Sesión 3: Modelos de regresión y clasificación

ML-IA: Módulo de Machine learning

Jacob Muñoz-Castro

Universidad de los Andes

10 de noviembre de 2020

Agenda

1. Mínimos cuadrados ordinarios
2. Predicción dentro y fuera de muestra
3. Regresión polinomial
4. Métricas de evaluación
5. Sobreajuste y validación cruzada

Introducción al análisis predictivo (DSD)

Predicción

- La meta es predecir Y con otra variable aleatoria $\hat{Y}$.
- La predicción de Y va a estar dada por $\hat{Y} = f(X)$.
- Definamos el **error de predicción** como:

$$Err(\hat{Y}) = E(Y - \hat{Y})$$

- Con esto en mente, asumimos un puente entre Y y X dado por un modelo:

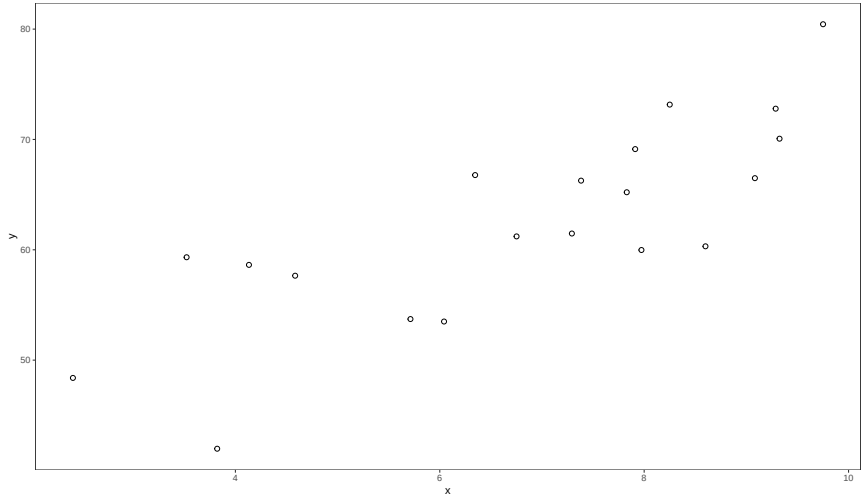
$$Y = f(X) + \varepsilon$$

- ε es una variable aleatoria no es observable con $E(\varepsilon) = 0$ y $Var(\varepsilon) = \sigma^2$.
- En la práctica no se observa f. Entonces, estimamos con $\hat{f}$.
- Por lo tanto, tenemos

$$Err(\hat{Y}) = \underbrace{Bias(\hat{f})^2 + Var(\hat{f})}_{\textbf{Reducible}} + \sigma^2$$

Modelo de regresión lineal

Caballito de batalla del aprendizaje de máquinas.



Mínimos cuadrados ordinarios

- Usemos entonces $f(X) = X\beta$ y vamos entonces a tener interés en estimar β.
- Queremos tener β tal que:

$$y = X\beta + \varepsilon$$

  tenga el mejor ajuste.
- Tenemos entonces que resolver $\text{mín}_{\beta}\ \varepsilon'\varepsilon$.
- **Intuición:** reducir la magnitud del error (el cuadrado me quita la dirección).
- La solución de la optimización es:

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

- **Intuición:** Nuestra predicción de y es $f(X) = X\beta = X(X'X)^{-1}X'y = Py$, donde P es la matriz de proyección (de X sobre y).

Predicción y Regresión Lineal

- La precisión de nuestra predicción va a estar dada por:

$$E(\hat{\beta} - \beta)^2 = E(\beta - E(\hat{\beta}))^2 + Var(\hat{\beta}) = Bias(\hat{\beta})^2 + Var(\hat{\beta})$$

- Predecir Y implica *aprender* de f. Buscamos mejorar la capacidad predicitiva.
- La elección de modelos y el aprendizaje va a estar orientado a elegir f y una estrategia para estimarla ($\hat{f}$).
- Elegir entre modelos implica un costo de oportunidad entre sesgo y varianza (bias/variance trade-off).
- En nuestro caso de regresión lineal:
  - Incluir más variables respecto al modelo verdadero reduce el sesgo pero aumenta la ineficiencia del modelo.
  - Incluir menos variables respecto al modelo verdadero aumenta el sesgo al costo de mayor eficiencia.

Predicción fuera de muestra

- El objetivo del aprendizaje de máquinas es la predicción fuera de muestra.
- En nuestro modelo de regresión lineal, MCO minimizaba la suma de residuales al cuadrado (maximiza el ajuste del modelo).
- **Pero**, para predecir lo que realmente importa es la habilidad de predecir nuevos datos.

Muestra de entrenamiento y prueba

- Para lograr esa noción de datos nuevos, un acercamiento simple es hacer una partición de los datos:
  - Muestra de entrenamiento sirve para construir, estimar y entrenar el modelo.
  - Muestra de prueba sirve para evaluar el desempeño del modelo.

Regresión polinomial

- La idea principal no va más allá de identificar relaciones no lineales entre X y Y (particularmente polinómicas). Ej: producción $\sim$ costos, salario $\sim$ edad, riesgo moral $\sim$ cobertura de un seguro

R^2 (Coeficiente de determinación)

- Vimos de antes que MCO (OLS) minimiza la suma de los residuales al cuadrado.
- El R^2 lo definimos como:

$$R^2 = \frac{SSM}{SST} = \frac{\sum(\hat{y} - \bar{y})^2}{\sum(y - \bar{y})^2} = 1 - \frac{SSR}{SST}$$

- **Intuición:** ¡Cuánta de la variación de los datos está explicada por la predicción del modelo!
- Esta métrica solo mide el desempeño del modelo dentro de muestra.
- Pero nuestro interés es la capacidad predictiva del modelo!
- Entonces: *¿Cómo medimos el desempeño de nuestros modelos fuera de muestra?*

MSE

- Habíamos definido el error de predicción como el valor esperado de la diferencia al cuadrado del valor predicho y el valor observado.
- Esta medida es conceptualmente el **error cuadrático medio** (MSE).
- La diferencia radica en que el MSE mide el ajuste de un estimador mientras que el error de predicción mide el ajuste de la predicción.
- Entonces, cuando usemos MSE para ML, realmente estamos usando el error cuadrático medio de la predicción (MSPE).

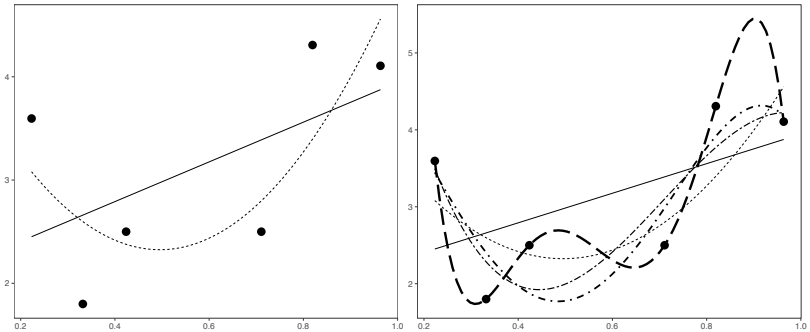
MAPE

- Un inconveniente que tiene medir ajuste de predicción con MSE es que este depende de la escala en la que está la variable.
- Si se busca comparar el ajuste de predicciones para variables con diferentes escalas, lo más adecuado es usar MAPE:

$$MAPE = E\left|\frac{Y - \hat{Y}}{Y}\right|$$

- Con esto removemos la escala y además se vuelve más interpretable ya que no es un valor cuadrático (interpretación en porcentajes).
- En general, nos interesa el sentido ordinal que tiene el MSE y por lo tanto es la medida por defecto.

Sobreajuste



Sobreajuste

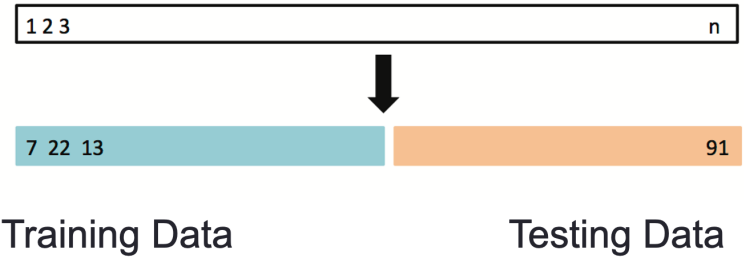
- Esto va en línea con el argumento del trade-off sesgo varianza.
- Añadir variables irrelevantes aumenta la varianza y omitir variables relevantes aumenta el sesgo.
- Entonces, entrenar modelo complejos va a reducir el sesgo y va a hacer que el modelo encaje a la perfección con los datos de entrenamiento.
- Aplicar este modelo a otros datos va a hacer que la varianza sea gigante (mala predicción fuera de muestra que es lo que nos interesa).

Métodos de remuestreo

- Cuando ajustamos un modelo a un solo conjunto de entrenamiento, corremos el peligro de sobreajuste.
- **Los métodos de remuestreo** son herramientas para hacer muestreos a partir de una muestra de entrenamiento para reajustar el modelo en cada muestra en aras de obtener más información sobre el modelo ajustado.

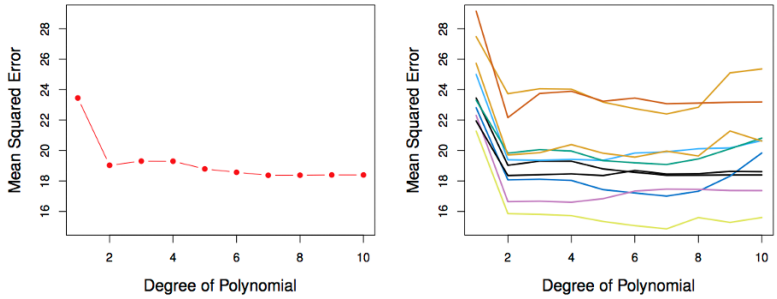
Conjunto de validación

- Suponga que queremos encontrar el conjunto de variables que dan la menor tasa de error de prueba.
- Si tenemos un conjunto de datos grande, podemos lograr esto partiendo los datos de forma aleatoria en prueba y entrenamiento.
- Si tenemos el modelo $y = f(x) + \varepsilon$ donde f es un polinomio de grado p.



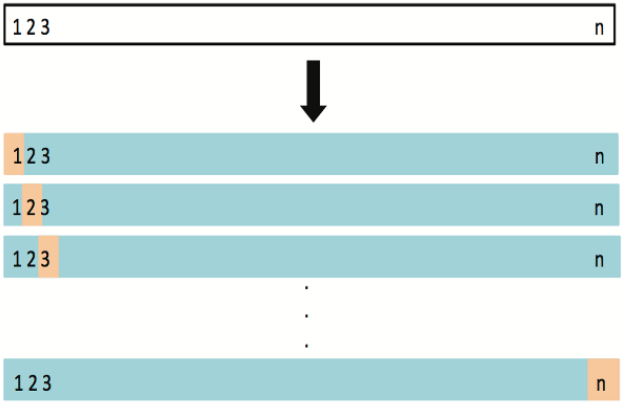
Conjunto de validación

- Ventajas: simple y fácil de implementar.
- Desventajas: El MSE de validación puede ser altamente vairable y solo un subconjunto de datos son usados para ajustar el modelo (el modelo ajusta mal cuando los datos son pocos).



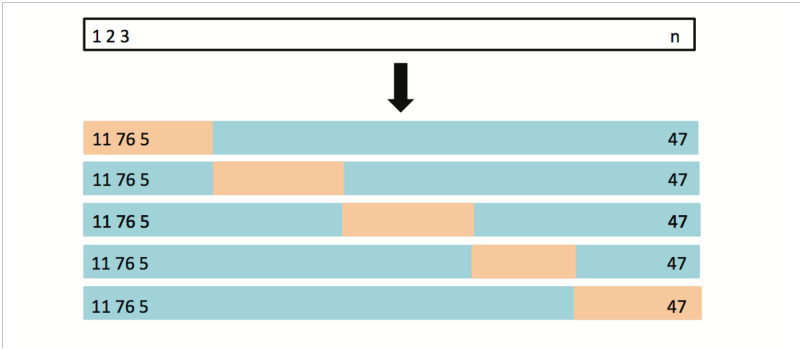
Leave-One-Out Cross Validation (LOOCV)

- Otra propuesta similar a la anterior que intenta solucionar el problema de mal desempeño por pocos datos.



K-fold Cross Validation

- LOOCV es muy intensivo en términos computacionales. Podemos hacer menos exigente esta forma de validación.



Pasos para K-fold

1. Dividir los datos en k partes ($N = \sum_{j=1}^{k} n_j$).
2. Ajustar el modelo dejando afuera la k-ésima parte ($f_{-k}(x)$.
3. Calcular error de predicción sobre la parte que queda por fuera (MSE_j).
4. Promediar y obtener $CV_k = \frac{1}{k} \sum_{j=1}^{k} MSE_j$

LOOCV es un caso especial de k-folds donde $k = n$.