The Predictive Paradigm

Machine Learning

Ignacio Sarmiento-Barbieri

Universidad de La Plata

¿Qué entendemos por Big Data y ML?

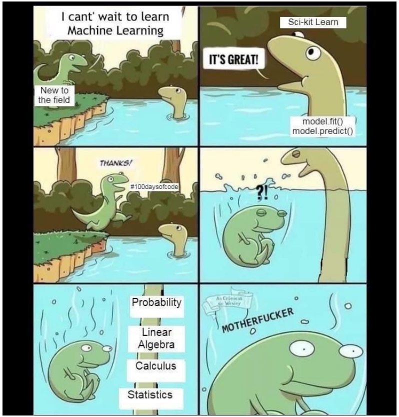
- ¿Que es Big Data?
  - Big n, es solo parte de la historia
  - Big también es big k, muchos covariates, a veces $n << k$
  - Vamos a entender Big también como datos que no surgen de fuentes tradicionales (cuentas nac., etc)
    - Datos de la Web, Geográficos, etc.
- Machine Learning
  - Cambio de paradigma de estimación a predicción

Agenda

Agenda

Lenguajes

- Estadística y Econometría
- Inglés
- Código
  - Elijan el que quieran:
    - `Python`, `R`, o cualquier otro
    - no hay restricción
    - yo me basare en `Python`
  - Github
  - Slack
- Aprender haciendo y mucha prueba y error!



Materiales

1. Página web: link
2. Statistical Learning (FREE!!! (as beer, not speech))
   `https://www.gnu.org/philosophy/free-sw.en.html`
   - James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (ISLP)
   - Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction
   - Békés, G., & Kézdi, G. (2021). Data analysis for business, economics, and policy. Cambridge University Press.

Agenda

Machine learning is all about prediction

- Machine learning is a branch of computer science and statistics, tasked with developing algorithms to predict outcomes y from observable variables x.
- The learning part comes from the fact that we don't specify how exactly the computer should predict y from x.
- This is left as an empirical problem that the computer can "learn".
- In general, this means that we abstract from the underlying model, the approach is pragmatic

Machine learning is all about prediction

- Machine learning is a branch of computer science and statistics, tasked with developing algorithms to predict outcomes y from observable variables x.
- The learning part comes from the fact that we don't specify how exactly the computer should predict y from x.
- This is left as an empirical problem that the computer can "learn".
- In general, this means that we abstract from the underlying model, the approach is pragmatic

"Whatever works, works...."

"Whatever works, works...."

“Whatever works, works....”????

- In many applications, ML techniques can be successfully applied by data scientists with little knowledge of the problem domain.
- For example, the company Kaggle hosts prediction competitions (`www.kaggle.com/competitions`) in which a sponsor provides a data set, and contestants around the world can submit entries, often predicting successfully despite limited context about the problem.

“Whatever works, works....”????

- However, much less attention has been paid to the limitations of pure prediction methods.
- When ML applications are used “off the shelf” without understanding the underlying assumptions then the validity and usefulness of the conclusions can be compromised.
- A deeper question concerns whether a given problem can be solved using only techniques for prediction, or
- whether statistical approaches to estimating the causal effect of an intervention are required.

Agenda

Policy Prediction Problems

- Empirical policy research often focuses on causal inference.
- Since policy choices seem to depend on understanding the counterfactual— what happens with and without a policy—this tight link of causality and policy seems natural.
- While this link holds in many cases, there are also many policy applications where causal inference is not central, or even necessary.

Prediction vs. Causality

Prepare

- A loan officer wants to know the likelihood of an individual repaying a loan based on income, employment, and other characteristics.

Influence

- A mortgage lender wants to know if direct debit will increase loan repayments.

Prediction vs. Causality

Prepare

- A home seller wants to know what price homes with the characteristics of his or her home typically sell for.

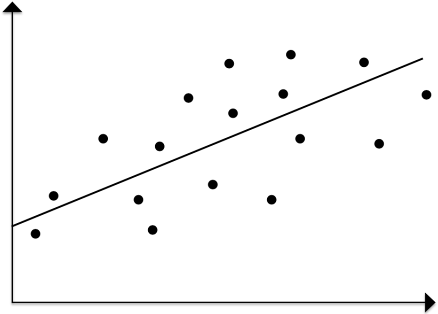
Influence

- A home seller wants to know by how much installing new windows will raise the value of his or her home.

Prediction vs. Causality: Target

$$y = f(x) + \epsilon \tag{1}$$

$$y = \alpha + \beta x + \epsilon \tag{2}$$



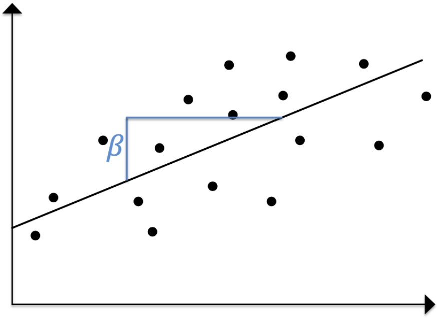
The Causal Paradigm

$$y = f(X) + u \tag{3}$$

- Interest lies on inference
- "Correct" $f()$ to understand how y is affected by X
- Model: Theory, experiment
- Hypothesis testing (std. err., tests)

Prediction vs. Causality: Target

$$y = \alpha + \beta x + \epsilon \tag{4}$$



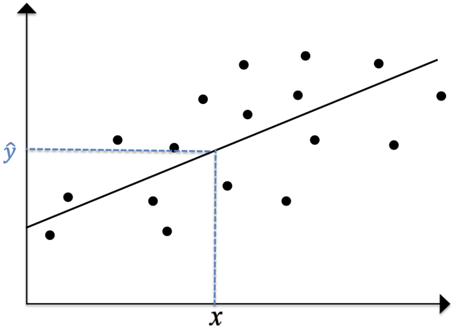
The Predictive Paradigm

$$y = f(X) + u \tag{5}$$

- Interest on predicting y
- "Correct" $f()$ to be able to predict (no inference!)
- Model? We treat $f()$ as a black box, and
- any approximation $\hat{f}()$ that yields a good prediction is good enough (*Whatever works, works.*).

Prediction vs. Causality: Target

$$y = \underbrace{\alpha + \beta x}_{\hat{y}} + \epsilon \tag{6}$$



Prediction vs. Causality: The garden of the parallel paths?

- We've seen that prediction and causality
  - Answer different questions
  - Serve different purposes
  - Seek different targets
- Different strokes for different folks, or complementary tools in an applied economist's toolkit?

Agenda

Agenda

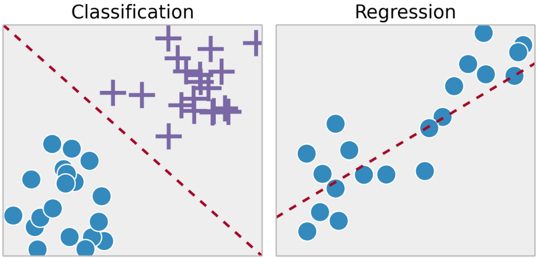
ML branches

- ML tasks can (¿?) be divided into two main branches:

  1. Supervised Learning

ML branches

- Supervised Learning
  - for each predictor x_i a 'response' is observed y_i.
  - everything we have done in econometrics is supervised



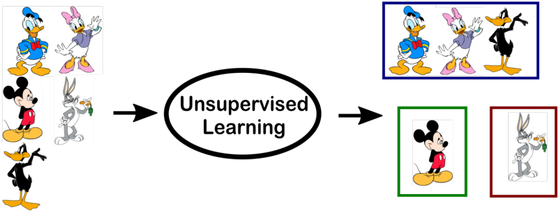
Source: shorturl.at/opqKT

ML branches

- ML tasks can (¿?) be divided into two main branches:
  1. Supervised Learning
  2. Unsupervised Learning

ML branches

- Unsupervised Learning
  - observed x_i but no response.
  - example: cluster analysis



Source: shorturl.at/opqKT

Agenda

1. About the Course
2. Machine learning is all about prediction
3. Prediction vs Causality
4. ML Tasks
5. **Regression, Prediction and loss functions**
6. Bias/Variance Decomposition
7. Prediction and linear regression
8. In-Sample and Out-of-Sample Prediction.
   - AIC: Akaike Information Criterion
   - SIC/BIC: Schwarz/Bayesian Information Criterion
   - Cross-Validation
9. Review

Getting serious about prediction: Regression

$$y = f(X) + u \tag{7}$$

- Interest on predicting y
- Model? We treat $f()$ as a black box, and any approximation $\hat{f}()$ that yields a good prediction is good enough ("*Whatever works, works...*").
- How do we measure "what works"?

Getting serious about prediction: Regression

$$y = f(X) + u \tag{7}$$

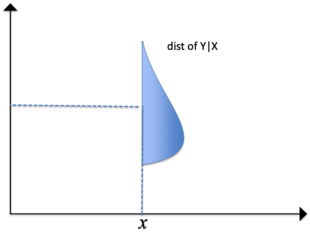
- Interest on predicting y
- Model? We treat $f()$ as a black box, and any approximation $\hat{f}()$ that yields a good prediction is good enough (*"Whatever works, works..."*).
- How do we measure "what works"?
- Formal statistics can help figure out this: what is a good prediction.

Minimizing our losses

- Want our prediction to be "close" i.e. minimize the expected loss function
- Formally, a supervised learning algorithm takes as an input a loss function $L(\hat{y}, y)$ and searches for a function $\hat{f}$ within a function class $\mathcal{F}$ that has a low expected prediction loss

$$E_{(y,X)}[L(\hat{f}(X), y)] \tag{8}$$

on a new data point from the same distribution.



Minimizing our losses

- A very common loss function in a regression setting is the squared loss $L(d) = d^2$
- Under this loss function the expected prediction loss is the mean squared error (MSE)

- Can we find the function f^* within a function class $\mathcal{F}$ that has a low expected prediction loss?

Minimizing our losses

- By conditioning on X, it suffices to minimize the $MSE(f)$ point wise so

$$f(x) = argmin_{f^*} E_{y|X}[(y - f^*)^2 | X = x) \tag{9}$$

- f^* a random variable and we can treat f^* as a constant (predictor)

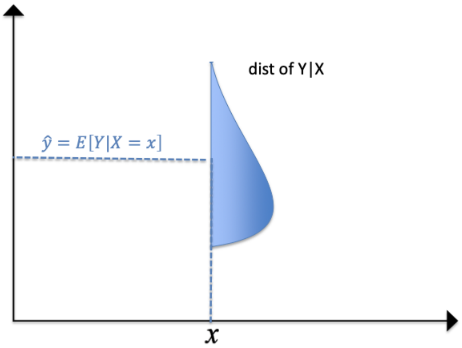
$$min_{f^*} E(y - f^*)^2 = \int (y - f^*)^2 f(y) dy \tag{10}$$

- **Result**: The best prediction of y at any point $X = x$ is the conditional mean, when best is measured using a square error loss

Minimizing our losses

- **Result**: The best prediction of y at any point $X = x$ is the conditional mean, when best is measured using a square error loss

$$f^* = E[y|X = x] \tag{11}$$



Minimizing our losses

- Prediction problem solved if we knew $f^* = E[y|X = x]$
- But we have to settle for an estimate: $\hat{f}(x)$
- The MSE of this

$$E(y - \hat{y})^2 = E(f(X) + u - \hat{f}(X))^2 \tag{12}$$

Reducible and irreducible error

$$E(y-\hat{y})^2 = \underbrace{E[f(X)-\hat{f}(X)]^2}_{Reducible} + \underbrace{Var(u)}_{Irreducible} \tag{13}$$

- The focus the is on techniques for estimating f with the aim of minimizing the reducible error
- It is important to keep in mind that the irreducible error will always provide an upper bound on the accuracy of our prediction for y
- This bound is almost always unknown in practice

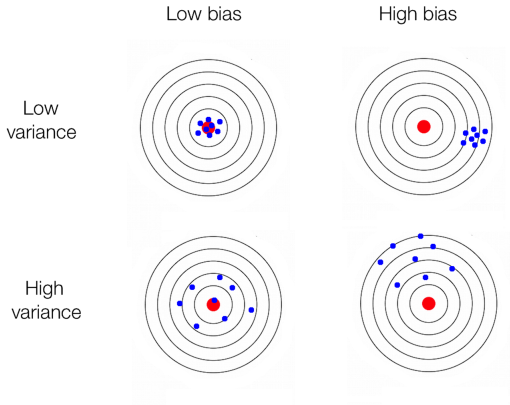
Agenda

Bias/Variance Decomposition

Recall that

- $Bias(\hat{f}(X)) = E(\hat{f}(X)) - f(X) = E(\hat{f}(X) - f(X))$
- $Var(\hat{f}(X)) = E(\hat{f}(X) - E(\hat{f}(X)))^2$

Bias/Variance Decomposition



Source: https://tinyurl.com/y4lvjxpc

Bias/Variance Decomposition

Recall that

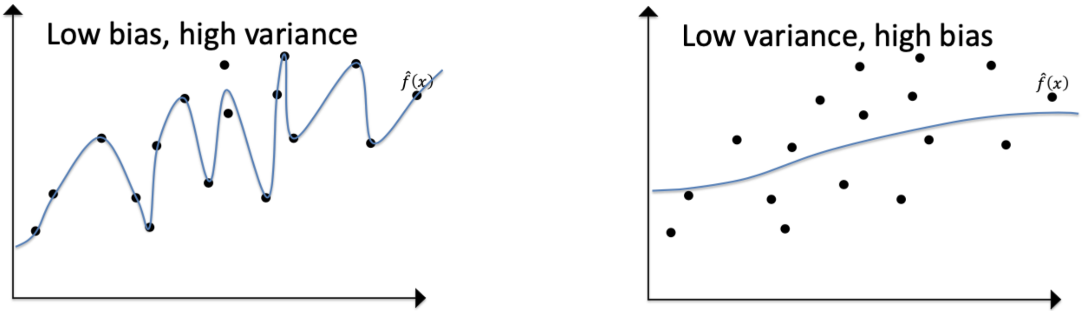
- $Bias(\hat{f}(X)) = E(\hat{f}(X)) - f(X) = E(\hat{f}(X) - f(X))$
- $Var(\hat{f}(X)) = E(\hat{f}(X) - E(\hat{f}(X)))^2$

Result (very important!)

$$MSE = Bias^2(\hat{f}(X)) + V(\hat{f}(X)) + \underbrace{Var(u)}_{Irreducible} \tag{14}$$

HW: Proof

Bias/Variance Decomposition



Agenda

Prediction and linear regression

- The goal is to predict y given another variables X.
- We assume that the link between y and X is given by the simple model:

$$y = f(X) + u \tag{15}$$

- we just learned that under a squared loss we need to approximate $E[y|X = x]$

Prediction and linear regression

- As economists we know that we can approximate $E[y|X = x]$ with a linear regression

$$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \tag{16}$$

- The problem boils down to estimating βs
- We can estimate these using
  - OLS
  - MLE
  - MM

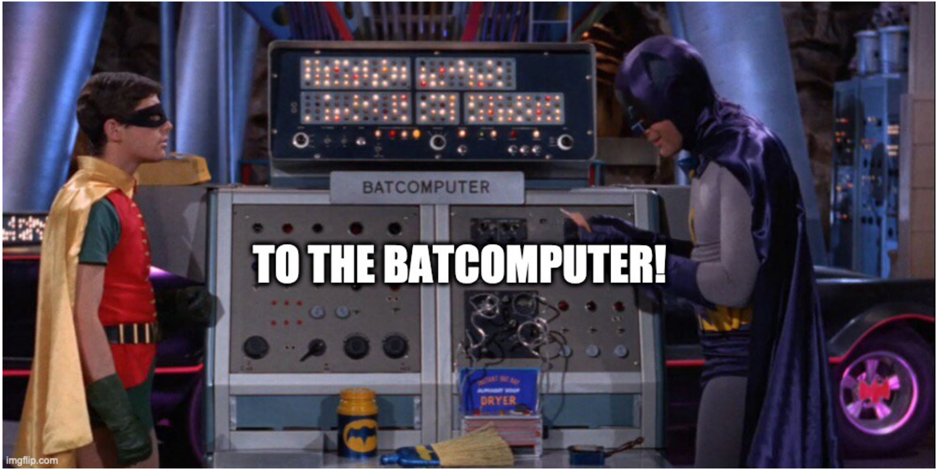


photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

Agenda

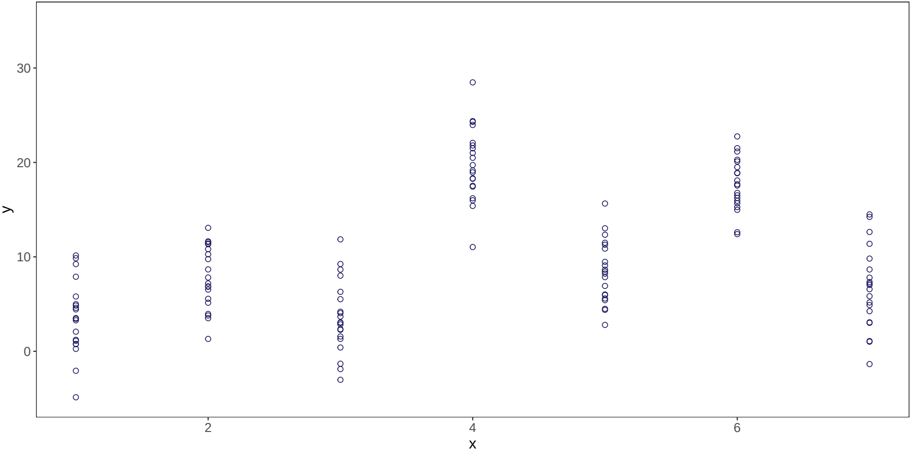
Train and Test Sets. In-Sample and Out-of-Sample Prediction.

- El objetivo es predecir y dadas otras variables X. Ej: salario dadas las características del individuo
- Asumimos que el link entre y and X esta dado por el modelo:

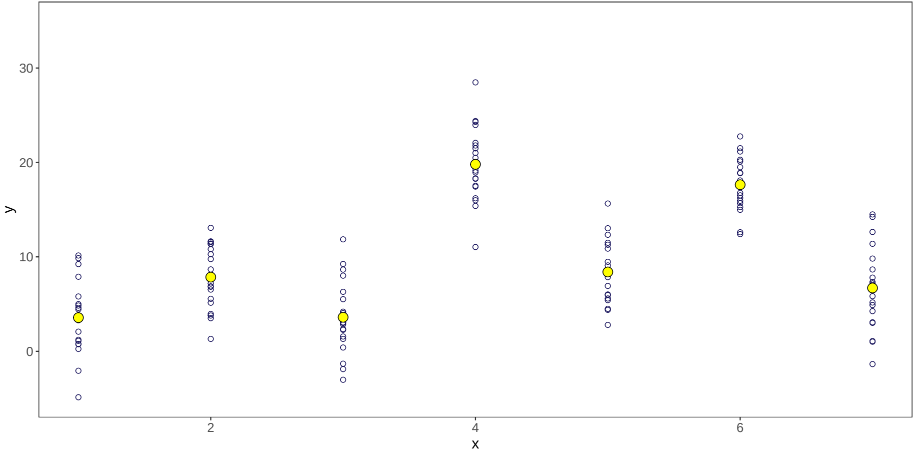
$$y = f(X) + u \tag{17}$$

- donde $f(X)$ por ejemplo es $\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$
- u una variable aleatoria no observable $E(u) = 0$ and $V(u) = \sigma^2$

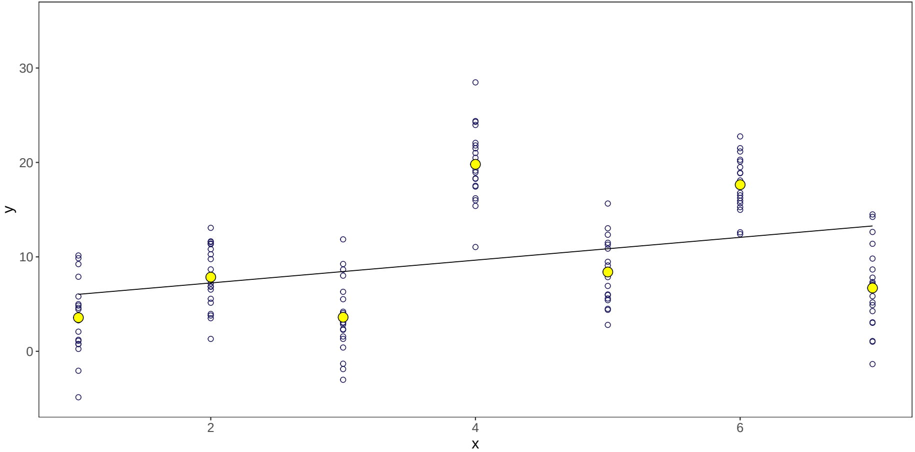
In-Sample Prediction and Overfit



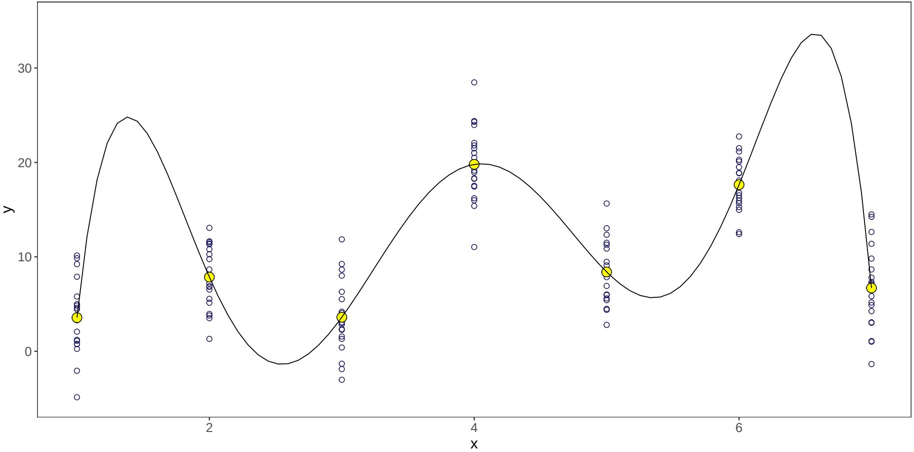
In-Sample Prediction and Overfit



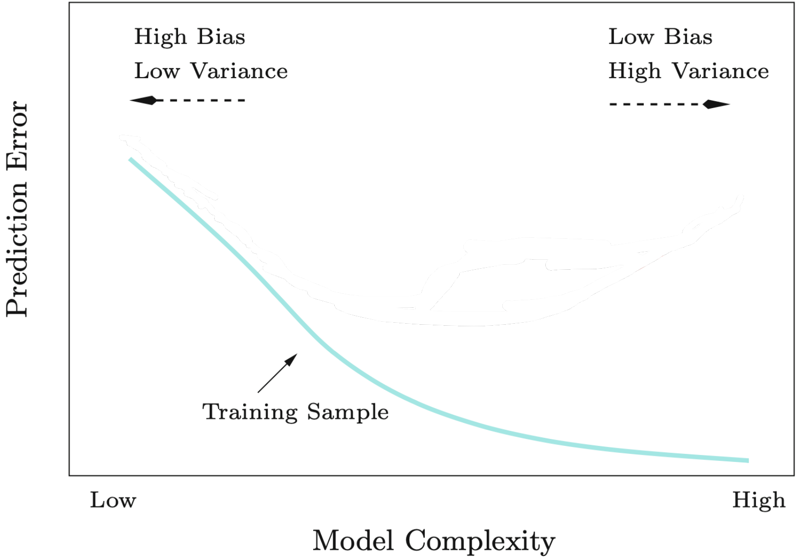
In-Sample Prediction and Overfit



In-Sample Prediction and Overfit



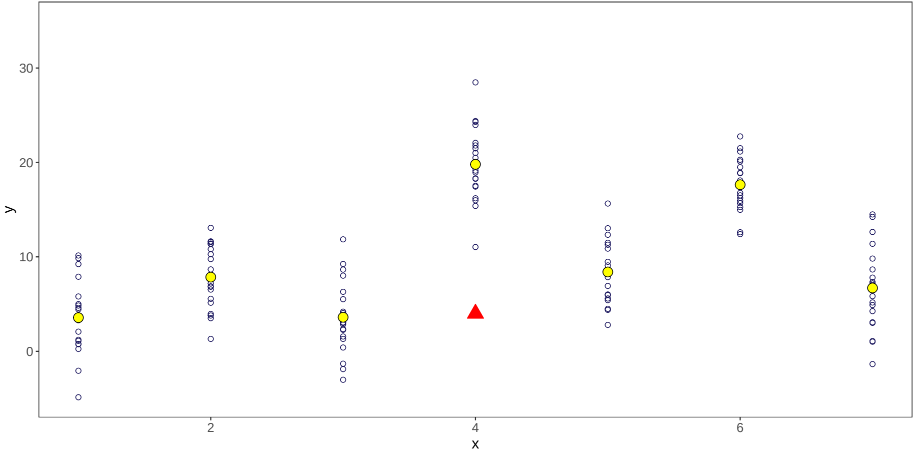
In-Sample Prediction and Overfit



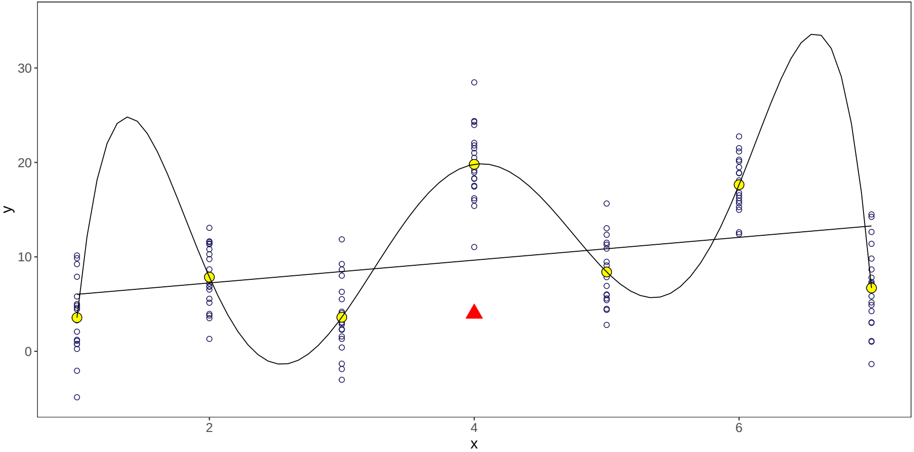
Out-of-Sample Prediction and Overfit

- ML nos interesa la predicción fuera de muestra

Out-of-Sample Prediction and Overfit



Out-of-Sample Prediction and Overfit



Out-of-Sample Prediction and Overfit

- ML nos interesa la predicción fuera de muestra
- Overfit: modelos complejos predicen muy bien dentro de muestra, pero tienden a hacer un trabajo fuera de muestra
- Hay que elegir el nivel adecuado de complejidad

Train and Test Sets. In-Sample and Out-of-Sample Prediction.

- Dos conceptos importantes
  - *Training error*: es el error de predicción en la muestra que fue utilizada para ajustar el modelo

$$Err_{\mathcal{T}rain} = MSE[(y,\hat{y})|\mathcal{T}rain] \tag{18}$$

  - *Test Error*: es el error de predicción fuera de muestra

$$Err_{\mathcal{T}est} = MSE[(y,\hat{y})|\mathcal{T}est] \tag{19}$$

Train and Test Sets. In-Sample and Out-of-Sample Prediction.

- Como estimamos el error de predicción fuera de muestra?

Train and Test Sets. In-Sample and Out-of-Sample Prediction.

- Como estimamos el error de predicción fuera de muestra?
- Problema: solo contamos con una muestra

In-Sample Prediction and Overfit

- Notemos que el MSE no es otra cosa que la suma de los residuales al cuadrado

$$
\begin{aligned}
MSE &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(X))^2 && (20)\\
&= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 && (21)\\
&= \frac{1}{n}\sum_{i=1}^{n}(e)^2 && (22)\\
&= SSR && (23)
\end{aligned}
$$

- Esta medida nos da una idea de *lack of fit* que tan mal ajusta el modelo a los datos

In-Sample Prediction and Overfit

- Un problema del SSR es que nos da una medida absoluta de ajuste de los datos, y por lo tanto no esta claro que constituye un buen SRR.
- Una alternativa muy usada en economía es el R^2
- Este es una proporción (la proporción de varianza explicada),
  - toma valores entre 0 y 1,
  - es independiente de la escala (o unidades) de y

$$
\begin{aligned}
R^2 &= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} && (24) \\
&= 1 - \frac{SRR}{TSS} && (25)
\end{aligned}
$$

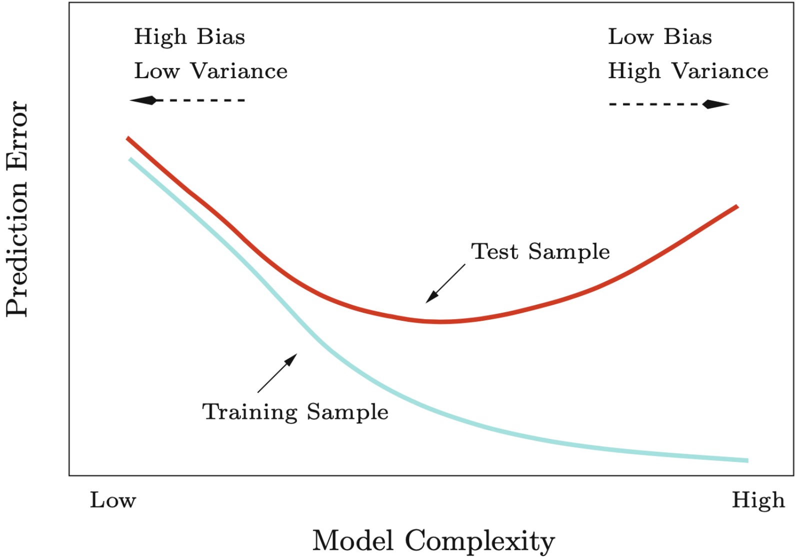
Out-of-Sample Prediction and Overfit

- El problema de usar estas medidas?

Out-of-Sample Prediction and Overfit

- El problema de usar estas medidas?
- Overfit: modelos complejos predicen muy bien dentro de muestra, pero tienden a hacer un trabajo fuera de muestra
- Estas medidas calculadas dentra de muestra tienden a ser optimistas sobre el error fuera de mustra
  - Por ej: R^2 es no decreciente en complejidad

Out-of-Sample Prediction and Overfit



Test Error

- Para seleccionar el mejor modelo con respecto al Test Error (error de prueba), es necesario estimarlo.
- Hay dos enfoques comunes:
  - Podemos estimar indirectamente el error de la prueba haciendo un ajuste al error de entrenamiento para tener en cuenta el sesgo debido al sobreajuste $\Rightarrow$ Penzalización ex post: AIC, BIC, R2 ajustado

Test Error

AIC

- Akaike (1969) fue el primero en ofrecer un enfoque unificado al problema de la selección de modelos.
- Elegir el modelo j tal que se minimice:

$$AIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - p_j \tag{26}$$

Test Error

SIC/BIC

- Schwarz (1978) mostró que el AIC es inconsistente, (cuando $n \to \infty$, tiende a elegir un modelo demasiado grande con probabilidad positiva)
- Schwarz (1978) propuso:

$$SIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - \frac{1}{2}p_j log(n) \tag{27}$$

Test Error

AIC vs BIC

$$AIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - p_j \tag{28}$$

$$SIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - p_j\frac{1}{2}log(n) \tag{29}$$

- SIC tiende a elegir modelos más pequeños.
- En efecto, al dejar que la penalización tienda al infinito lentamente con n, eliminamos la tendencia de AIC a elegir un modelo demasiado grande.

Test Error

- Para seleccionar el mejor modelo con respecto al Test Error (error de prueba), es necesario estimarlo.
- Hay dos enfoques comunes:
  - Podemos estimar indirectamente el error de la prueba haciendo un ajuste al error de entrenamiento para tener en cuenta el sesgo debido al sobreajuste $\Rightarrow$ Penzalización ex post: AIC, BIC, R2 ajustado
  - Levantarnos de nuestros bootstraps (resampling methods) y estimar directamente el Test Error (error de prueba)

Test Error

Cross-Validation

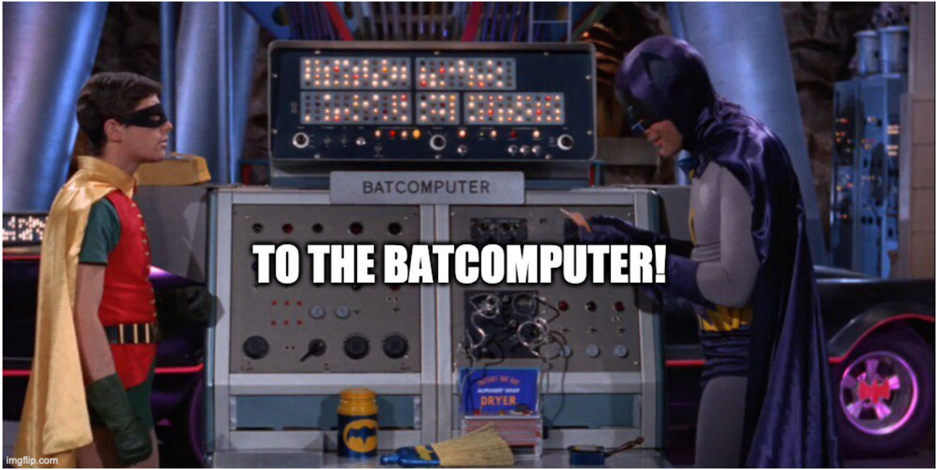


photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

Agenda

Review

- This Week:
  - Machine Learning is all about prediction
  - ML targets something different than causal inference, they can complement each other
  - Bias Variance trade-off: tolerating some bias is possible to reduce $V(\hat{f}(X))$ and lower MSE (ML best kept secret)
  - Overfit and Model Selecction
    - AIC y BIC
    - Validation Approach
    - LOOCV
    - K-fold Cross-Validation