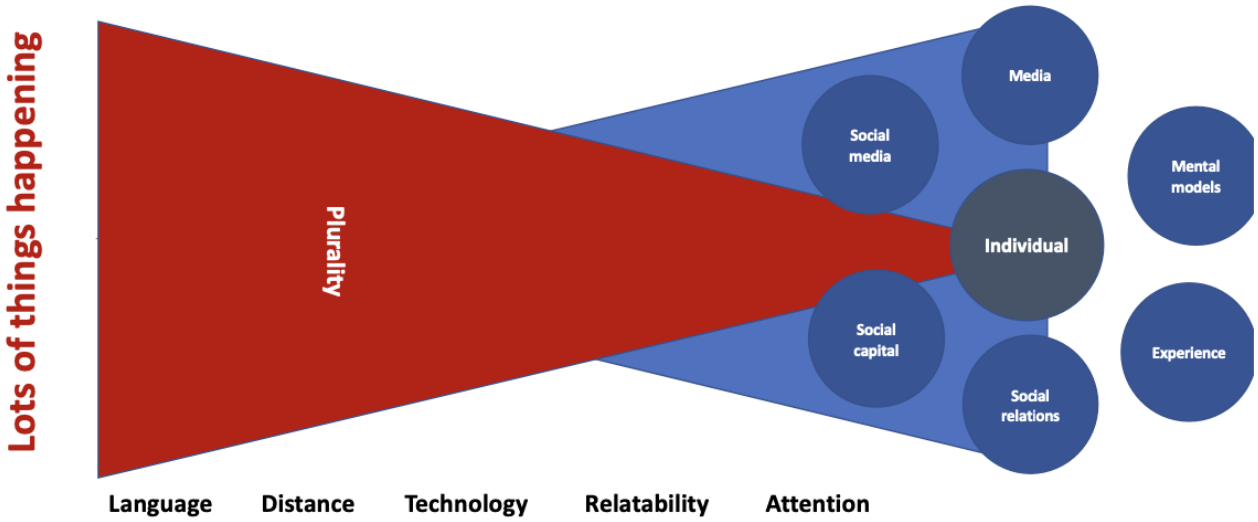
from Keynote at SHAPER Workshop 2023 on the *Political Economy of Climate (In)Action*.

Information flow

from Keynote at SHAPER Workshop 2023 on the *Political Economy of Climate (In)Action*.

Influencers (?)

Potential social optimum is a mixture



Key Insight

He who controls the information sphere can effectively exercise "control". This has geopolitical implications.

How Do We Perceive Time?

- Time perception is **psychological**, not physical
- Influenced by:
  - **Attention and arousal**
  - **Novelty and emotional intensity**
  - **Memory encoding**
- More change/events = longer perceived duration

How Do We Perceive Time?

- Time perception is **psychological**, not physical
- Influenced by:
  - **Attention and arousal**
  - **Novelty and emotional intensity**
  - **Memory encoding**
- More change/events = longer perceived duration

Example

"Time flies when you're having fun" — but we *remember* those moments more vividly.

Why Does the Brain Crave Experience?

- Brain = **predictive engine**
- Constantly seeks to minimize surprise
- Novel experiences provide:
  - Model updates (learning)
  - Emotional salience (dopamine, memory consolidation)
  - Subjective expansion of time

Why Does the Brain Crave Experience?

- Brain = **predictive engine**
- Constantly seeks to minimize surprise
- Novel experiences provide:
  - Model updates (learning)
  - Emotional salience (dopamine, memory consolidation)
  - Subjective expansion of time

Implication

We pursue experience not just to "feel alive" — but to update internal models of the world.

The Experience Economy and Cognitive Design

- Routine compresses time perception
- Novelty stretches it and anchors memory
- This has real design implications:
  - Travel, art, learning, risk-taking
  - Avoid passive consumption and algorithmic dullness

The Experience Economy and Cognitive Design

- Routine compresses time perception
- Novelty stretches it and anchors memory
- This has real design implications:
  - Travel, art, learning, risk-taking
  - Avoid passive consumption and algorithmic dullness

Designing for Memory

Shape experiences that maximize meaningful change, not just stimulation.

Demographics and the Experience Economy

- As people age, life often contains fewer genuine “firsts” and less novelty
- Fewer new memory anchors can compress subjective time and make years feel as if they pass faster
- When this is paired with sharper awareness of mortality, the value of **salient experience** can rise
- This creates demand for excitement, travel, learning, culture, wellness, community, and other experience-intensive services
- Older cohorts also often control more wealth and disposable resources, so this can become a powerful **demand pillar** in aging economies

Demographics and the Experience Economy

- As people age, life often contains fewer genuine "firsts" and less novelty
- Fewer new memory anchors can compress subjective time and make years feel as if they pass faster
- When this is paired with sharper awareness of mortality, the value of **salient experience** can rise
- This creates demand for excitement, travel, learning, culture, wellness, community, and other experience-intensive services
- Older cohorts also often control more wealth and disposable resources, so this can become a powerful **demand pillar** in aging economies

Economic pivot

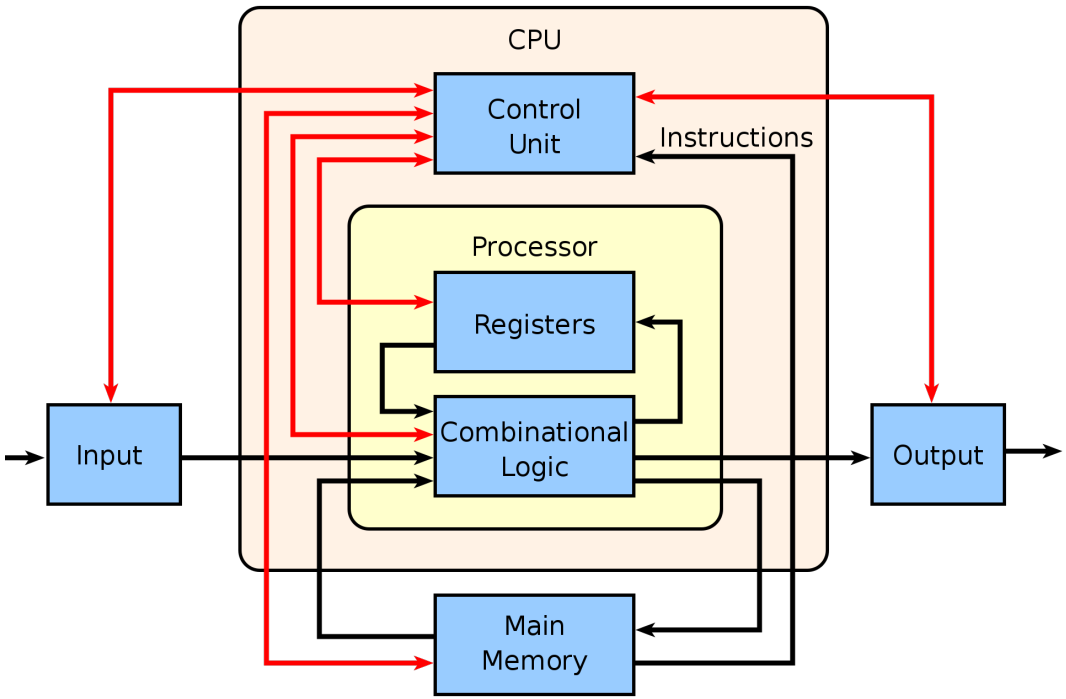
Demographic change may shift consumption away from mere accumulation toward experiences, memory creation, and time-saving services.

Summary

- The brain filters and encodes vast amounts of information daily
- Our perception of time is shaped by memory, attention, and novelty
- The craving for experience is biological, not just cultural
- Designing for experience = designing for salience and memory

"We are not made to remember days, we are made to remember moments."

Processing information in a Von Neuman Computer architecture



A CPU interacts with memory, input/output, and internal components like registers and control logic.

Why do you need RAM at all?

- RAM = Rapid Access Memory
- Data in R is stored in RAM
- Speed of data manipulations depends on memory speed
- Large datasets may exceed available RAM $\rightarrow$ data is swapped to HDD
- Swap reduces performance

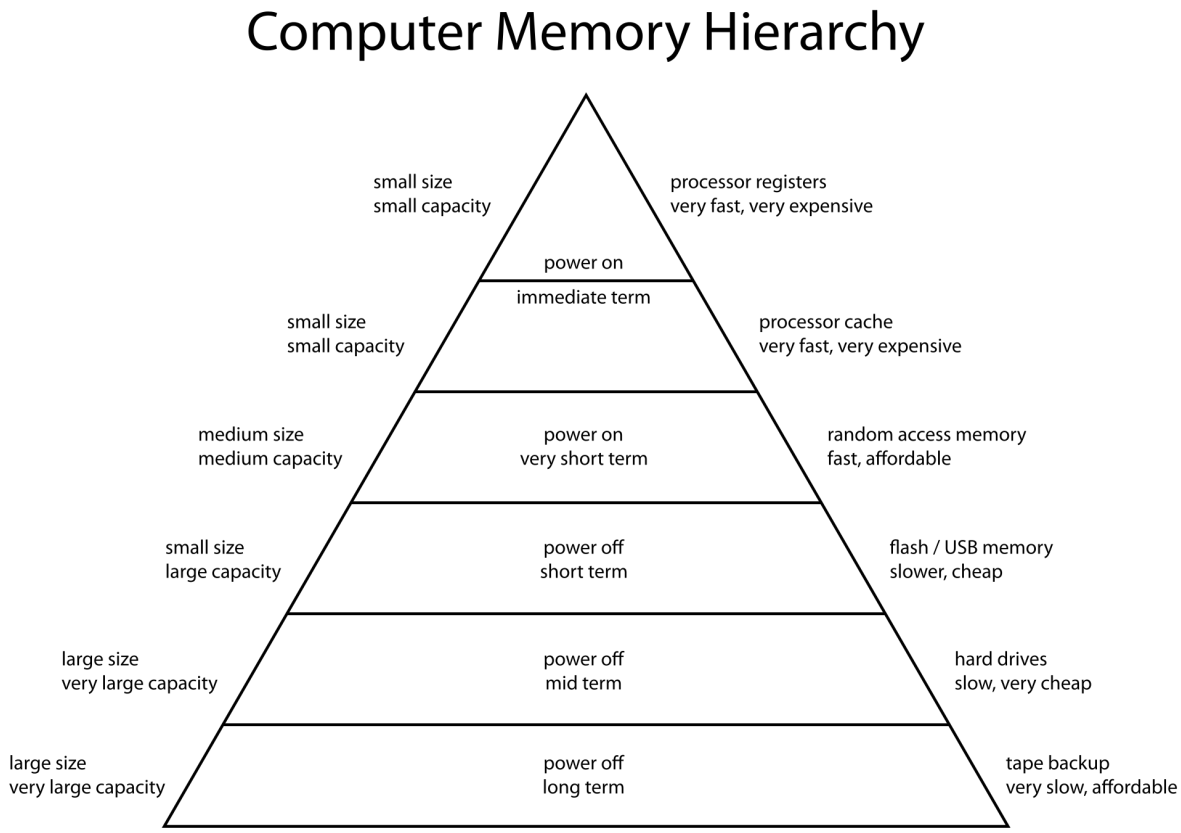
Typical read times:

- RAM: $\sim$100 nanoseconds
- SSD: $\sim$16,000 nanoseconds

Typical throughput:

- HDD: 80–160 MB/s
- SSD: 200–550 MB/s

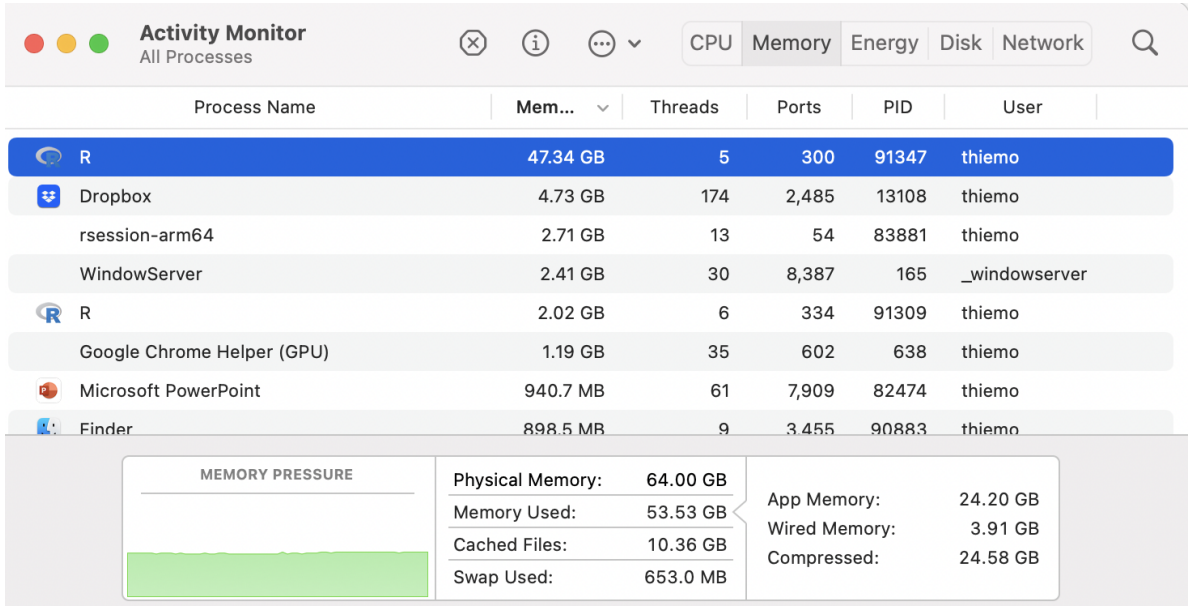
Hierarchy of memory in terms of speed



What are the implications?

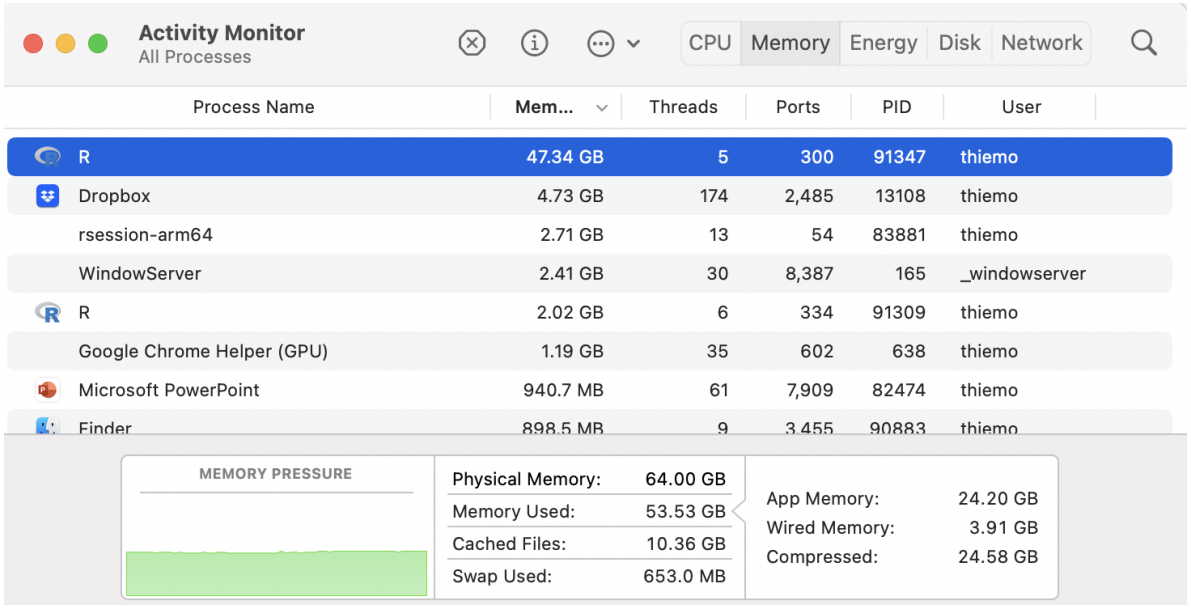
- Know how to **slice and dice** data efficiently
- Use **clean data objects** and good coding practice
- Process data in well-sized **chunks**
- Chunks should **fit into memory**
- Many researchers **lack awareness** of basic architecture

Keeping track of resource usage



R handling of memory

- Loading an object in R allocates memory
- Offloading from lists helps manage memory better



Plan

How does machine learning differ from what economists typically do?

Classical econometrics is different...

1. **No black box models** "Econometrics" is the branch of economics that aims to give empirical content to theory based economic relations providing a candidate f.
2. Focus on **Causality**: rich set of methods that help researchers establish *causality*, while data sciencetist are happy about associations.
3. **Internal validty** Applied Econometrics tends to exclusively focus on internal validity, i.e. getting an unbiased estimate of a coefficient, but does not make (out of sample) predictions.
4. **Linear parametric** Applied econometrics is strongly focused on methods to estimate linear parametric relationships, i.e. $f(X) = \beta_0 + X_1\beta_1 + ... + X_p\beta_p + \epsilon$

Plan

Statistical Learning

- Statistical learning refers to the set of methods to obtain robust predictive models.
- Suppose you have data on a set of variables $X_1, X_2, ..., X_p$, and we *assume* there exists a relationship between Y and $\mathbf{X} = (X_1, X_2, ..., X_p)$, that can be written as:

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

- f represents the systematic way that $\mathbf{X}$ carries information about Y.
- Statistical learning refers to the different ways of estimating f.

Why would we want to Estimate f?

There are two main motivations to estimate f...

1. **Prediction** Suppose you have data on $\mathbf{X}$, but lack information on Y; a "good" estimate $\hat{f}$ of the true f allows you to predict.

   If you assume that $\hat{f}$ and X are fixed, then:

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

2. **Inference**

   Which variables $X_1, X_2, ..., X_p$ are associated with Y?

   Are the effects meaningful in terms of size? Do effects work through the interaction?

   Is linearity an adequate assumption?

A good predictive model can...

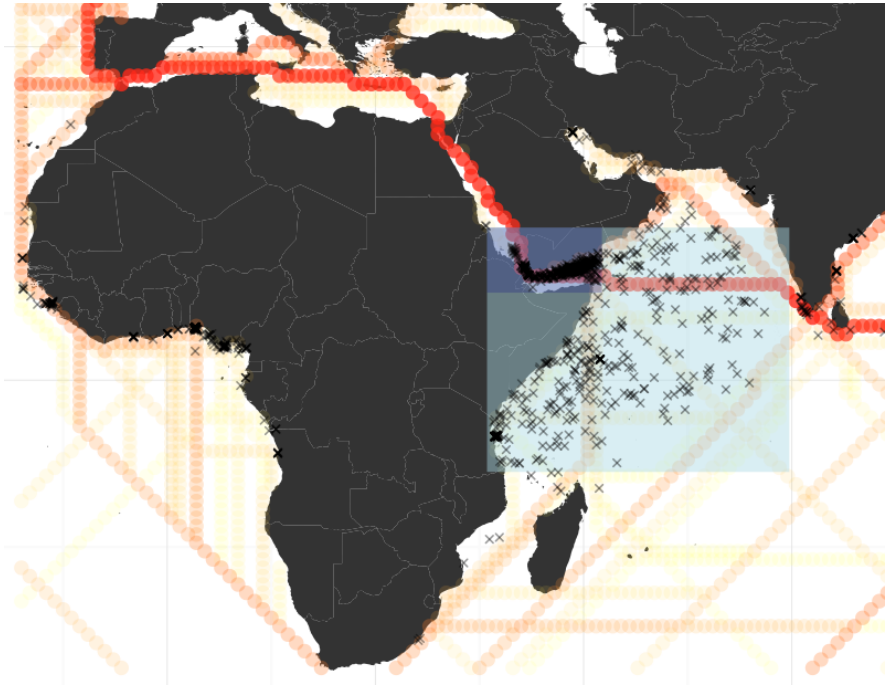


Figure: Piracy, Attacks and the Role of Weather, from Besley, Fetzer, Mueller (2014).

A good predictive model can...

Figure: Piracy, Attacks and the Role of Weather, from Besley, Fetzer, Mueller (2014).

A good predictive model can...

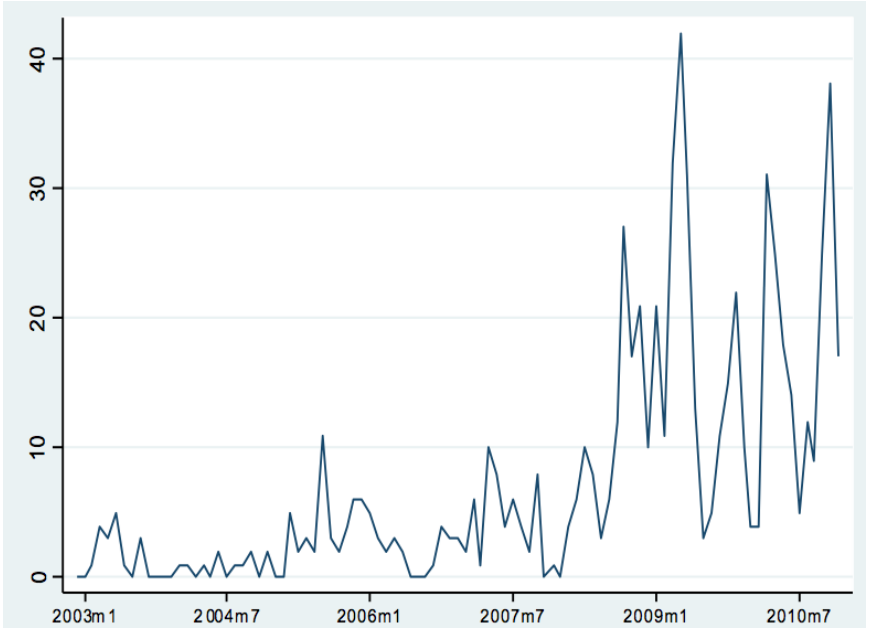


Figure: Piracy, Attacks and the Role of Weather, from Besley, Fetzer, Mueller (2014).

A good predictive model can...

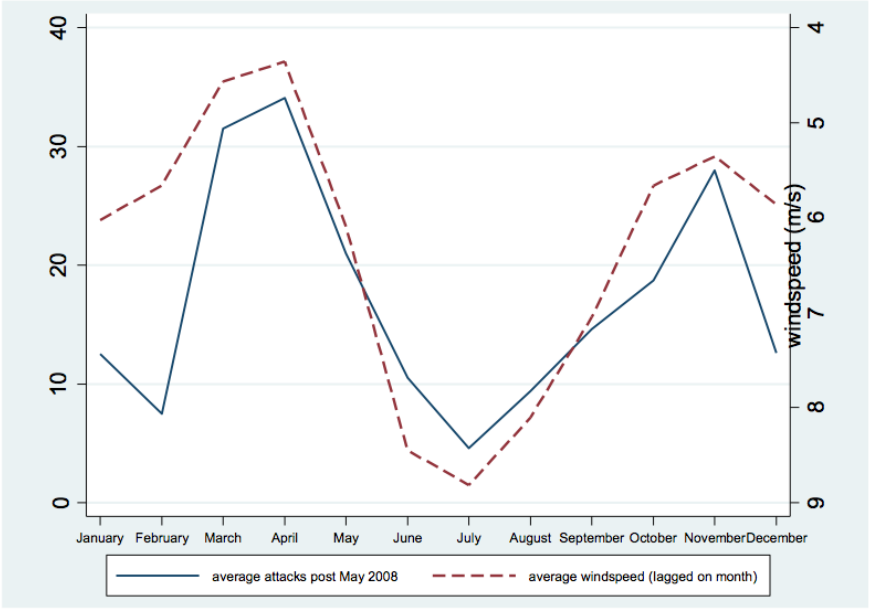
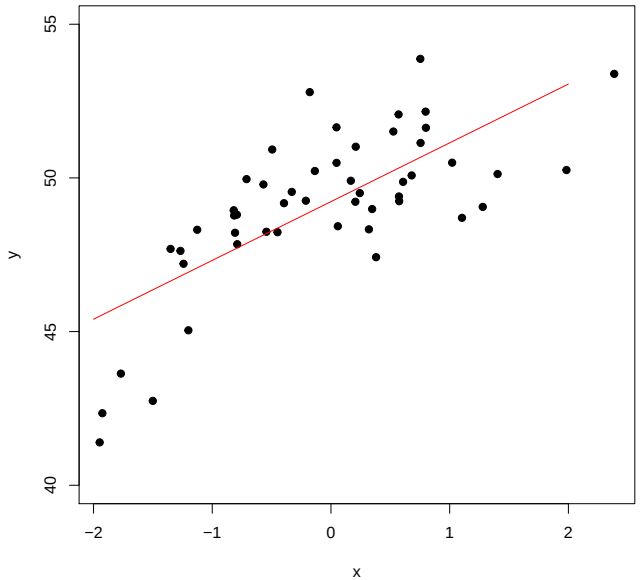
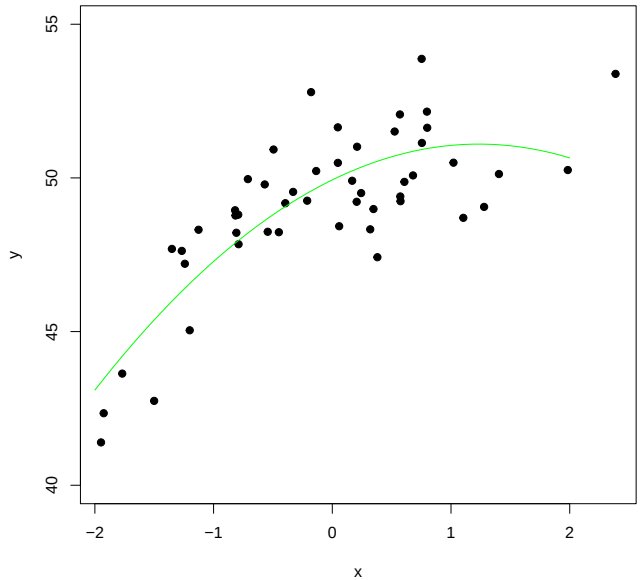
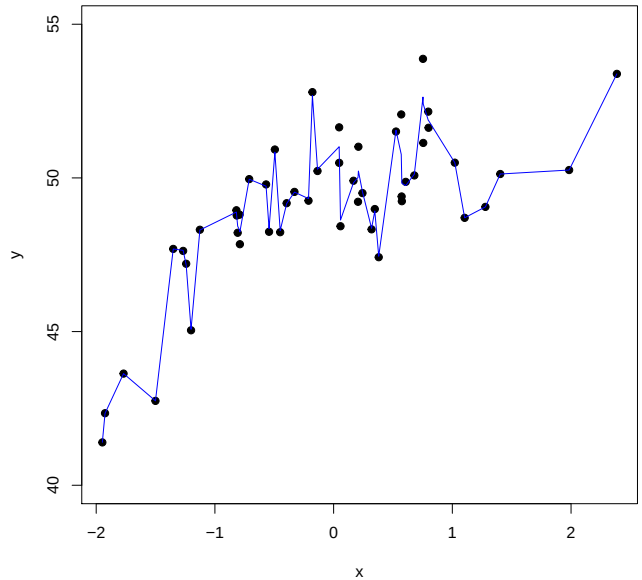


Figure: Piracy, Attacks and the Role of Weather, from Besley, Fetzer, Mueller (2014).

How to Estimate f?







A multitude of empirical methods: fitting linear, polynomials or a spline as an example.

Two types of methods

1. Parametric Methods
2. Non parametric approaches

Parametric Methods

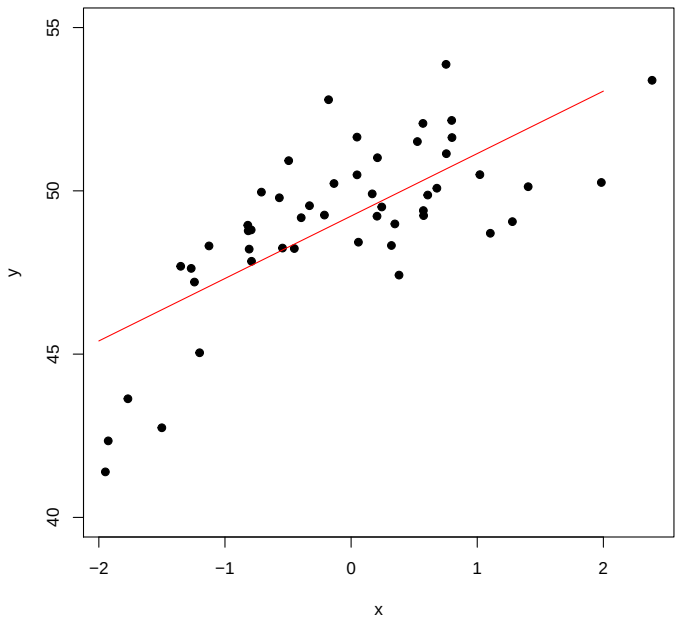
Parametric methods follow in two steps. (1) Assume the functional form of f, and then (2) fit the model using an appropriate method.

- The linear regression model has the form:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{1}$$

- Assumption: Regression function $E(Y|X)$ is linear or approximately linear, where β_j are to be estimated.
- Typically regression function is estimated using OLS.
- Not necessarily given, that the vector of parameters defining f is the correct model
- We will see later, that more flexible models with more parameters may result in **overfitting**.
- Overfitted models essentially explain the sample variation in y that is contained in the irreducible random error term ϵ, which leads to poor out of sample performance.

Parametric Methods: Linear regression



An example of fitting a linear regression, clearly there are quite some large errors.
Overall fit:

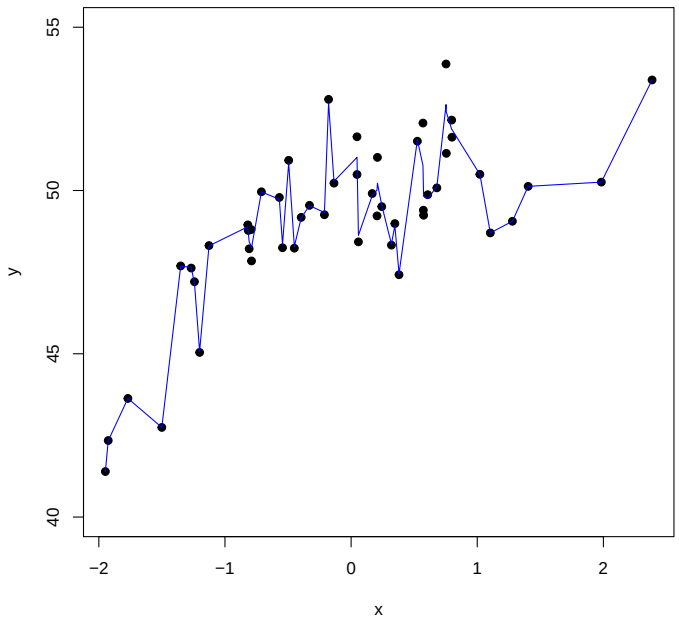
$$MSE = 1/n \sum_i \hat{e}_i{}^2 = 1/n \sum_i (\hat{y}_i - y_i)^2 =$$

$$= 3.08$$

Nonparametric Methods

- Non-parametric methods do not make explicit assumptions about the functional form of f.
- Economists generally dont like that too much, since f is a *black box*.
- Non parametric models try to fit f that gets *as close as possible* to the data points, without being too rough or wiggly.
- Typically, non parametric methods have a **tuning parameter**, that determines how smooth or wiggly the fit can get.

Nonparametric Methods: Spline

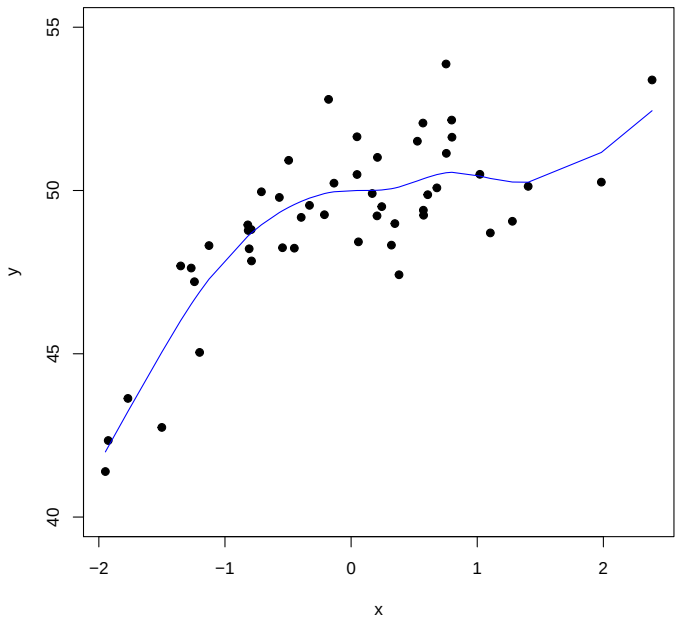


An example of fitting a spline fit, we get much closer to the data.
Overall fit:

$$MSE = 1/n \sum_i \hat{e}_i^2 = 1/n \sum_i (\hat{y}_i - y_i)^2 =$$

$$= 0.17$$

Nonparametric Methods: Spline (2)



An example of fitting a spline fit with a higher penality.
Overall fit:

$$MSE = 1/n \sum_i \hat{e}_i^2 = 1/n \sum_i (\hat{y}_i - y_i)^2 =$$

$$= 1.56$$

Trade-offs : No free lunch

- Flexible methods are much more difficult to interpret: can we say anything about what happens to y if x changes in the spline example?
- Restrictive models, such as linear models, are much more interpretable.
- In this course, we will discuss a range of methods and the applications one after the other.
- It will become evident, that there is no "one size fits all" approach and different situations require different methods.

Trade-offs visualized

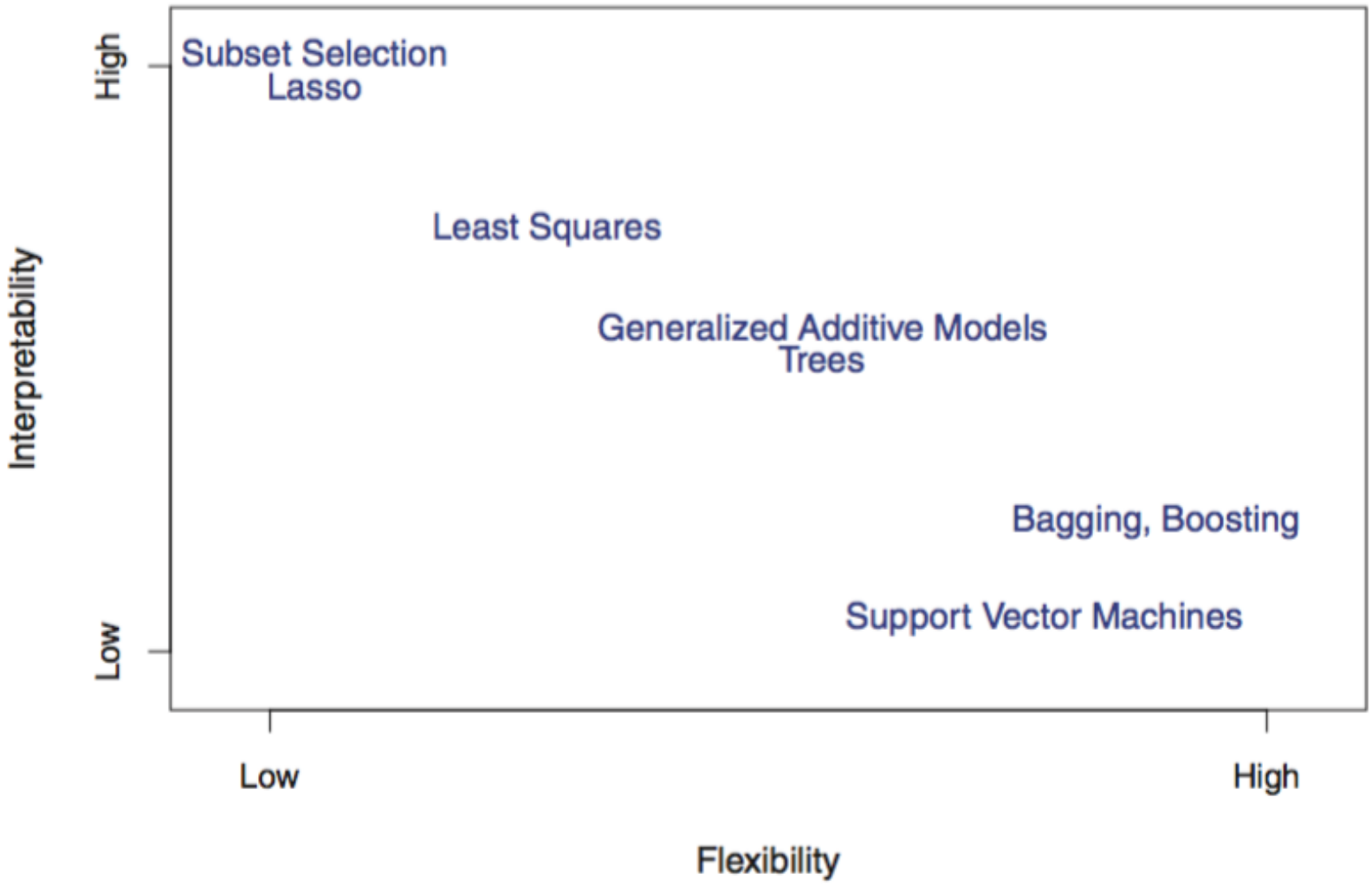


Figure: Trade off between flexibility and interpretability of different methods, from Hastie et al, 2013.

Set of Methods discussed for Numeric Prediction

- Lasso/ subset selection: essentially linear models, but getting rid of those whose $\beta_j \approx 0$
- Least squares: our fixed point, known from econometrics.
- Trees: divide and conquer
- Support Vector Machines

Supervised versus Unsupervised Learning

- **Supervised learning**

Supervised versus Unsupervised Learning

- **Supervised learning**

  Typically observe x_i, $i = 1, ..., n$ and some y_i, divide the data into a **training set** and an **test set**, estimate a model and then evaluate the fit.

Supervised versus Unsupervised Learning

- **Supervised learning**

  Typically observe x_i, $i = 1, ..., n$ and some y_i, divide the data into a **training set** and an **test set**, estimate a model and then evaluate the fit.

  Previous example of Somali piracy is a good example: use the fitted model to predict future $y_i's$ based on the observed wind speed patterns in the sea area.

Supervised versus Unsupervised Learning

- **Supervised learning**

  Typically observe x_i, $i = 1, ..., n$ and some y_i, divide the data into a **training set** and an **test set**, estimate a model and then evaluate the fit.

  Previous example of Somali piracy is a good example: use the fitted model to predict future $y_i's$ based on the observed wind speed patterns in the sea area.

- **Unsupervised learning**

Supervised versus Unsupervised Learning

- **Supervised learning**

  Typically observe x_i, $i = 1, ..., n$ and some y_i, divide the data into a **training set** and an **test set**, estimate a model and then evaluate the fit.

  Previous example of Somali piracy is a good example: use the fitted model to predict future $y_i's$ based on the observed wind speed patterns in the sea area.

- **Unsupervised learning**

  More challenging, since we observe x_i, $i = 1, ..., n$, but **do not know** or have no data on the underlying y_i's

Supervised versus Unsupervised Learning

- **Supervised learning**

  Typically observe x_i, $i = 1, ..., n$ and some y_i, divide the data into a **training set** and an **test set**, estimate a model and then evaluate the fit.

  Previous example of Somali piracy is a good example: use the fitted model to predict future $y_i's$ based on the observed wind speed patterns in the sea area.

- **Unsupervised learning**

  More challenging, since we observe x_i, $i = 1, ..., n$, but **do not know** or have no data on the underlying y_i's

  Online retailers observe somebody browsing on their website, they do not know much about the website visitor, but their behavior on the website may be used to infer how to induce him or her to become a customer (*behavioral clustering*).

Regression versus Classification

- Classification refers to cases where the y_i is a categorical variable, such as Eye color, Gender, Brand, Sentiment = Positive, Negative, Neutral.
- Regression refers to cases where the dependent variable is numeric, like price, quantity, ...
- There are cases, where categorical variables can be given a numeric interpretation, e.g. a categorical variable with two levels can be expressed as a dummy variable/ binary variable, such as Gender = 1 if male, Gender = 0 if female.

A Classification Example: Land Use Patterns

Figure: Classifiying pixels from satellite imagery into common land use clusters: Cropland, Forest and Shrubland as used in Fetzer and Marden (2015).

A Classification Example: Land Use Patterns

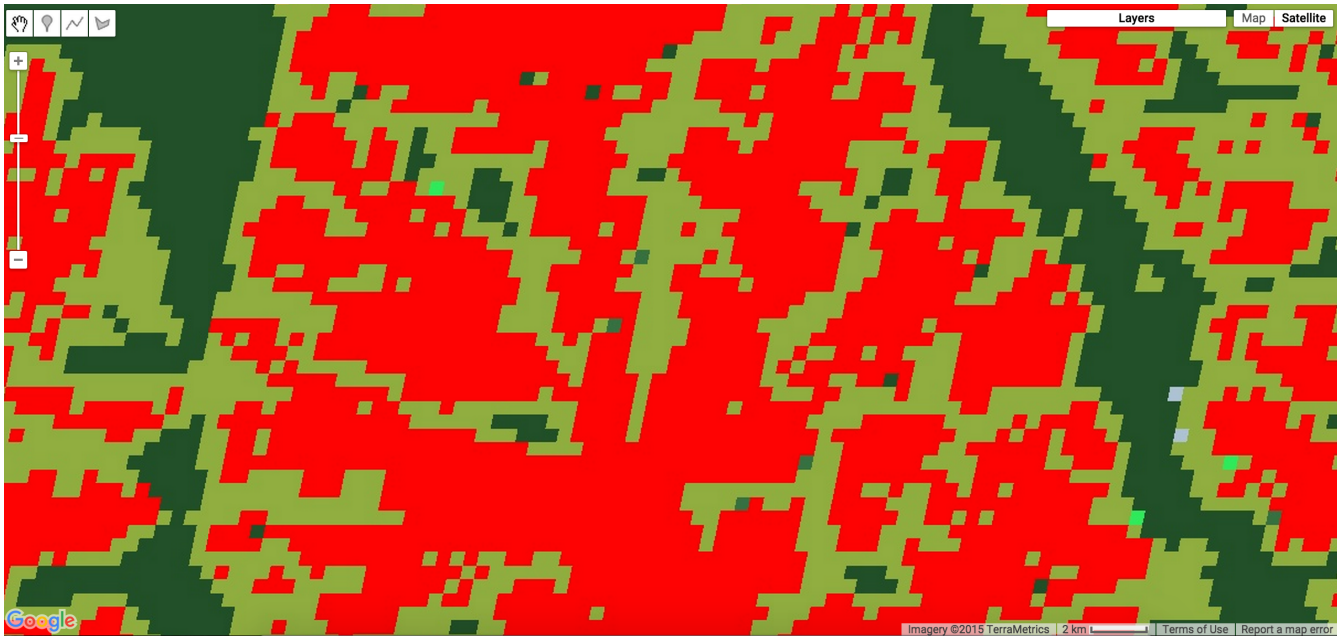


Figure: Classifiying pixels from satellite imagery into common land use clusters: Cropland, Forest and Shrubland as used in Fetzer and Marden (2015).

Plan

How to Assess Model Accuracy? In Theory...

- We need a way to measure and evaluate the relative performance of different statistical learning methods. In regression framework, the most commonly used objective is "mean squared error" (MSE), defined as:

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

- Assuming f, X are constant, MSE is an estimate of

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

- While the first part is "reducible" error that can be removed by improving the way we estimate the true f, the second component $Var(\epsilon)$ is "irreducible".

Validation Set Approach to Assess Model Accuracy

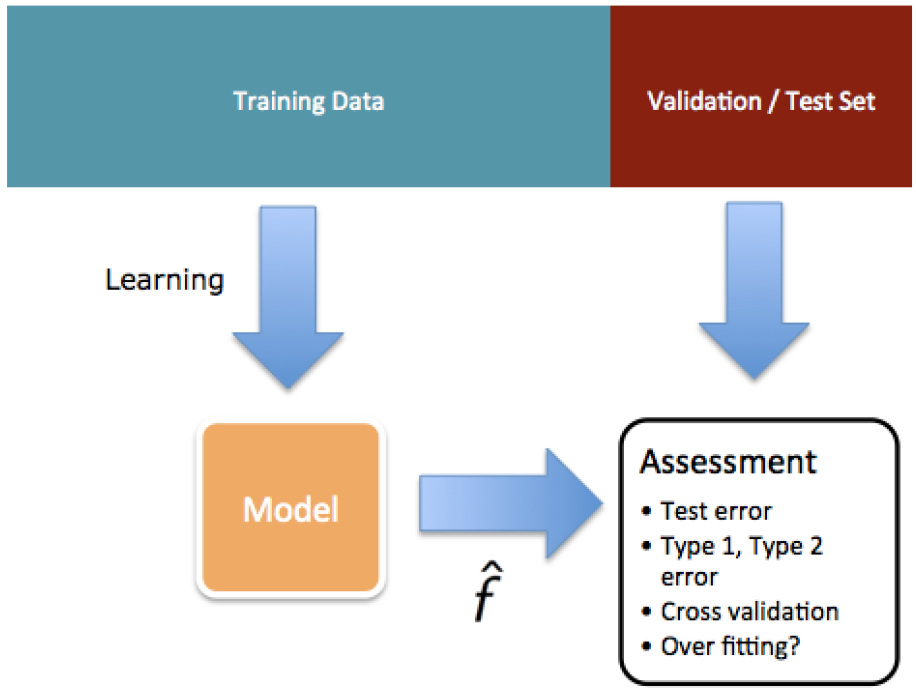


Figure: Validation Set Workflow Illustration.

Validation Set Approach to Assess Model Accuracy

- $\hat{f}$ is estimated using only a subset of the data, known the "training set". This is the part of the data that was used to estimate f. We compute the **training MSE**.
- On the rest of the data, the *test data*, we study how well the estimated $\hat{f}$ performs on virgin data that has not been used to fit the model. You can think of the test data as new data for which you want to *predict* the outcome.

  Based on this, we compute the **test MSE** and we would like this test MSE to be as small as possible: the smaller **test MSE**, the better is our predictive model.
- Its important that the training and test data are randomly selected subset of the data; otherwise, you may introduce systematic differences between the training dataset and the test dataset.

An Example: Training MSE and Model Complexity

- Create random variables x and noise ϵ.
- Divide the sample up into training data (100 obs) and test data (50 obs) by taking a random subset. This is called the "validation set" approach.
- True relationship between x and y given as:

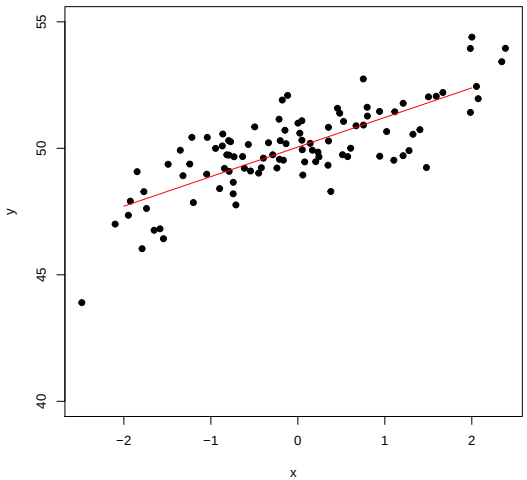
$$f(x) = 1 + 0.5x - 0.1x^2 + 0.25x^3$$

- We will add some noise that is a random variable with distribution $\epsilon \sim N(0, 1)$.
- What does this tell us about "the best possible model fit", i.e. the *best possible test MSE* we can achieve?
- So "best" true model should be a polynomial of third order...
- We will fit polyonomials of ever higher order and see how the Training versus the Test MSE evolve...

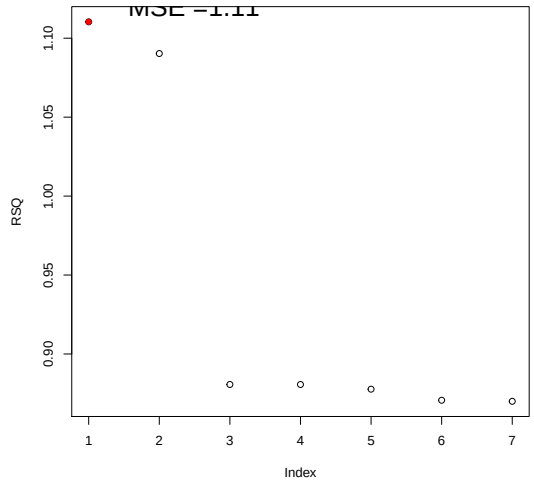
See `polynomials.R`

Training MSE versus test MSE Evolution

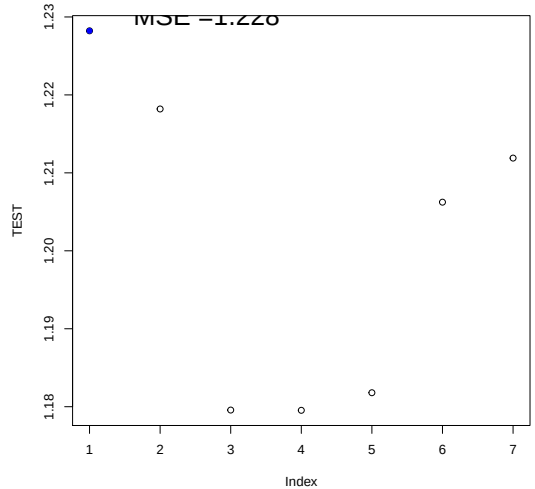
Polynomial of order 1



Training MSE



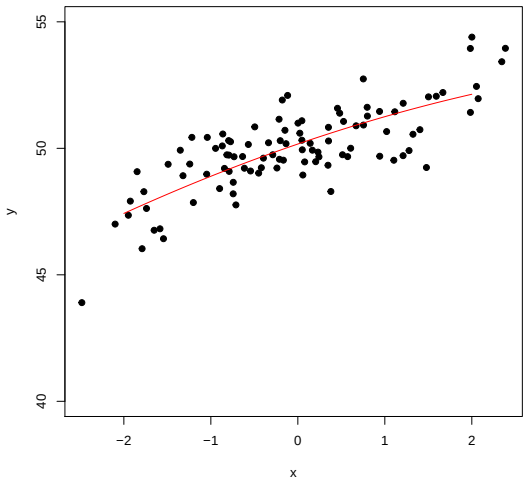
Test MSE



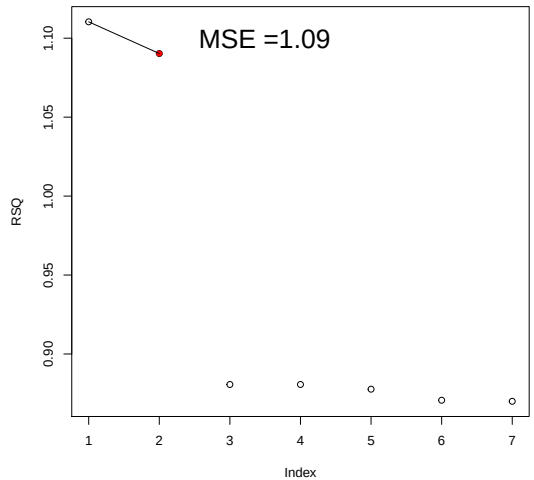
This is fitting a 1-th order polynomial to the scatterplot on the left. As we increase the order, the **training MSE** decreases monotonically, while the **test MSE** stops decreasing after a certain point. We are *overfitting* the data.

Training MSE versus test MSE Evolution

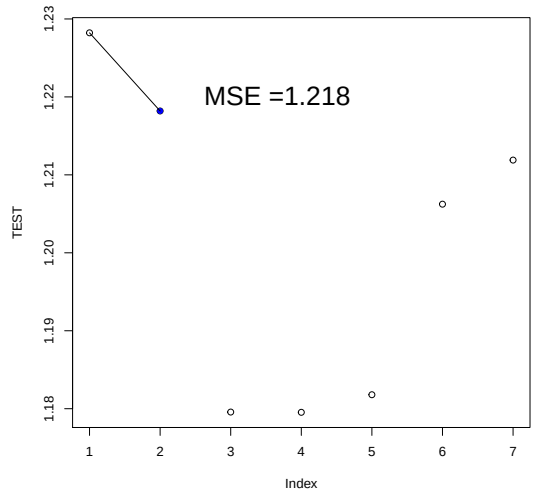
Polynomial of order 2



Training MSE



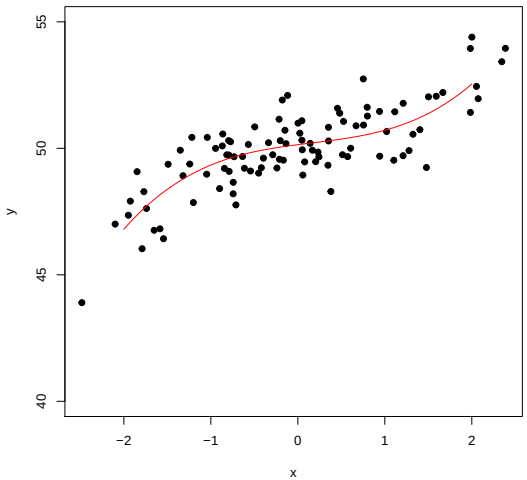
Test MSE



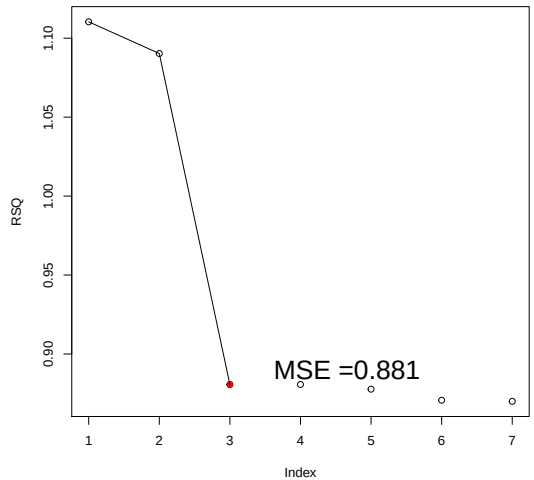
This is fitting a 2-th order polynomial to the scatterplot on the left. As we increase the order, the **training MSE** decreases monotonically, while the **test MSE** stops decreasing after a certain point. We are *overfitting* the data.

Training MSE versus test MSE Evolution

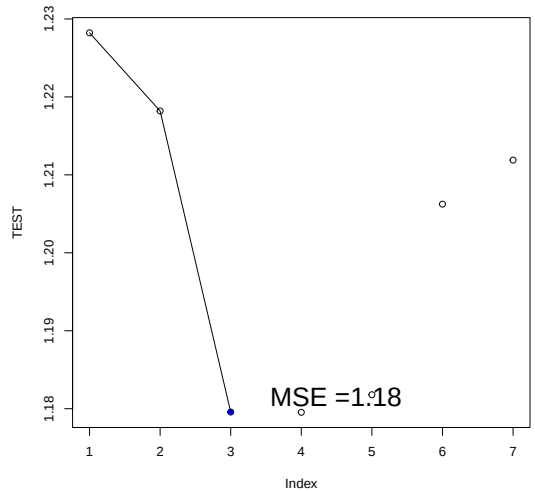
Polynomial of order 3



Training MSE



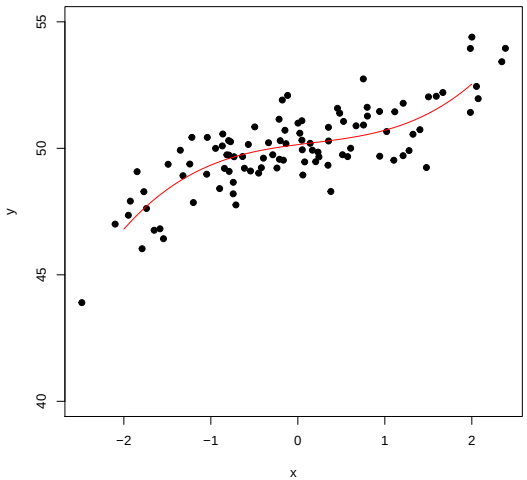
Test MSE



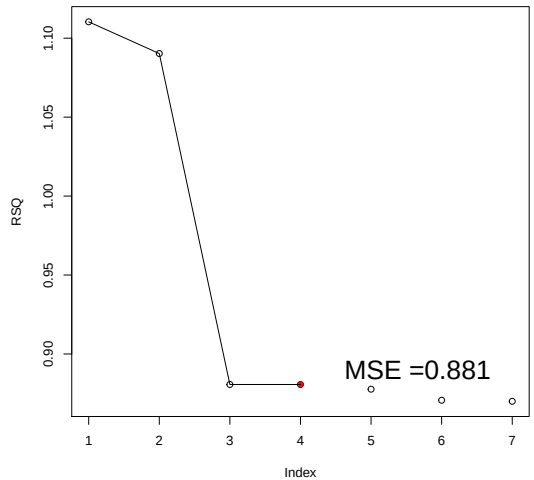
This is fitting a 3-th order polynomial to the scatterplot on the left. As we increase the order, the **training MSE** decreases monotonically, while the **test MSE** stops decreasing after a certain point. We are *overfitting* the data.

Training MSE versus test MSE Evolution

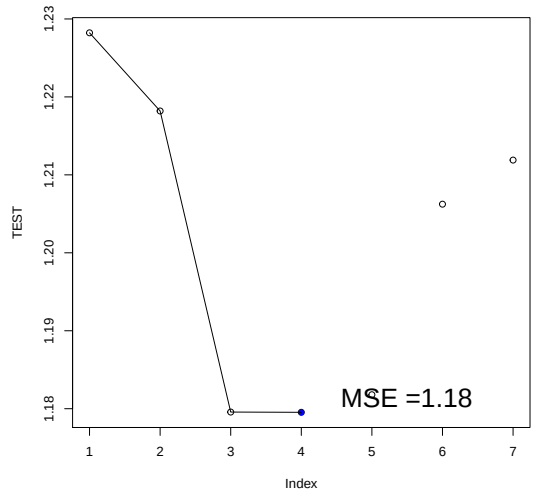
Polynomial of order 4



Training MSE



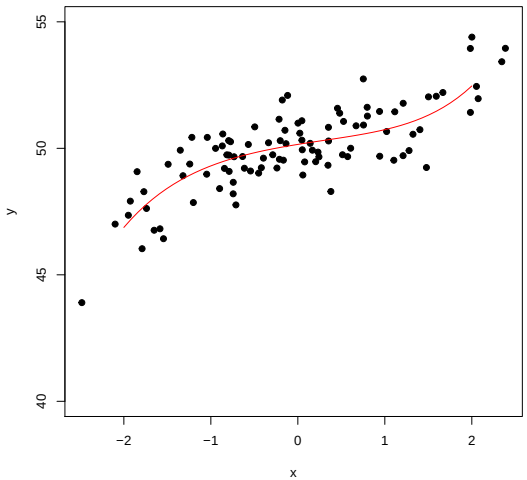
Test MSE



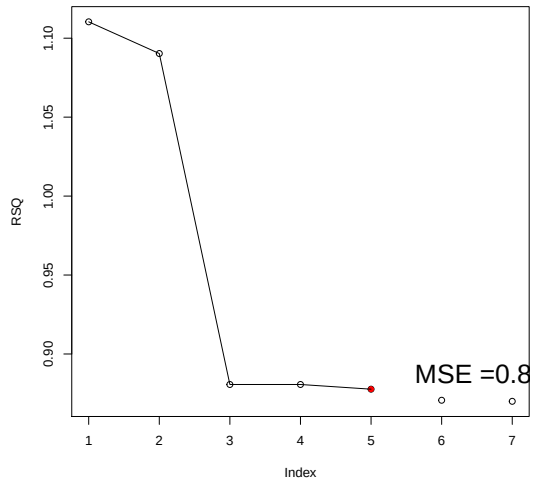
This is fitting a 4-th order polynomial to the scatterplot on the left. As we increase the order, the **training MSE** decreases monotonically, while the **test MSE** stops decreasing after a certain point. We are *overfitting* the data.

Training MSE versus test MSE Evolution

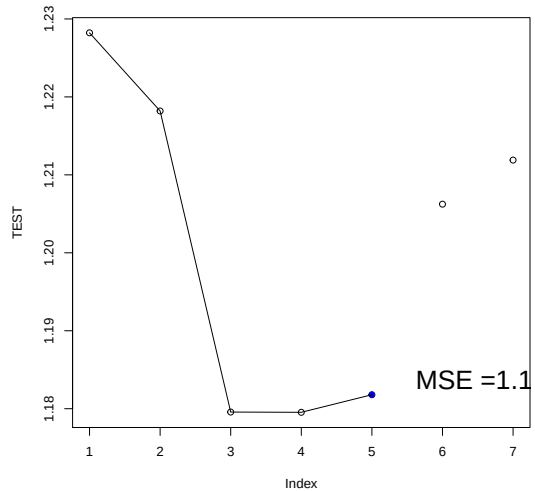
Polynomial of order 5



Training MSE



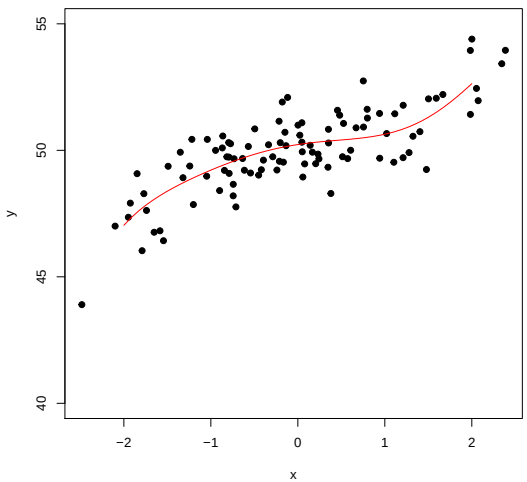
Test MSE



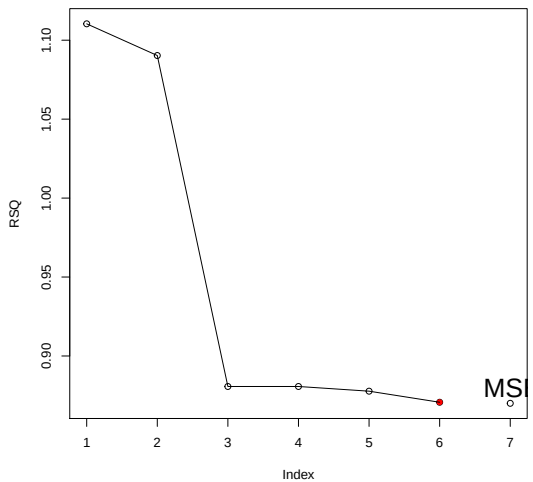
This is fitting a 5-th order polynomial to the scatterplot on the left. As we increase the order, the **training MSE** decreases monotonically, while the **test MSE** stops decreasing after a certain point. We are *overfitting* the data.

Training MSE versus test MSE Evolution

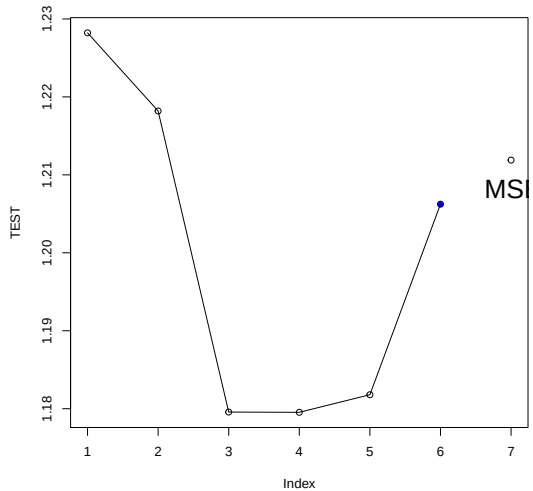
Polynomial of order 6



Training MSE



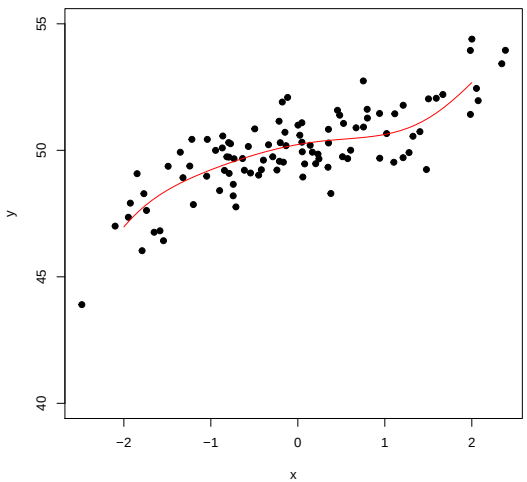
Test MSE



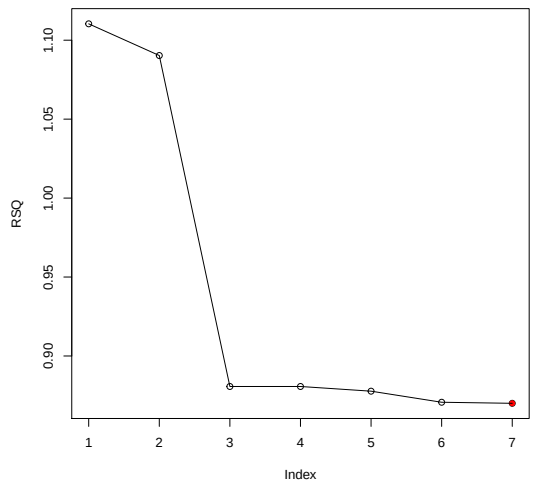
This is fitting a 6-th order polynomial to the scatterplot on the left. As we increase the order, the **training MSE** decreases monotonically, while the **test MSE** stops decreasing after a certain point. We are *overfitting* the data.

Training MSE versus test MSE Evolution

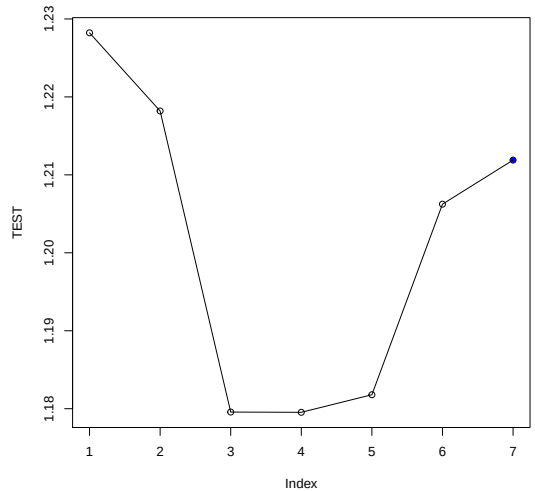
Polynomial of order 7



Training MSE



Test MSE



This is fitting a 7-th order polynomial to the scatterplot on the left. As we increase the order, the **training MSE** decreases monotonically, while the **test MSE** stops decreasing after a certain point. We are *overfitting* the data.

Overfitting...why does this happen?

- In the given example, we know that the minimal attainable test MSE =1, since the irreducible error has variance 1, given $\epsilon \sim N(0, 1)$.
- Fitting ever more complicated polynomials, while ignoring the true model implies, that the estimated models are starting to **explain the noise** contained in ϵ.
- From Econometrics: including *irrelevant variables* (that is those with coefficients ≈ 0) does not result in biased point estimates, but the resulting estimators are not *efficient*, i.e. OLS is not BLUE.
- Since the noise is randomly drawn and thus, the noise in the training set is **independent** from the noise in the test set, the performance of the fit estimated from the data in the training set will become worse and worse

Overfitting...why does this happen?

- Its important that you do not look at the test set, as otherwise this independence assumption may be violated, if you eg. adjust the set model to capture some features in the training set.
- In Applied Economics, people generally only look at models explaining a given data set (the training data), but do not assess model performance out of sample.
- In reality, we do not know what the true function for f is...hence, in order to assess whether we are overfitting the data, we need to study test MSE as we try different methods of estimating f.
- The validation set approach selects one set of training data and one set of test data; *cross-validation* is an extension to the validation set approach, which we will look at in the lecture on Shrinkage methods.
- In our example, the test MSE is minimal for a 3rd/4th order polynomial... so we are getting quite close to the true model.

Plan

Bias vs Variance Tradeoff

- We can more formally describe what is happening to the test MSE as model complexity increases.
- The U-shape of the test MSE is driven by two competing forces
- The expected test MSE at any different point x_0 can be decomposed as:

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

where $Bias(\hat{f}(x_0)) = E(f(x_0) - \hat{f}(x_0))$. Can you show this?

Bias vs Variance Tradeoff Proof

Show that:

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

Proof:

- To get rid of the indices, let $f = f(x_0)$, and $\hat{f} = \hat{f}(x_0)$.
- Remember that for some random variable X, $Var(X) = E[(X - E(X))^2] = E(X)^2 - E(X^2)$, so in particular $Var(\epsilon) = E(\epsilon^2)$, since $E[\epsilon] = 0$.
- Further: $E(y) = f$, since $y = f + \epsilon$ and $E(\epsilon) = 0$ and f is itself not a random variable.
- This implies $Var(y) = E[(y - E(y))^2] = E[(f + \epsilon - f)^2] = E[\epsilon^2] = Var(\epsilon)$
- Also remember that for some random variables X, Y: $Cov(x, Y) = E(XY) - E(X) * E(Y)$
- So in particular: $E(\hat{f}\epsilon) = 0$, since ϵ from test set is randomly and independently drawn from prediction $\hat{f}$.

Together this yields

$$\begin{aligned}
\mathrm{E}\big[(y - \hat{f})^2\big] &= \mathrm{E}[y^2 + \hat{f}^2 - 2y\hat{f}] \\
&= \mathrm{E}[y^2] + \mathrm{E}[\hat{f}^2] - \mathrm{E}[2y\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{E}[y]^2 + \mathrm{Var}[\hat{f}] + \mathrm{E}[\hat{f}]^2 - 2f\mathrm{E}[\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{Var}[\hat{f}] + (f - \mathrm{E}[\hat{f}])^2 \\
&= \mathrm{Var}[\epsilon] + \mathrm{Var}[\hat{f}] + [\mathrm{E}(f - \hat{f})]^2
\end{aligned}$$

Bias vs Variance Intuition

$$E(f(x_0) - f(\hat{x}_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

Look at the individual elements here:
$Var(\epsilon)$

... is a constant.

it remains unchanged for different $\hat{f}$'s.

it represents the lowest bound for a test error that is attainable, since both the other terms are positive.

Minimizing test error requires finding an $\hat{f}$ that minimizes the sum between squared bias and variance.

Bias vs Variance Intuition

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

Look at the individual elements here:
$Var(\hat{f}(x_0))$

... refers to the amount by which $\hat{f}$ would change if we estimated it *using a different training data set*.

Since the training data are used to fit the statistical learning method, different training data sets produce different $\hat{f}$. Ideally the estimate for f should not vary too much between training sets.

Different methods have different variances: more flexible methods have larger variances, while less flexible ones (e.g. linear regression) have lower variance.

This is pushing up our test MSE for highly flexible specifications.

Bias vs Variance Intuition

$$E(f(x_0) - f(\hat{x}_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

Look at the individual elements here:
$[Bias(\hat{f}(x_0))]^2$

... an approximate model, that leaves our relevant factors systematically introduces errors by not allowing e.g. for more complex interactions between variables X_i.

e.g. a linear model may be inadequate in case the true relationship is non-linear, introducing significant bias.

This is akin to the idea of ommitted variable bias in regression, which causes the true effect of some variable to be under or over-stated, thus, distorting the predictive power of that variable.

Bias vs Variance Intuition

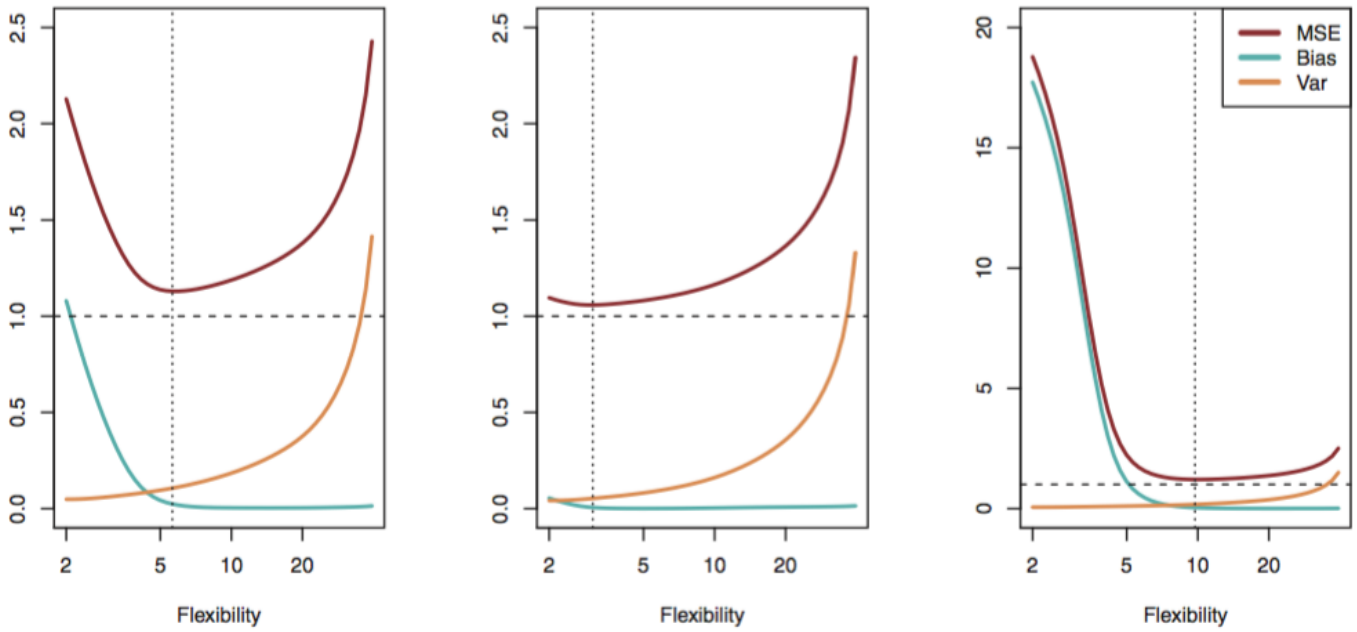


Figure: Bias-Variance tradeoff illustrated: U-shape due to increasing variance at high level of model flexibility. Taken from Hastie et al., 2013.

Linear Methods

- Pick up straight from econometrics with linear methods; many of the non-linear methods presented later are generalizations of the basic linear models.
- Linear methods often have low bias, but high variance resulting in bad out of sample performance.
- We want to find ways of increasing out of sample performance for prediction, by identifying covariates that actually "matter a lot".

Plan

Linear Regression Revisited

- The linear regression model gas the form:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{2}$$

- Assumption: Regression function $E(Y|X)$ is linear or approximately linear, where β_j are to be estimated.
- Typically the X_j's come from different sources:
  - quantitative inputs and transformations of them (eg. log)
  - polynomial base expansions, e.g. for two $p = 2$, a second order expansion would imply a total of five regressors $X_1, X_1^2, X_1 * X_2, X_2^2, X_2$
  - Numeric categorical variables (e.g. points on a Likert scale).
- Assume X has dimensions $N \times p$, i.e. N rows and p columns (regressors).

Least squares fit

- Least square fit solves the following optimization problem

$$argmin_{\beta} RSS(\beta) = (y - \mathbf{X}\beta)'(y - \mathbf{X}\beta) \tag{3}$$

- This is a matrix way of writing: find the vector $\beta = (\beta_1, ..., \beta_p)$, such that:

$$\sum_{i=1}^{n} (y_i - \sum_{k} x_k \beta_k)^2$$

  is minimized.
- Solution obtain by solving FOC:

$$\frac{\partial RSS}{\partial \beta} = -2\mathbf{X}'(y - X\beta)$$

- and setting equal to zero

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'y$$

Least squares fit visually

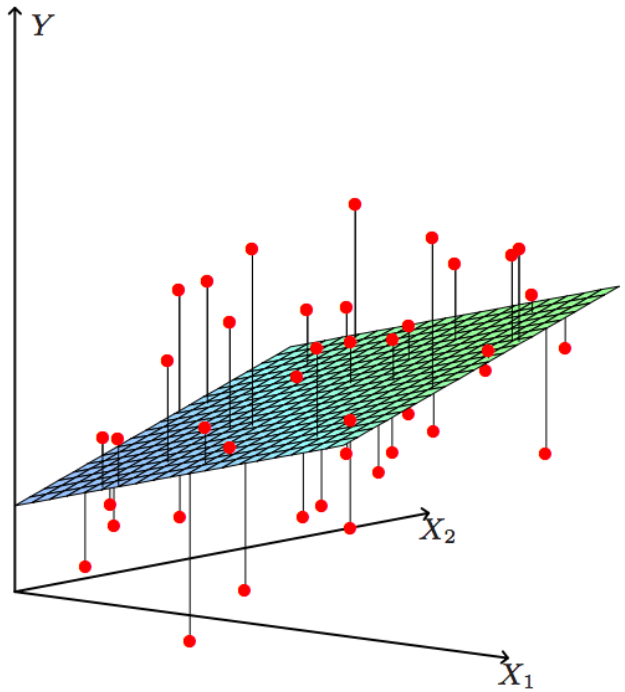


Figure: Three dimensional linear regression visualized.

Some properties of the fit

- Least squares solution is the best linear unbiased estimator if the y's are conditionally indepedent for a given set of inputs x_i (Gauss Markov Theorem)
- In case of homoskedastic errors, $Var(\hat{\beta}) = (X'X)^{-1}\sigma^2$
- The σ^2 is typically estimated by

$$\hat{\sigma}^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- It can be shown that $E(\hat{\sigma^2}) = \sigma^2$, i.e. $\hat{\sigma^2}$ is an unbiased estimator of the population variance.
- With normally distributed error term, i.e. $\epsilon \sim N(0, \sigma^2)$, one can show that

$$\hat{\beta} \sim N(\beta, (X'X)^{-1}\sigma^2)$$

- The distribution of β can be estimated, when plugging in an estimate of σ^2. This can be used for hypothesis testing on the effects of β_i's.

Projection visually represented

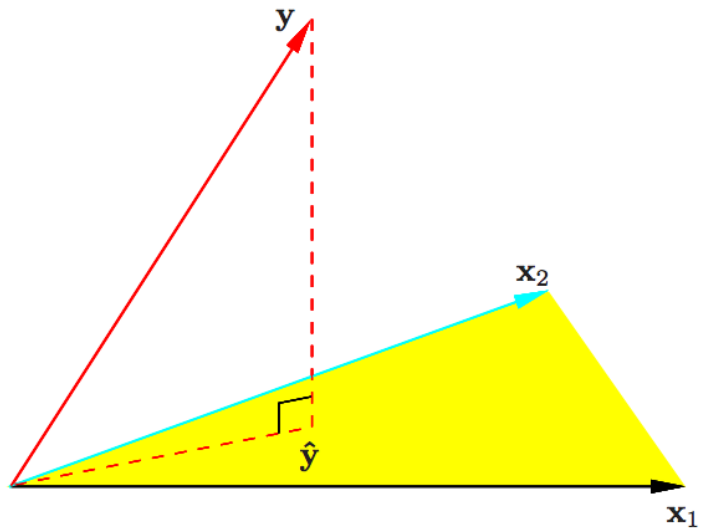


Figure: Three dimensional linear regression visualized.

- Fitted values from a regression are given as $\hat{y} = X\hat{\beta} = X(X'X)^{-1}X'y$
- The matrix $X(X'X)^{-1}X'$ has dimensions $N \times N$ and projects the Nx1 vector y into the p dimensional subspace

Further properties of the fit

- We can express the variability in y as

$$TSS = \sum (y_i - \bar{y})^2$$

- Of which a regression explains a proportion

$$ESS = \sum (\hat{y}_i - \bar{y})^2$$

- And leaves unexplain

$$RSS = \sum (y_i - \hat{y}_i)^2$$

- $TSS = ESS + RSS$ [Do you remember how to show this?]
- A measure of goodness of fit is given as

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

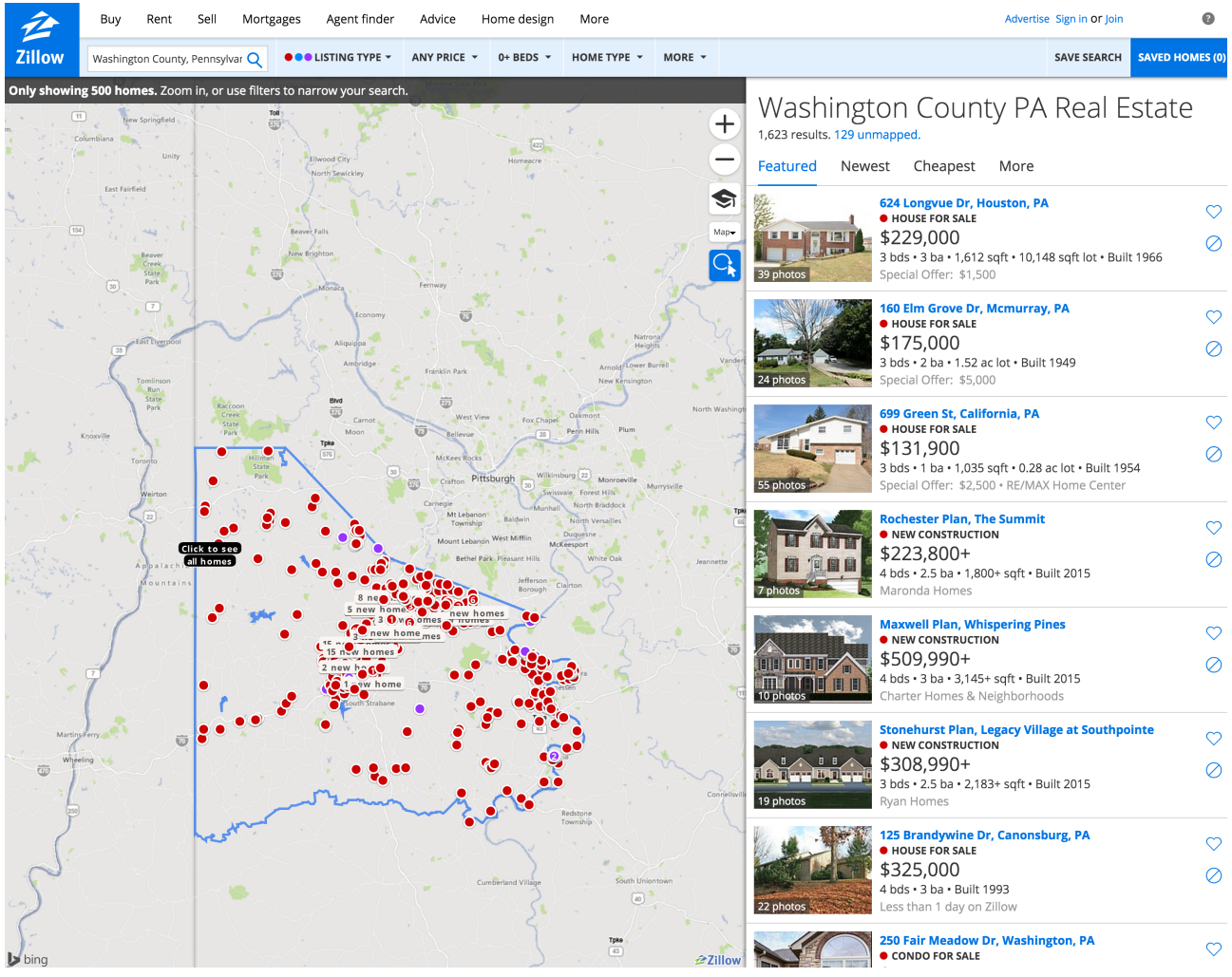
- Its hard to say, what is a "good" R^2 - why? RSS only knows one direction... it goes down as we include more controls, even if these controls dont have any empirical content. They simply fit the noise in the error term ϵ.

Application: Hedonic Pricing Model of Property Prices

- Illustrate linar regression and prediction performance for a pricing model of real estate.
- The idea is that a house can be decomposed into characteristics such as number of bedrooms, size of lot, or distance to the city center,...
- A hedonic regression equation treats these attributes (or bundles of attributes) separately, and estimates prices
- This is important for people who want to work in companies like...

- This information can be used to construct a price index that can be used to compare the price of housing in different cities, or to do time series analysis.

An example: predicting property prices



Obtaining some data

- In fact, you are not really allowed to use this data as its Zillow proprietary information.
- Characteristics to look at are: Price that a house sold for, its footage (size), number of bathrooms, number of bedrooms and when it was built, latitude and longitude as geographic information.
- Most of the properties are located in the counties of Pennsylvania and were sold between 2010-2013.
- A lot of such data can be web scraped... in fact, a lot of economists tend to hire RAs doing web scraping.

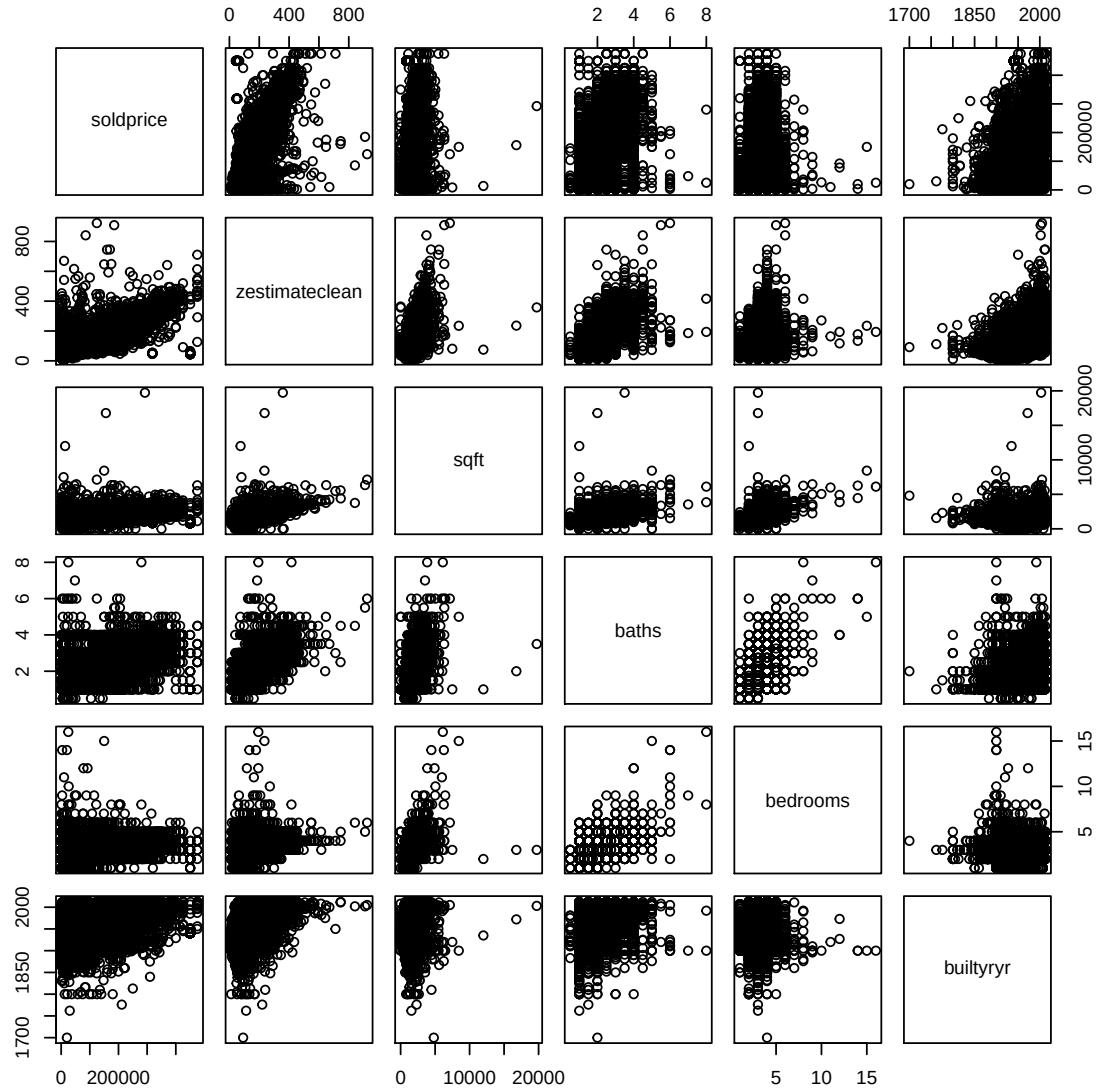
Exploring the data: Summary Statistics

```
set.seed(1312)
SAMPLE<-sample(1:nrow(HOUSES),18000)
summary(HOUSES[SAMPLE,c("soldprice","zestimateclean","sqft","baths","bedrooms","builtyryr", "distgas2010")
```

```
##    soldprice      zestimateclean      sqft          baths         bedrooms
##  Min.   :  1018   Min.   : 11    Min.   :    1   Min.   :0.50   Min.   : 1.00
##  1st Qu.: 25000   1st Qu.: 64    1st Qu.: 1108   1st Qu.:1.00   1st Qu.: 2.00
##  Median : 93000   Median :114    Median : 1409   Median :1.50   Median : 3.00
##  Mean   :109101   Mean   :131    Mean   : 1547   Mean   :1.77   Mean   : 2.98
##  3rd Qu.:163500   3rd Qu.:173    3rd Qu.: 1850   3rd Qu.:2.50   3rd Qu.: 3.00
##  Max.   :475871   Max.   :923    Max.   :19732   Max.   :8.00   Max.   :16.00
##    builtyryr     distgas2010
##  Min.   :1700   Min.   :   73
##  1st Qu.:1925   1st Qu.: 4087
##  Median :1953   Median : 7047
##  Mean   :1952   Mean   :10301
##  3rd Qu.:1977   3rd Qu.:10964
##  Max.   :2013   Max.   :67289
```

Exploring the data: Correlograms

```
pairs(~soldprice+zestimateclean+sqft+baths+bedrooms+builtyryr, data=HOUSES[SAMPLE])
```



An example: predicting property prices

- A linear hedonic pricing model can be as follows:

$$\text{SalePrice}_i = \beta_0 + \beta_1 SQFT_i + \beta_2 baths_i + \beta_3 bedrooms_i + \beta_4 builtyr_i + \epsilon_i$$

- We can estimate this model via OLS.

An example: predicting property prices

Some linear regression output...

```
results<-lm(soldprice~sqft+baths+bedrooms+builtyryr, data=HOUSES[SAMPLE])
summary(results)
```

```
##
## Call:
## lm(formula = soldprice ~ sqft + baths + bedrooms + builtyryr,
##     data = HOUSES[SAMPLE])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -449932  -35386   -4314   32312  403301
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.18e+06   3.22e+04  -67.79   <2e-16 ***
## sqft         2.82e+01   9.89e-01   28.55   <2e-16 ***
## baths        3.62e+04   8.98e+02   40.26   <2e-16 ***
## bedrooms    -8.14e+02   7.32e+02   -1.11     0.27
## builtyryr    1.12e+03   1.67e+01   67.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64400 on 17995 degrees of freedom
## Multiple R-squared:  0.521, Adjusted R-squared:  0.521
## F-statistic: 4.9e+03 on 4 and 17995 DF,  p-value: <2e-16
```

Some Important Questions

1. Is at least one of the predictors $X_1, ..., X_n$ useful in predicting the response?
2. Do all predictors help explain Y, or is only a subset useful?
3. How well do we fit the data? How much is left unexplained?
4. How well do we predict?

Is at least one of the predictors useful in predicting the response?

This boils down to testint the hypothesis:

$$H_0 : \beta_1 = \beta_2 = ... = \beta_p = 0$$

What is H_1? You will remember that the way to perform this test is to look at the F-statistic, computed as:

$$F = \frac{(TSS - RSS)/p}{RSS/(n - p - 1)}$$

Remember $TSS = \sum (y_i - \bar{y})^2$ and $RSS = \sum (y_i - \hat{y}_i)^2$ and $TSS = ESS + RSS$
You can show that

$$E[RSS/(n - p - 1)] = \sigma^2$$

Further, provided that H_0 is true:

$$E[(TSS - RSS)/p] = \sigma^2$$

[Why is that?] So if H_0 is true, then the test F statistic should be close to 1...

Is at least one of the predictors useful in predicting the response?

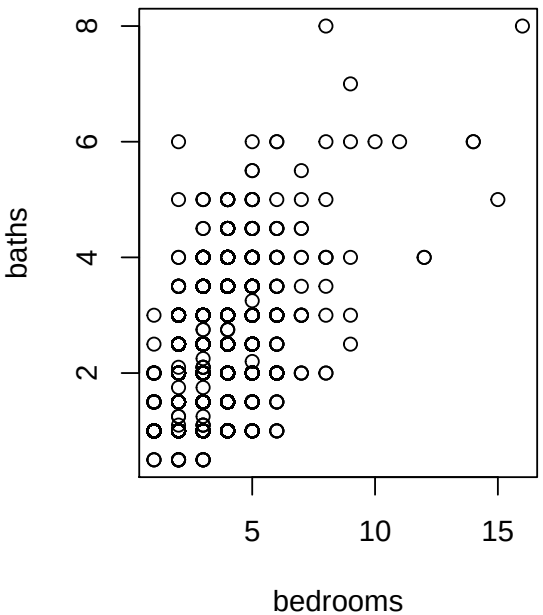
```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.18e+06   3.22e+04  -67.79   <2e-16 ***
## sqft         2.82e+01   9.89e-01   28.55   <2e-16 ***
## baths        3.62e+04   8.98e+02   40.26   <2e-16 ***
## bedrooms    -8.14e+02   7.32e+02   -1.11     0.27
## builtyryr    1.12e+03   1.67e+01   67.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64400 on 17995 degrees of freedom
## Multiple R-squared:  0.521, Adjusted R-squared:  0.521
## F-statistic: 4.9e+03 on 4 and 17995 DF,  p-value: <2e-16
```

So can safely reject H_0.

Do all predictors help explain Y, or is only a subset useful?

It turns out, bedrooms dont really predict prices. Why may that be?

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.18e+06   3.22e+04  -67.79   <2e-16 ***
## sqft         2.82e+01   9.89e-01   28.55   <2e-16 ***
## baths        3.62e+04   8.98e+02   40.26   <2e-16 ***
## bedrooms    -8.14e+02   7.32e+02   -1.11     0.27
## builtyryr    1.12e+03   1.67e+01   67.20   <2e-16 ***
##           bedrooms baths
## bedrooms     1.000 0.514
## baths        0.514 1.000
```



How do we perform in terms of prediction?

In linear regression, one measure of goodness of fit is the training R^2.

```
##
## Residual standard error: 64400 on 17995 degrees of freedom
## Multiple R-squared:  0.521, Adjusted R-squared:  0.521
## F-statistic: 4.9e+03 on 4 and 17995 DF,  p-value: <2e-16
```

We are getting an R^2 of around 0.5. How do we perform in terms of out of sample prediction? Not so good.

```
##       sample      MSE
## 1: Training 4.15e+09
## 2:     Test 4.15e+09
```

But what does good mean?
In fact, we can introduce a whole range of further variables X_i, such as locational information, distance to amenities, ... in the next sections, we discuss some algorithms that help us select between more or less complicated models.

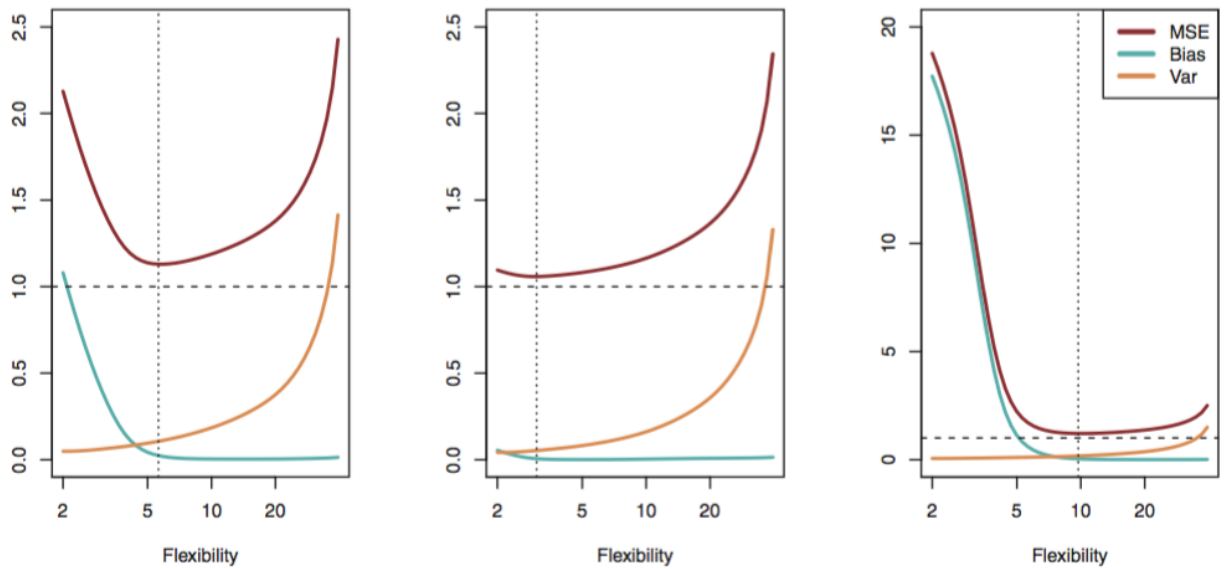
Plan

Model Selection

We are primarily concerned with *prediction accuracy* versus *model interpretability*. In some cases, we may have to trade-off one for the other; in other cases, we can achieve both objectives.

1. Prediction Accuracy

   If the true relationship is linear, then OLS will have low bias; if n becomes very large relative to p, the estimates are precise and model fit is good. Otherwise, model fit has high variance; in this case, we may want to trade off some bias.

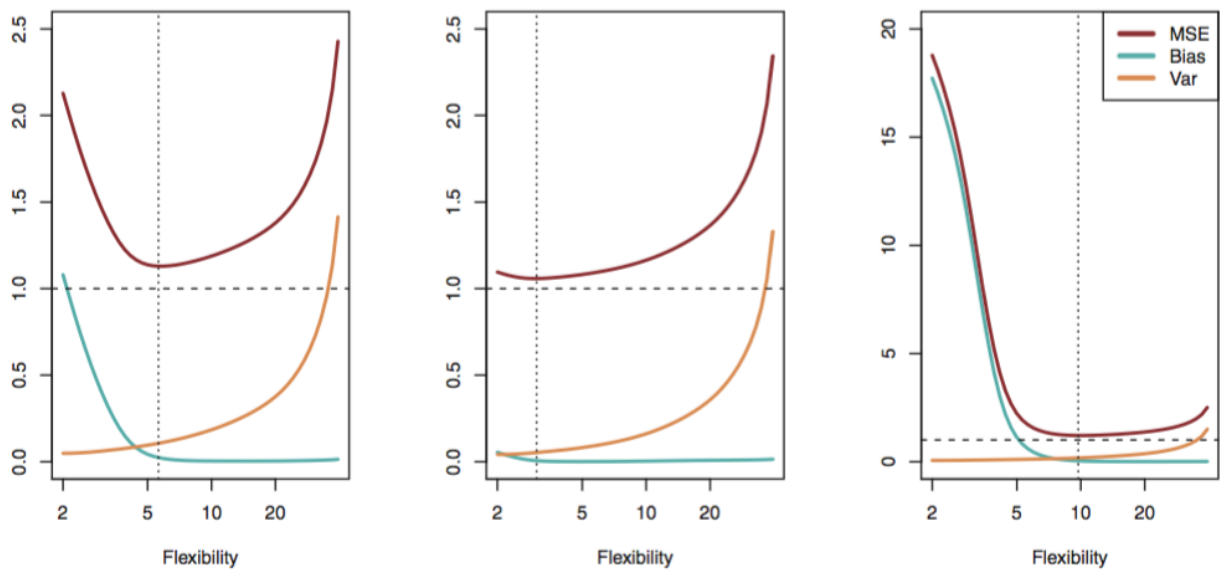


Model Selection

We are primarily concerned with *prediction accuracy* versus *model interpretability*. In some cases, we may have to trade-off one for the other; in other cases, we can achieve both objectives.

2. Model Interpretability

   Often time we include variables in our regressions, that are not associated with the response variable. This holds true, in particular, if we control for various interaction terms or polynomials of variables. The more variables, the more difficult it is to interpret a model... but how do we identify "irrelevant variables"?



Model Selection

We are going to discuss three different methods.

1. Subset selection

   Different (algorithmic) approaches to identify a subset of our predictors p, which we believe to be driving the variation in y

   - Best subset selection
   - Forward / backward

Model Selection

We are going to discuss three different methods.

2. Shrinkage

   Fitting a model involving all predictors p, but penalizing coefficients on predictors that are "not important". Depending on the shrinkage method, we can remove some predictors completely, as their coefficient β_i is estimated to be exactly zero.

   - Ridge regression
   - Lasso

Model Selection

We are going to discuss three different methods.

3. Dimensionality reduction

   Maybe all the information in some variables X_1 and X_2 is already contained in some variable X_3; this could be mechanic or by association. The method we discuss in the last section of the course is Principal Component Analysis.

   - Principal Components regression

Subset selection

Suppose you have p predictors, that is $X_1, ..., X_p$ and you want to identify the best subset of predictors M, possibly with $M < p$, that achieve "best performance".
This is a difficult task...why?

- There are p models, containing exactly 1 predictor
- There are $\binom{p}{2} = p(p-1)/2$ possible ways to choose 2 predictors [Remember: $\binom{p}{k} = \frac{p!}{k!(p-k)!}$]
- There are $\binom{p}{3} = \frac{p!}{3!(p-3)!} = p(p-1)(p-2)/6$
- ...

In total there are: $\sum_{k=1}^{p} \binom{p}{k} = 2^p$ possible models. How would we proceed to find the "best" possible model?

Best Subset selection

[Best subset selection]

1. Let $\mathcal{M}_0$ denote the null model containing no predictors except for a constant (i.e. predicting the mean).
2. For $k = 1, ..., p$

   a) Fit all $\binom{p}{k}$ models that contain k predictors.

   b) Pick the best among these k dimensional models, calling it $\mathcal{M}_k$. The *best* model is the one that has smallest RSS or largest R^2.
3. Select the single best model from the set $\mathcal{M}_0, ..., \mathcal{M}_p$. Here *best* is determined using best performance in terms of MSE on a training set, or some measure of goodness of fit that adjusts for the fact that R^2 monotonically decreases as k gets larger.

Some comments on the different steps...

2. b) You know that R^2 monotonically increases as you add more control variables. Since you hold k fixed, i.e. you compare only the performance of models with the same set of predictors, you can choose the one with the largest value of R^2, since you compare "like with like".

Some comments on the different steps...

2. b) You know that R^2 monotonically increases as you add more control variables. Since you hold k fixed, i.e. you compare only the performance of models with the same set of predictors, you can choose the one with the largest value of R^2, since you compare "like with like".
3. Here, we are not comparing "like with like", but rather we want to penalize models that mechanically bound to perform better in the training sample; we can do this by computing their performance in terms of MSE on a training set, or, in absence of a training set, we can look at various statistics that adjust the measure of goodness of fit.

   We will consider AIC or adjusted R^2 as alternative estimates of *test error*.

Best Subset Selection: Optimization Problem

- We can express the best subset selection problem as a nonconvex and combinatorial optimization problem.
- The objective is to find the optimal s

$$min_{\beta} \underbrace{\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2}_{\text{Residual Sum of Squares}} \text{ subject to } \sum_{j=1}^{p} \mathbf{I}(\beta_j \neq 0) \leq s \tag{4}$$

- This requires that the optimial solution involves finding a vector β such that RSS is minimal and no more than s coefficients are non-zero.
- The algorithm presented above solves this optimization problem for every value of s and then picks among the optimal models for the different values of s.

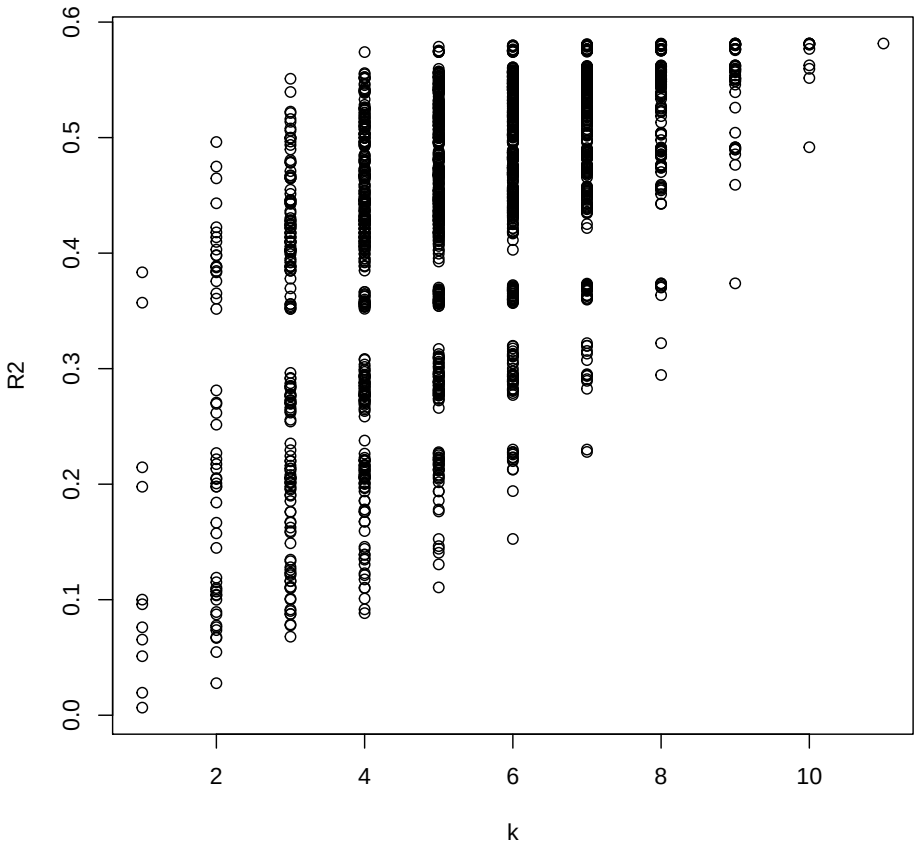
Lets do "best subset selection" in our hedonic pricing example...

- We have a range of variables to consider, they are:
- sqft, bathrooms, bedroom, built year, County (Fixed Effects), Year Fixed Effect, latitude (y), longitude (x), distance to nearest Natural gas well in 2009, 2010 and 2011
- I worked on this to study whether natural gas extraction in the neighbourhood adversely affects house prices.
- So in total, we have 11 variables to consider - in total there are $2^{11} = 2{,}048$ models to estimate.

```
#this code snippet does steps 2a, b
 vars<-c("sqft","baths","bedrooms","builtyryr","distgas2009","distgas2010","distgas2011","lat","lon","factor(
curlyM<-list()
R2.df<-NULL
for(k in 1:length(vars)) {
#create all possible ways of picking k variables from list vars
varcombs<-combn(vars,k)
#this runs all the regressions for some fixed k and returns the r.squared
R2<-apply(varcombs, 2, function(x) summary(lm(paste("soldprice ~ ",paste(x,collapse="+")),
                                              data=HOUSES[SAMPLE]))$r.squared)
R2.df<-rbind(R2.df, cbind("k"=k,R2))
#this keeps the specification for every k that has highlest R2
curlyM[[k]]<-varcombs[,which(R2==max(R2))]
}
```

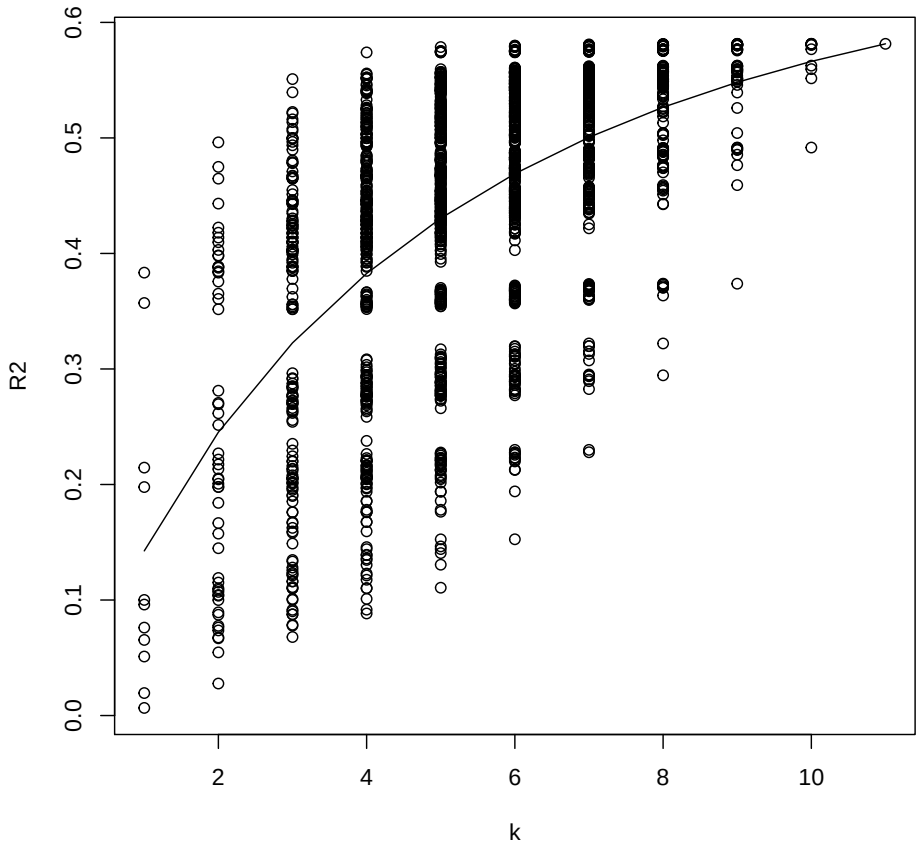
Step 2a,b: How does R^2 evolve for different values of k?

- Plotting the values for R^2 for the 2,048 models we have fit. At k=1, there are exactly 11 points, at k=11, there is exactly 1 point.



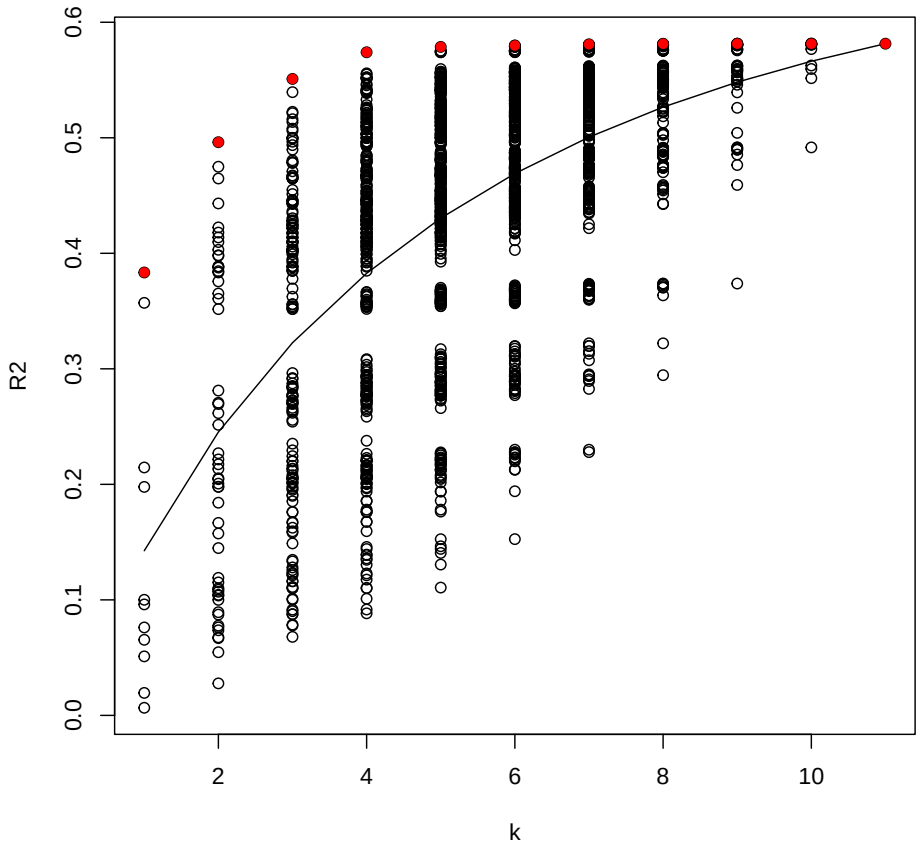
Step 2a,b: How does R^2 evolve for different values of k?

- R^2 decreases monotonically thoughout.



Step 2a,b: How does R^2 evolve for different values of k?

- "red" points indicate those models with maximal R^2 chosen in each step 2b)



Lets explore $\mathcal{M}_k$.

```
curlyM

## [[1]]
## [1] "baths"
##
## [[2]]
## [1] "baths"     "builtyryr"
##
## [[3]]
## [1] "baths"        "builtyryr"    "factor(NAME)"
##
## [[4]]
## [1] "sqft"         "baths"        "builtyryr"    "factor(NAME)"
##
## [[5]]
## [1] "sqft"         "baths"        "builtyryr"    "factor(NAME)" "factor(year)"
##
## [[6]]
## [1] "sqft"         "baths"        "builtyryr"    "distgas2009"  "factor(NAME)"
## [6] "factor(year)"
##
## [[7]]
## [1] "sqft"         "baths"        "builtyryr"    "distgas2009"  "lon"
## [6] "factor(NAME)" "factor(year)"
##
## [[8]]
## [1] "sqft"         "baths"        "builtyryr"    "distgas2009"  "lat"
## [6] "lon"          "factor(NAME)" "factor(year)"
##
## [[9]]
## [1] "sqft"         "baths"        "builtyryr"    "distgas2009"  "distgas2010"
## [6] "lat"          "lon"          "factor(NAME)" "factor(year)"
##
## [[10]]
##  [1] "sqft"         "baths"        "builtyryr"    "distgas2009"  "distgas2010"
##  [6] "distgas2011"  "lat"          "lon"          "factor(NAME)" "factor(year)"
##
## [[11]]
##  [1] "sqft"         "baths"        "bedrooms"     "builtyryr"    "distgas2009"
##  [6] "distgas2010"  "distgas2011"  "lat"          "lon"          "factor(NAME)"
## [11] "factor(year)"
```

Lets pick the "best" $\mathcal{M}_k$.

As discussed, R^2 is adequate to choose the best model, when we compare "like with like", i.e. models of the same number of parameters; however, we can not compare them across different values of k. We want to select the model with the lowest test error, i.e. the model that provides most robust predictions and does not suffer from "overfitting". There are two approaches...

1. We can add a penality term to a measure of goodness of fit for the training set, to account for the bias that is likely to arise due to overfitting. The statistics that do this are AIC and adjusted R^2.
2. Directly estimate the test error, by computing *test MSE* for the different models based on a validation set.

Approach (1) is usually followed, in case no test data set is available.

AIC and Adjusted R^2

- From first set of lectures, we know that the training set MSE is an underestimate of the of the test MSE [Remember $MSE = RSS/n$]
- That is, the MSE estimated from the training set is *too low* relative to the true test MSE.
- We can correct for this "optimism" directly.
- The first one we discuss is the *Akaike information criterion* for a model with k predictors.

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2k\hat{\sigma}^2)$$

- Its beyond this course to theoretically derive where this is coming from... but, it is derived in the context of maximum likelihood estimation.
- $\hat{\sigma}^2$ is an estimate of the variance of the residuals ϵ.
- As we increase k, the second term becomes larger. The model fit is better, **the smaller** AIC.

AIC and Adjusted R^2

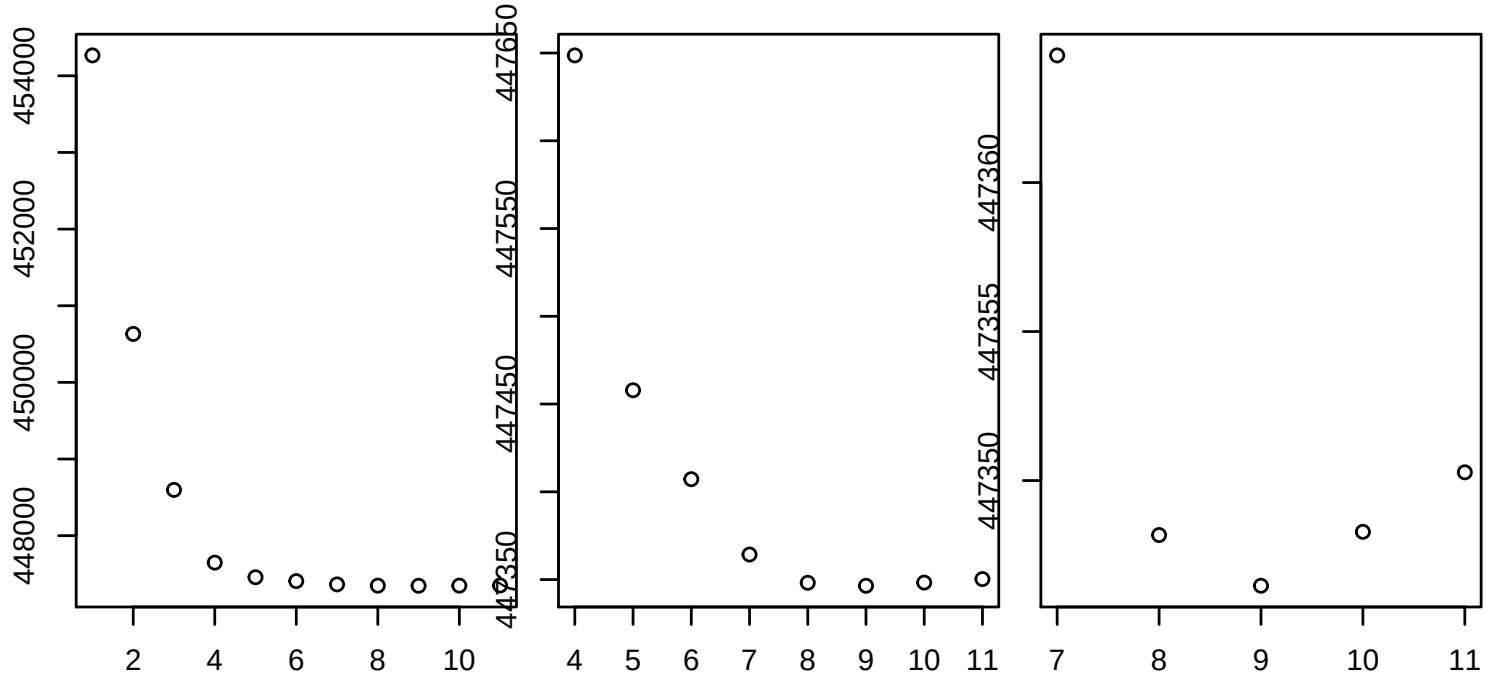
$$Adjusted R^2 = 1 - \frac{RSS/(n-k-1)}{TSS/(n-1)}$$

- Note that the denominator is constant. Maximizing Adjusted R^2 is equivalent to minimizing $\frac{RSS}{(n-k-1)}$
- However, the numerator changes non-linearily in k.
- As $k \uparrow$, $RSS \downarrow$ and $(n-k-1) \downarrow$.
- So R^2 keeps increasing, if the decrease in RSS is larger than the decrease in $(n-k-1)$.

Intuition: Once all the "correct" variables have been included in the model, additional *noise* variables will lead to only a small decrease in RSS, while the denominator $(n-k-1)$ decreases a lot, so the ratio increases and adjusted R^2 goes down.

Plotting AIC for our models in $\mathcal{M}_k$

Zooming in on AIC in the panels...



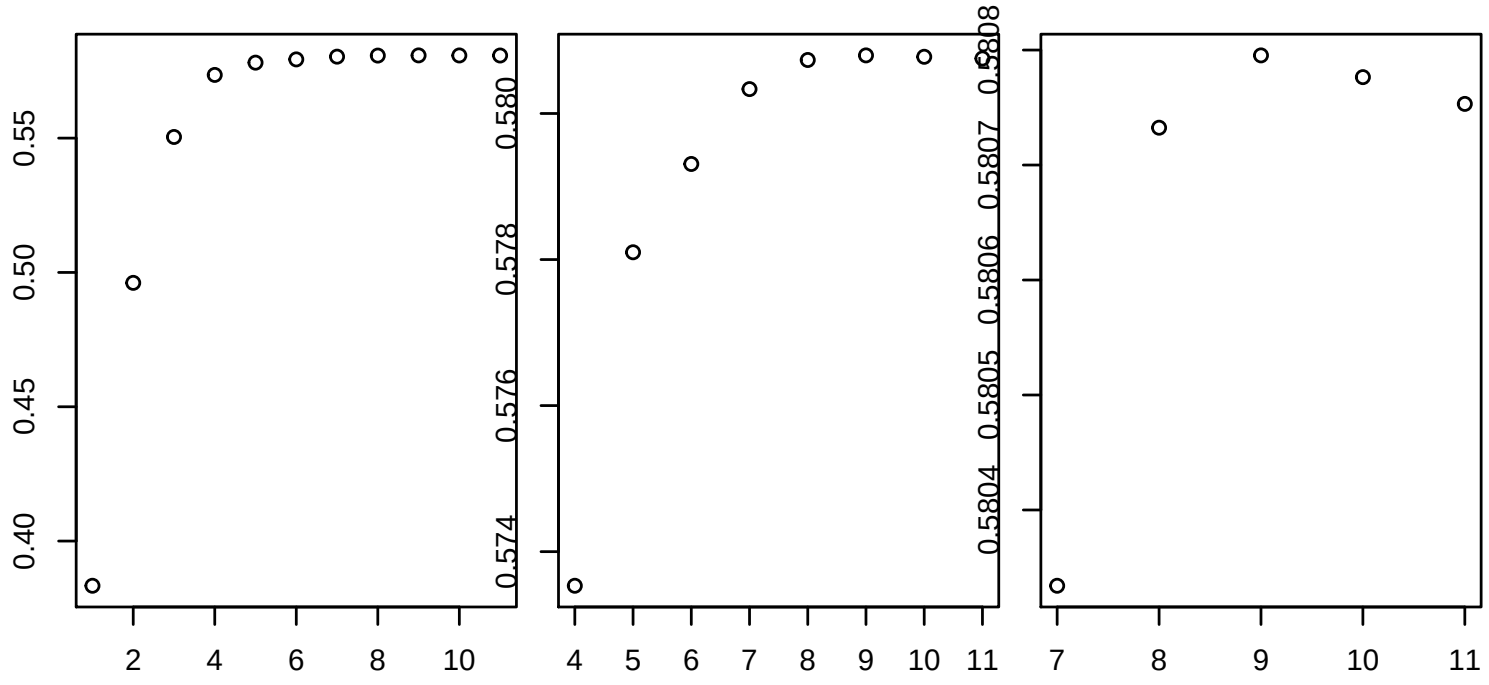
it looks like the best model is the one with $k = 9$, which includes

```
curlyM[[9]]

## [1] "sqft"          "baths"         "builtyryr"     "distgas2009"  "distgas2010"
## [6] "lat"           "lon"           "factor(NAME)" "factor(year)"
```

Plotting adjusted R^2 for our models in $\mathcal{M}_k$

Zooming in on adjusted R^2 in the panels...



it looks like the best model is the one with $k = 9$, which includes

```
curlyM[[9]]

## [1] "sqft"          "baths"         "builtyryr"     "distgas2009"  "distgas2010"
## [6] "lat"           "lon"           "factor(NAME)" "factor(year)"
```

Lets pick the "best" $\mathcal{M}_k$.

As discussed, R^2 is adequate to choose the best model, when we compare "like with like", i.e. models of the same number of parameters; however, we can not compare them across different values of k. We want to select the model with the lowest test error, i.e. the model that provides most robust predictions and does not suffer from "overfitting". There are two approaches...

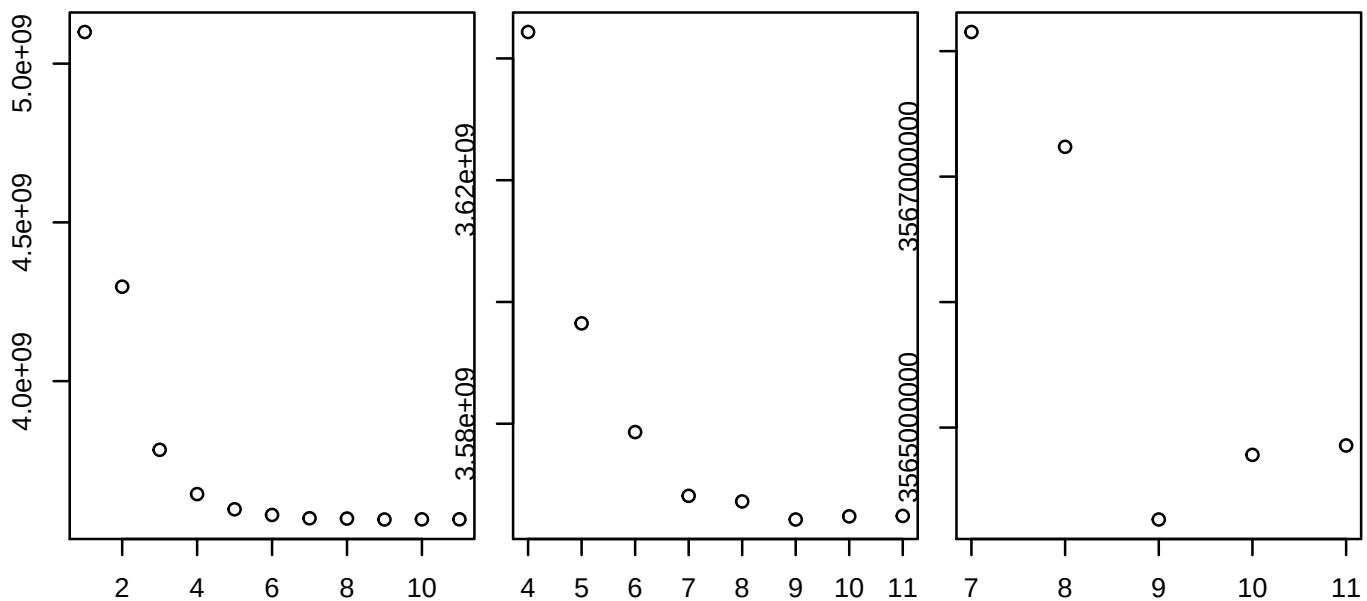
1. We can add a penality term to a measure of goodness of fit for the training set, to account for the bias that is likely to arise due to overfitting. The statistics that do this are AIC and adjusted R^2.
2. Directly estimate the test error, by computing MSE for different models based on the training set.

Approach (1) is usually followed, in case no test data set is available.

Directly estimate model performance by computing MSE for models in $\mathcal{M}_k$.

In case you have a test sample available, then the best thing to do is to compute the MSE. Practically, this is implemented by taking each "best" fitted model with k parameters, and predicting the sales price given the observable values for the X's and then computing the test MSE.

$$\hat{MSE} = 1/n\hat{RSS}$$



So what is the "best" $\mathcal{M}_k$.

So in our application, we would pick a model with a subset of predictors. The additional predictors that we have available, do not add much signal, but rather add noise affecting our prediction accuracy.

```
##
## Call:
## lm(formula = paste("soldprice ~ ", paste(x, collapse = "+")),
##     data = HOUSES[SAMPLE])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -405248  -33602   -2664   29122  392004
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -2.93e+06   3.71e+05   -7.91  2.8e-15 ***
## sqft                       2.64e+01   8.49e-01   31.12  < 2e-16 ***
## baths                      3.05e+04   8.14e+02   37.42  < 2e-16 ***
## builtyryr                  9.85e+02   1.57e+01   62.74  < 2e-16 ***
## distgas2009                1.38e+00   2.65e-01    5.20  2.0e-07 ***
## distgas2010               -5.66e-01   2.94e-01   -1.92   0.0546 .
## lat                       -2.40e+04   5.96e+03   -4.02  5.8e-05 ***
## lon                       -2.47e+04   3.67e+03   -6.73  1.7e-11 ***
## factor(NAME)Allegheny      1.61e+03   6.48e+03    0.25   0.8040
## factor(NAME)Armstrong     -4.16e+04   9.67e+03   -4.30  1.7e-05 ***
## factor(NAME)Beaver        -7.01e+04   9.19e+03   -7.63  2.4e-14 ***
## factor(NAME)Butler        -5.59e+04   1.18e+04   -4.75  2.0e-06 ***
## factor(NAME)Cambria        1.30e+04   5.36e+03    2.42   0.0155 *
## factor(NAME)Columbiana    -7.62e+04   1.01e+04   -7.51  6.3e-14 ***
## factor(NAME)Fayette       -1.53e+04   5.02e+03   -3.04   0.0024 **
## factor(NAME)Garrett        1.47e+03   4.20e+03    0.35   0.7260
## factor(NAME)Greene        -2.80e+04   1.26e+04   -2.22   0.0264 *
## factor(NAME)Indiana       -3.77e+04   1.54e+04   -2.45   0.0143 *
## factor(NAME)Lawrence      -7.13e+04   1.20e+04   -5.93  3.2e-09 ***
## factor(NAME)Mahoning      -9.75e+04   1.21e+04   -8.04  9.6e-16 ***
## factor(NAME)Mercer        -7.41e+04   1.43e+04   -5.18  2.2e-07 ***
## factor(NAME)Monongalia     3.73e+03   5.05e+03    0.74   0.4605
## factor(NAME)Somerset       1.53e+04   4.87e+03    3.15   0.0016 **
## factor(NAME)Trumbull      -1.05e+05   1.36e+04   -7.66  2.0e-14 ***
## factor(NAME)Washington     4.41e+03   6.60e+03    0.67   0.5041
## factor(NAME)Westmoreland   1.01e+04   5.58e+03    1.80   0.0714 .
## factor(year)2011           6.96e+03   2.46e+03    2.83   0.0047 **
## factor(year)2012           2.01e+04   2.45e+03    8.22  < 2e-16 ***
## factor(year)2013           1.85e+04   2.53e+03    7.30  3.0e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60300 on 17971 degrees of freedom
## Multiple R-squared:  0.581, Adjusted R-squared:  0.581
## F-statistic:  892 on 28 and 17971 DF,  p-value: <2e-16
```

When Best Subset selection becomes infeasible,...

- There are p models, containing exactly 1 predictor
- There are $\binom{p}{2} = p(p-1)/2$ possible ways to choose 2 predictors [Remember: $\binom{p}{k} = \frac{p!}{k!(p-k)!}$]
- There are $\binom{p}{3} = \frac{p!}{3!(p-3)!} = p(p-1)(p-2)/6$
- ...

In total there are: $\sum_{k=1}^{p} \binom{p}{k} = 2^p$ possible models.

With $p = 40$, this is a total of 1,099,511,627,776 different combinations of dependent variables... you can wait a long time. This is sometimes referred to as the **curse of dimensionality**.

Alternative Selection methods

There are "stepwise selection" algorithms that provide an alternative. They are effectively *a constrained search* for the "best" model.

Just present two algorithms.

1. Forward stepwise selection: Make model iteratively more complex.
2. Backward stepwise selection: Make model iteratively less complex.

Instead of considering all 2^p possible models, these algorithms adds (remove) one predictor at a time until all predictors are included (excluded). This greatly reduces the search space.

Forward Stepwise Selection

[Forward stepwise selection]

1. Let $\mathcal{M}_0$ denote the null model containing no predictors except for a constant (i.e. predicting the mean).
2. For $k = 0, ..., p-1$

   a) Consider all $p-k$ models that augment the predictors of $\mathcal{M}_k$ with one additional predictor.

   b) Choose the best among these $p-k$ models and call it M_{k+1}. The *best* model is the one that has smallest RSS or largest R^2.
3. Select the single best model from the set $\mathcal{M}_0, ..., \mathcal{M}_p$. Here *best* is determined using best performance in terms of MSE on a training set, or some measure of goodness of fit that adjusts for the fact that R^2 monotonically decreases as k gets larger.

Forward Stepwise Selection

Where or how do we save time here?

- Each fowrard step involves fitting only $p - k$ models in each iteration k, rathar than $\binom{p}{k}$.
- So in first iteration, we fit p models, in second, we fit $p - 1$, in third ...
- This means, in total we fit $\sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models.
- in case $p = 40$, this amounts to 821 models.

Backward Stepwise Selection

[Backward stepwise selection]

1. Let $\mathcal{M}_p$ denote the *full* model containing all p predictors.
2. For $k = p, p-1, ..., 1$

   a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, i.e. that contain a total of $k-1$ predictors.

   b) Choose the best among these k models and call it M_{k-1}. The *best* model is the one that has smallest RSS or largest R^2.
3. Select the single best model from the set $\mathcal{M}_0, ..., \mathcal{M}_p$. Here *best* is determined using best performance in terms of MSE on a training set, or some measure of goodness of fit that adjusts for the fact that R^2 monotonically increases in more complex models.

Plan

From Cloud AI to Self-Sovereign AI

- AI today is predominantly centralized: hosted on cloud platforms with opaque data pipelines.
- This raises key concerns:
  - **Privacy:** Who owns the prompts, outputs, and embeddings?
  - **Autonomy:** Dependency on external APIs and licensing terms.
  - **Sustainability:** Network latency and carbon footprint of cloud inference.
- Self-sovereign AI = fully local AI, under user control.

What is Self-Sovereign AI?

Definition

Self-sovereign AI refers to running and managing AI models locally, with full control over data, inference, and updates.

What is Self-Sovereign AI?

Definition

Self-sovereign AI refers to running and managing AI models locally, with full control over data, inference, and updates.

Core Principles:

- **Local inference:** No cloud dependency.
- **Model portability:** Bring your own model (BYOM).
- **Interoperability:** Use with any local app or dataset.
- **Transparency:** Inspect and customize all steps.

Enter Llamafile

- **Llamafile** is a single-file executable bundling:
  - A GGUF model (e.g., LLaMA, Mistral)
  - The llama.cpp runtime
  - A lightweight local web server
- Created by Mozilla engineer
- Works on Linux, macOS, and Windows – no install needed

One File, All You Need

```
./llamafile --model model.gguf
```

Why Llamafile?

Advantages:

- Zero setup, portable
- Fast, quantized inference
- Works offline
- Local API endpoint (OpenAI-compatible)
- Embedding + chat in one file

Example Use Cases

- **Local document Q&A** (e.g., PDF or email archives)
- **Knowledge agents with embeddings**
- **Coding assistants** that respect your local codebase
- **Offline research copilot** for field work or secure contexts

Combine Llamafile with tools like:

- `phi`, `langchain`, `llamaindex`
- Embedding tools: `text-embeddings-ada`, `nomic-embed`

Demo: Local AI with Llamafile

Goal: Load a LLaMA 3 8B model via Llamafile, chat locally via browser.

1. Download the model: `llama3-8b.Q4_K_M.gguf`
2. Get the Llamafile binary: `wget llamafile.com/llama3.llamafile`
3. Run it: `chmod +x llama3.llamafile && ./llama3.llamafile`
4. Visit `localhost:8080`

Demo: Local AI with Llamafile

Goal: Load a LLaMA 3 8B model via Llamafile, chat locally via browser.

1. Download the model: `llama3-8b.Q4_K_M.gguf`
2. Get the Llamafile binary: `wget llamafile.com/llama3.llamafile`
3. Run it: `chmod +x llama3.llamafile && ./llama3.llamafile`
4. Visit `localhost:8080`

Optional: Use with OpenAI-compatible clients or plug it into your local RAG setup.

The Future of Personal AI

- Local LLMs will be as ubiquitous as text editors
- Model distillation and quantization make this possible on laptops and phones
- Self-sovereign AI empowers researchers, educators, developers

The Future of Personal AI

- Local LLMs will be as ubiquitous as text editors
- Model distillation and quantization make this possible on laptops and phones
- Self-sovereign AI empowers researchers, educators, developers

Call to Action

Experiment with Llamafile. Share use cases. Build open, local, interpretable AI.

Resources

- llamafile.com
- github.com/Mozilla-Ocho/llamafile
- TheBloke GGUF models
- @justine-lindsey (GitHub)
- llama.cpp documentation