Práctica 5.
Estimación pasiva de efectos causales.

Docente: Gustavo Landfried

Inferencia Bayesiana Causal 1
1er cuatrimestre 2025
ECyT - UNSAM

Índice

1. Introducción **2**
1.1. Intervenciones y experimentos 2
1.2. Variables de control backdoor 2
1.3. Frontdoor y do-calculus 4
1.4. Evaluación de modelos 5

2. Ejercicios **6**
2.1. Realidad causal 1: ejemplo básico. 6
2.2. Realidad causal 2. 7
2.3. Realidad causal 3. 7
2.4. Realidad causal 4. Ejemplo complejo. 8
2.5. Realidad causal 5 11
2.6. Realidad causal 6 12
2.7. Realidad causal 7 12
2.8. Realidad causal 8 12
2.9. Realidad causal complejo 13
2.10. Realidad causal frontdoor lineal 13
2.11. Efecto de X en M 14
2.12. Realidad causal frontdoor no-lineal 15

1. Introducción

¿Cómo estimar el efecto causal que un tratamiento $\text{do}(x)$ tiene sobre una variable objetivo y?

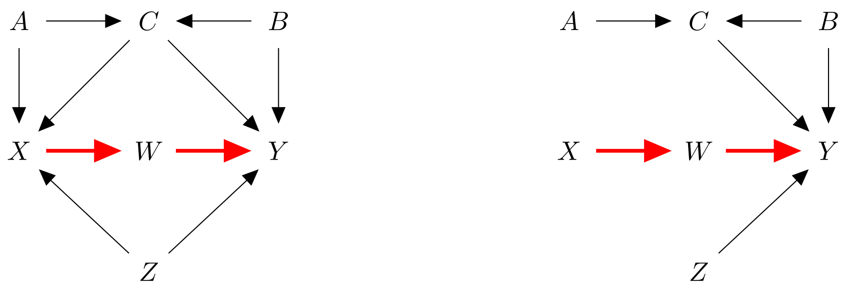
$$P(y|\text{do}(x)) \tag{1}$$

1.1. Intervenciones y experimentos

Supongamos que la figura 1a representa la realidad causal subyacente con la que se generan los datos observados sin intervenciones. Allí, la elección del tratamiento X se genera naturalmente de sus causas $\{A, C, Z\}$. Intervenir una variable significa modificar el mecanismo causal asignándole un valor predeterminado,

$$P(\text{do}(X = x)) = \mathbb{I}(X = x) \tag{2}$$

lo que se representar con la distribución de probabilidad indicadora. Este mecanismo causal de las intervenciones fijan un valor determinado del tratamiento independientemente de sus causas naturales. La figura 1 representa la realidad causal subyacente modificada por la intervenciones en X.



(a) Realidad causal sin intervención

(b) Realidad causal con intervención

Figura 1: En rojo se encuentra la relación causal que queremos estimar.

Las consecuencias que se producen en la estructura causal cuando fijamos una variable en una valor arbitrario (Eq. 2) son las mismas que cualquier tipo de asignación aleatoria,

$$P(\text{do}(X = x)) = P(x) \tag{3}$$

pues en ambos casos la variable x se vuelve independiente del resto de las variables que no forman parte de sus descendientes. Notar que en las dos realidades causales, sea con o sin intervención, X no es independiente de Y, $Y \not\perp\!\!\!\perp X$. Esto es deseable, pues esperamos que el tratamiento tenga un efecto sobre la variable objetivo. Sin embargo, cuando fijamos el valor o lo elegimos aleatoriamente estamos eligiendo la intervención de independientemente del *resultado potencial* que obtendría la variable objetivo, $Y'|\text{do}(x')$.

$$(Y'|\text{do}(x')) \perp\!\!\!\perp X \tag{4}$$

Por este motivo, cuando la industria farmacéutica busca aprobar una nueva droga, estima el efecto causal en *experimentos aleatorios*. Para ello se seleccionan las personas al azar y se las divide en dos grupos. Un grupo recibe el tratamiento, la nueva droga, y el otro recibe un placebo, algo que aparenta ser la droga pero que se sabe que no tiene efecto alguno. Los grupos pueden ser intercambiados y el resultado observado debería ser prácticamente el mismo.

1.2. Variables de control backdoor

En esta práctica queremos estimar el efecto causal en datos observados, sin hacer experimentos aleatorios. Para poder estimar este efecto causal en base a datos observados generados sin intervención va a ser necesario encontrar un contexto $\boldsymbol{Q}$ (un conjunto de variables de control) bajo el cual el tratamiento observado X pueda ser interpretado como si fuera un experimento aleatoria, es decir que su asignación sea independiente del *resultado potencial* que genera en la variable objetivo, $Y'|\text{do}(x')$.

$$(Y'|\text{do}(x')) \perp\!\!\!\perp X \mid \boldsymbol{Q} \tag{5}$$

La independencia que caracteriza a los experimentos aleatorios en la ecuación 4, y la independencia que queremos generar en base de datos sin intervenciones mediante un conjunto de variables de control $\boldsymbol{Q}$ en la ecuación 5, es entre el tratamiento X y el contrafactual $Y'|\text{do}(x')$.

Los contrafactuales por definición son variables no observables. Además, los contrafactuales admiten valores que son contradictorios con las observaciones factuales. Por eso los contrafactuales se modelan mediante variables *ocultas* adicionales, distintas a las variables factuales observables, compartiendo ambas las *mismas dependencias causales*. Por ejemplo, la figura 3 representa una especificación gráfica de la realidad causal subyacente original a la que se le agrega un contrafactual. Allí, el resultado potencial contrafactual $Y'|\text{do}(x')$ depende de las mismas variables $\{C, B, Z\}$ de las que depende Y y de los contrafactuales $\{W', X'\}$.

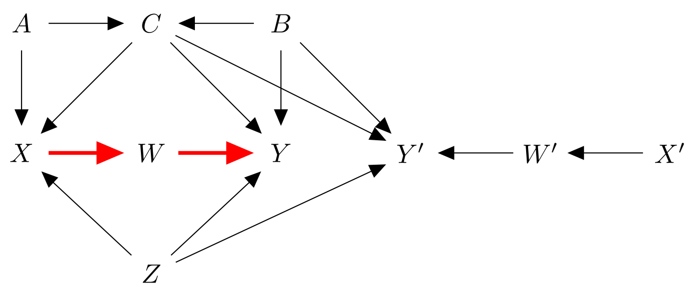


Figura 2: Modelo causal con contrafactuales.

Los modelos causales que incluyen contrafactuales se los conoce como "Twin networks". En ellas, para que un tratamiento observado X sea independiente del resultado potencial Y' es necesario contar con conjunto de variables de control $\boldsymbol{Q}$ que cierre todos los flujos de inferencia entre X e Y'. Es decir, primero tenemos que listar todos los caminos que conectan a X con Y' (notar que son los mismos caminos traseros que conectan a X con Y).

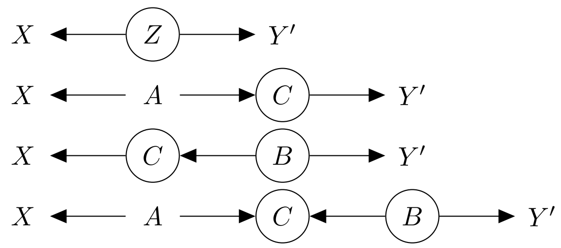


Figura 3: Todos los caminos que conectan a X con Y' (también caminos traseros entre X e Y). El conjunto de variables $\boldsymbol{Q} = \{B, C, Z\}$ cierra el flujo de inferencia en todos los caminos.

Para que los extremos de una cadena sean independientes entre sí es necesario que haya al menos un *collider* que no forme parte del conjunto de variables de control $\boldsymbol{Q}$ o que haya al menos una variable no *collider* que sí forme parte del conjunto de variables de control $\boldsymbol{Q}$. Cuando $\boldsymbol{Q}$ cierra todos los caminos traseros y no contiene ninguna variable descendiente a X decimos que $\boldsymbol{Q}$ cumple el *backdoor criterion*.

$$P(y|\text{do}(x)) := \sum_q \underbrace{P(y|\text{do}(x), q)P(q|\text{do}(x))}_{P(y,q|\text{do}(x))} \overset{*}{=} \sum_q P(y|x, q)P(q) \tag{6}$$

Gracias a que Q corta solo el flujo trasero podemos estimar $P(y|\text{do}(x), q) = P(y|x, q)$. Además, gracias a que Q no contiene descendientes de X podemos estimar $P(q|\text{do}(x)) = P(q)$. Estas dos transformaciones son casos especiales de las regla 2 y 3 de *do-calculus*, que discutiremos en la siguiente sección.

1.3. Frontdoor y do-calculus

Hay casos en los que no podemos observar las variables necesarias para bloquear el flujo trasero entre X e Y, pero igualmente es posible estimar el efecto causal en base de datos sin intervenciones. Por ejemplo, en la siguiente realidad causal subyacente (figura 4) la variable U no es observable y por lo tanto no contamos con un conjunto de control que nos permita estimar el efecto causal a través del *adjustment fórmula*.

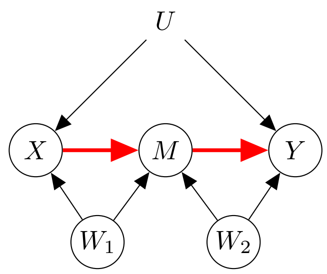


Figura 4: Si la variable U no es observable, hay que buscar un criterio alternativo para estimar el efecto causal entre X e Y.

Si bien en la figura 4 no podemos cerrar el flujo de inferencia trasero entre X e Y debido a que la variable U no es observable, sí podemos calcular por separado cada uno de los efectos causales,

$$P(m|\text{do}(x)) = \sum_{w_1} P(m|x, w_1)P(w_1)$$
$$P(y|\text{do}(m)) = \sum_{w_2} P(y|m, w_2)P(w_2)$$

el efecto causal de X en M (controlando por W_1) y el efecto causal de M en Y (controlando por W_2). Dado que podemos estimar ambos efectos causales delanteros, es posible estimar el efecto causal total,

$$P(y|\text{do}(x)) = \sum_{m} P(y|\text{do}(m))P(m|\text{do}(x)) \tag{7}$$

integrando todos los efectos causales que produce la variable M sobre Y, ponderado por la probabilidad de que ocurra m dado que intervenimos sobre $\text{do}(x)$.

El estimador de la ecuación 7 se conoce con el nombre de *frontdoor*. Aunque no sea tan común, es posible que existan otras situaciones donde sea necesario encontrar estimadores alternativos al backdoor y el frontdoor. En esos casos existen 3 reglas, el *do-calculus*, que resumen las trasformaciones necesarias y suficientes para determinar cualquier tipo de estimador de un efecto causal entre un tratamiento X y una variable objetivo Y dadas las co-variables $\boldsymbol{Q}$,

$$P(\boldsymbol{y}|\text{do}(\boldsymbol{t}), \boldsymbol{q})$$

siempre que exista, que haya alguna forma de identificar el efecto causal. Podemos resumir estas tres como:

1. *Ignorar una observación* cuando se cumple la **d-separación**. Por ejemplo, en los siguientes casos donde $\boldsymbol{x}$ es independiente de $\boldsymbol{v}$ dado $\boldsymbol{c}$, podemos ignorar $\boldsymbol{x}$.

$$\begin{aligned} P(\boldsymbol{v}|\boldsymbol{x}, \boldsymbol{c}) = P(\boldsymbol{v}|\boldsymbol{c}) \quad &\text{si} \quad \boldsymbol{V} \perp\!\!\!\perp_G \boldsymbol{X} \mid \boldsymbol{C} \\ P(\boldsymbol{v}|\text{do}(\boldsymbol{z}), \boldsymbol{x}, \boldsymbol{c}) = P(\boldsymbol{v}|\text{do}(\boldsymbol{z}), \boldsymbol{c}) \quad &\text{si} \quad \boldsymbol{V} \perp\!\!\!\perp_{G_{\overline{Z}}} \boldsymbol{X} \mid \boldsymbol{C} \end{aligned} \tag{$\substack{\text{Regla 1} \\ \text{Ignorar observación}}$}$$

El segundo caso dice simplemente que si hay una intervención, $\boldsymbol{x}$ tendrá que ser independiente de $\boldsymbol{v}$ dado $\boldsymbol{c}$ en el grafo intervenido para aplicar la regla.

2. *Interpretar una intervención como una observación* cuando **no hay flujo backdoor**. Por ejemplo, en los siguientes casos donde no hay flujo de inferencia trasero entre $\boldsymbol{x}$ y $\boldsymbol{v}$ dado $\boldsymbol{c}$, podemos ignorar interpretar $\text{do}(\boldsymbol{x})$ como $\boldsymbol{x}$.

$$\begin{aligned} P(\boldsymbol{v}|\text{do}(\boldsymbol{x}),\boldsymbol{c}) &= P(\boldsymbol{v}|\boldsymbol{x},\boldsymbol{c}) \quad \text{si} \quad \boldsymbol{V} \perp\!\!\!\perp_{G_{\underline{X}}} \boldsymbol{X} \mid \boldsymbol{C} \\ P(\boldsymbol{v}|\text{do}(\boldsymbol{z}),\text{do}(\boldsymbol{x}),\boldsymbol{c}) &= P(\boldsymbol{v}|\text{do}(\boldsymbol{z}),\boldsymbol{x},\boldsymbol{c}) \quad \text{si} \quad \boldsymbol{V} \perp\!\!\!\perp_{G_{\overline{Z},\underline{X}}} \boldsymbol{X} \mid \boldsymbol{C} \end{aligned} \quad \left(\substack{\text{Regla 2} \\ \text{Intervención como observación}}\right)$$

El segundo caso dice simplemente que si hay una intervención, el flujo de inferencia trasero entre $\boldsymbol{x}$ y $\boldsymbol{v}$ dado $\boldsymbol{c}$ también se tiene que cumplir en el grafo intervenido para aplicar la regla.

3. *Ignorar una intervención* cuando **no hay controles descendientes del tratamiento**. Cuando los controles $\boldsymbol{c}$ son no descendientes de la intervención $\boldsymbol{x}$, entonces la intervención no los afecta (primer caso).

$$\begin{aligned} P(\boldsymbol{c}|\text{do}(\boldsymbol{x})) &= P(\boldsymbol{c}) \quad \text{si} \quad \boldsymbol{C} \perp\!\!\!\perp_{G_{\overline{X}}} \boldsymbol{X} \\ P(\boldsymbol{c}|\text{do}(\boldsymbol{x}),\boldsymbol{v}) &= P(\boldsymbol{c}|\boldsymbol{v}) \quad \text{si} \quad \boldsymbol{C} \perp\!\!\!\perp_{G_{\overline{X(V)}}} \boldsymbol{X} \mid \boldsymbol{V} \\ P(\boldsymbol{c}|\text{do}(\boldsymbol{z}),\text{do}(\boldsymbol{x}),\boldsymbol{v}) &= P(\boldsymbol{c}|\text{do}(\boldsymbol{z}),\boldsymbol{v}) \quad \text{si} \quad \boldsymbol{C} \perp\!\!\!\perp_{G_{\overline{Z},\overline{X(V)}}} \boldsymbol{X} \mid \boldsymbol{V} \end{aligned} \quad \left(\substack{\text{Regla 3} \\ \text{Ignorar intervención}}\right)$$

donde $X(V)$ son las intervenciones no ancestros de V. En el resto de los casos se hace esa aclaración porque si la intervención es ancestra de V podría ocurrir que V sea collider y se abra el flujo de inferencia entre X y C.

1.4. Evaluación de modelos

Se suele decir que en inferencia causal, a diferencia del aprendizaje automático, no estamos interesados en predecir sino en evaluar el efecto causal. Esta afirmación resulta llamativa si tenemos en cuenta que bajo las reglas de la probabilidad los modelos causales (y cualquier otra hipótesis) son evaluados justamente en base a la capacidad predictiva. Quienes dicen que la inferencia causal es distinta al aprendizaje automático suelen mencionar los siguientes puntos.

a. En inferencia causal no hay datos supervisados pues nunca se observa la diferencia entre contrafactuales, por lo que ninguna predicción sobre efectos causales puede ser corroborada.

b. En inferencia causal nos vemos en la obligación de condicionar sobre un conjunto de control para evaluar el efecto causal, lo que no necesariamente mejora la predicción.

Ambos argumentos son solo parcialmente ciertos.

El hecho de que el contrafactual sea una variable oculta no tiene nada de especial. Todos los problemas de las ciencias empíricas, basadas en datos, contienen variables ocultas sobre las que tenemos incertidumbre de su valor real. Las reglas de la probabilidad son justamente el sistema de razonamiento que nos permite actualizar las creencias sobre hipótesis en general, y variables no observables en particular. Siempre que diseñamos modelos para estimar una variable oculta, los modelos alternativos se evalúan en base a la sorpresa que en cada una de las predicciones tiene el modelo respecto de las variables observadas. Si bien la diferencia entre contrafactuales no es observable para ningún individuo, sí podemos observar uno de los elementos de la diferencia: el factual.

Por otro lado, es un hecho que las correlaciones espurias (no causales) son útiles para predecir los nuevos datos, pero solo si ellos preservan la correlación espuria. Cuando los nuevos datos se generan con otra estructura causal subyacente, como ocurre cuando intervenimos en la realidad, las predicciones basadas en correlaciones espurias pasan a ser muy malas predicciones. En aprendizaje automático se conoce este problema de predicción como *out-of-distribution* (OOD). El hecho de que para predecir la intervención debamos condicionar sobre un conjunto de variables de control que cortar el flujo trasero (correlación espuria) tiene por objetivo hacer que la predicción se corresponda con la distribución de datos *out-of-distribution* de interés, basadas en efectos causales.

Es decir, el objetivo principal en inferencia causal es justamente mejorar la predicción y evaluar los modelos alternativos en base a su capacidad predictiva. Cuando los datos ya provienen de una distribución sin correlaciones espurias, como son los experimentos aleatorizados, no necesitamos hacer ninguna consideración especial para que evaluar los modelos: queremos el modelo que mejor predice los datos. Cuando los datos provienen de una distribución con correlaciones espurias, como son las observaciones naturales, necesitamos cortar el flujo trasero de inferencia para evaluar el desempeño que tendrían los modelos en la distribución causal de interés. En ambos casos estamos prediciendo contrafactuales, predicciones a priori de los factuales.

2. Ejercicios

Para inferir las distribuciones de probabilidad necesarias para estimar el efecto causal a partir de datos observados sin intervenciones podemos aplicar métodos no paramétricos (como un simple promedio condicional) o métodos paramétricos (como una regresión lineal, entre otros). Antes de empezar vamos a cargar algunas librerías que nos pueden resultar útiles.

```
import random
import numpy as np
from numpy.random import normal as noise
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal as normal
from scipy.stats import norm
```

Además, recomiendo cargar una librería de regresión bayesiana. En el repositorio `https://github.com/glandfried/bayesian-linear-model` se encuentra un versión completa del modelo lineal (que estima el ruido de los datos) bastante bien documentada. Solo deben clonar el repositorio y seguir las instrucciones del README.md. Cuando hagan eso, van a poder instalarla.

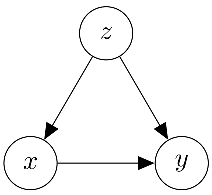
```
import linear_model as lm
from linear_model import BayesianLinearModel
```

Vamos a crear muestras de tamaño 5000.

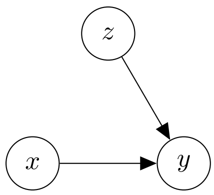
```
N = 5000
```

2.1. Realidad causal 1: ejemplo básico.

Estimar el efecto causal en el siguiente modelo.



(a) Realidad causal sin intervención



(b) Realidad causal con intervención

```
def simular1(N=5000, do_x=None):
    # Si hay intervencion, usar x = do_x, sino generarlo del modelo original
    Z = np.random.uniform(-3,3, size=N)
    if do_x is None:
        X = 1 + 3*Z + 2*Z**3 + np.random.normal(size=N,scale=6)
```

```python
    else:
        X = np.full(N, do_x)
    Y = -1 - 2*X + 6*Z**2 + np.random.normal(size=N,scale=1)
    return Z, X, Y

# Generamos los datos sin intervenciones
N = 5000
Z1s, X1s, Y1s = simular1(N)
```

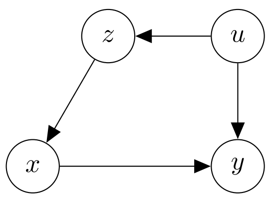
Mostramos un ejemplo de cómo usar la librería `lineal_model`.

```python
# Creamos el modelo
PHI1 = np.concatenate([
    np.ones(N).reshape(N, 1),      # c_0
    X1s.reshape(N, 1),             # c_x X
    (Z1s**2).reshape(N, 1)],       # c_z Z^2
axis=1)

# Ajustamos el modelo
blm1= BayesianLinearModel(basis=lambda x: x)
blm1.update(PHI1, Y1s.reshape(N,1) )

# Obtenemos las estimaciones
mean1 = blm1.location
cov1 = blm1.dispersion
ev1 = blm1.evidence()
```

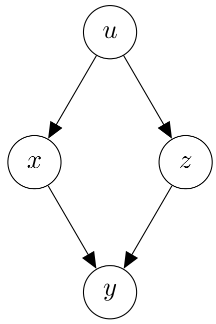
2.2. Realidad causal 2.




```python
U2s = np.random.uniform(-3,3, size=N)
Z2s = 3*U2s**3 + np.random.uniform(-1.5,1.5, size=N)
X2s =  3*Z2s + np.random.normal(size=N,scale=3)
Y2s = -1 - 2*X2s + 6*U2s**2 + np.random.normal(size=N,scale=1)
```

2.3. Realidad causal 3.

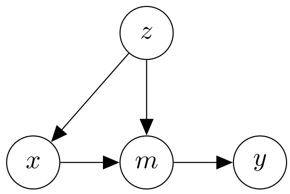
Estimar el efecto causal entre x e y.



```
U3s = np.random.uniform(-3,3, size=N)
Z3s = 3*U3s**2 + np.random.uniform(-1.5,1.5, size=N)
X3s =  3*U3s + np.random.normal(size=N,scale=3)
Y3s = -1 + 2*X3s + 5*Z3s**2 + np.random.normal(size=N,scale=1)
```

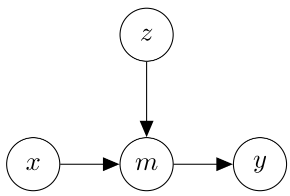
2.4. Realidad causal 4. Ejemplo complejo.

$$\begin{aligned} P(z) &= \mathcal{N}(z|0,1) \\ P(x|z) &= \mathcal{N}(x|z^2,1) \\ P(m|x,z) &= \mathcal{N}(m|2z^2+10x,1) \\ P(y|m) &= \mathcal{N}(y|-1+2m^2,1) \end{aligned}$$



El modelo 4 tiene relaciones no lineales anidadas entre las variables, lo cual puede ser difícil de interpretar inicialmente. Recordemos que queremos calcular la distribución de probabilidad $P(y|\mathrm{do}(x))$.

$$\begin{aligned} P(z) &= \mathcal{N}(z|0,1) \\ P(x|\mathrm{do}(x')) &= \mathbb{I}(x=x') \\ P(m|x,z) &= \mathcal{N}(m|2z^2+10x,1) \\ P(y|m) &= \mathcal{N}(y|-1+2m^2,1) \end{aligned}$$



Para simular N datos de estos modelos causales implementamos la siguiente función, y generamos 10000 datos del modelo original.

```
def simular4(N, do_x=None):
    # Si hay intervencion, usar x = do_x, sino generarlo del modelo original
    Z = np.random.normal(size=N, scale=1)
    if do_x is None:
        X = Z**2 + np.random.normal(size=N, scale=1)
    else:
        X = np.full(N, do_x)
    M = 2*Z**2 + 10*X + np.random.normal(size=N, scale=1)
    Y = -1 + 2*M**2 + np.random.normal(size=N, scale=1)
    return Z, X, M, Y

N = 10000
Z4s, X4s, M42, Y4s = simular4(N)
```

Nuestro objetivo es estimar el efecto causal que la variable X tiene sobre la variable objetivo Y usando únicamente éstos datos observados del modelo causal original. En esta práctica decidimos estimar los efectos causales usando regresiones lineales entre X e Y. Para cerrar los flujos traseros necesitamos controlar por Z.

$$P(y|x,z) = \mathcal{N}(y|f_4(x,z), \beta_4^{-1})$$

donde β_4 es la precisión (el inverso de la varianza). ¿Qué función $f_4(x,z)$ debemos elegir? En caso de que no sepamos, debemos recordar que gracias a que estamos trabajando con la regresión lineal bayesiana siempre poder evaluar los modelos alternativos en base a la evidencia, queremos el modelos que menor sorpresa tenga. Si revisamos con atención la estructura del modelo vamos a notar lo siguiente.

$$\begin{aligned} E[y|m] &= -1 + 2m^2 \\ &= -1 + 2(2z^2 + 10x)^2 \\ &= -1 + 200x^2 + 80xz^2 + 8z^4 = f_4(x,z) \end{aligned}$$

Esta es la regresión lineal que vamos a proponer. Si todo sale bien, deberíamos recuperar los coeficientes -1, 200, 80 y 8. ¿Pero como debemos interpretar estos coeficientes? En primer lugar, es importante recordar que debido a que z corta el flujo trasero de x en y, la regresión propuesta $P(y|x,z)$ puede ser interpretada como el efecto causal de x específico para un determinado valor de z.

$$P(y|\text{do}(x), z) = P(y|x,z)$$

Esta igualdad es la regla 2 del do-calculus, que nos permite reemplazar una intervención por una observación. Si consideramos a z como una constante, podemos notar que el efecto causal de x en y tiene una relación cuadrática.

$$E[y|\text{do}(x), z = \pm 1] = 7 + 80x + 200x^2 \qquad E[y|\text{do}(x), z = \pm 2] = 47 + 320x + 200x^2$$

¿Cuál es el efecto causal? Cuando conocemos z, simplemente es $E[y|x,z]$. Pero si no conocemos el valor de z, o si queremos calcular el efecto causal promedio en la población debemos integrar todos los posibles valores de z. Esto es lo que conocemos como el *backdoor criterion*. Sabemos que podemos estimar el efecto causal general de x en y integrando todos los efectos causales z-específicos.

$$P(y|\text{do}(x)) = \int_z P(y|x,z)P(z)$$

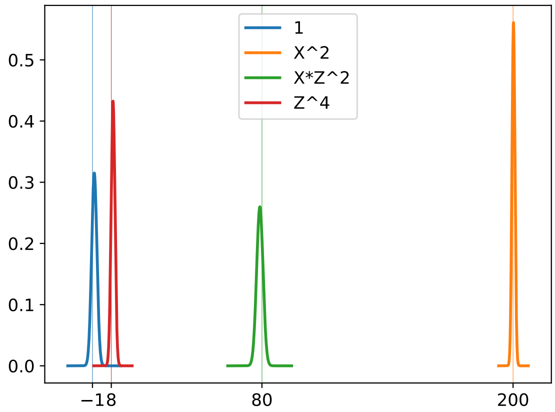
El factor $P(z)$ lo conocemos, es la gaussiana estándar $\mathcal{N}(z|0,1)$. Y $P(y|x,z)$ lo vamos a estimar haciendo una regresión lineal bayesiana de x y z a y.

```
# Base de la regresión
PHI4 = np.concatenate([
  np.ones(N).reshape(N, 1),
  (X4s**2).reshape(N, 1),
  (X4s*Z4s**2).reshape(N, 1),
  (Z4s**4).reshape(N, 1)],
axis=1)

# Modelo lineal
blm4= BayesianLinearModel(basis=lambda x: x)
blm4.update(PHI4, Y4s.reshape(N,1) )
print("Evidencia = ", blm4.evidence())

# Estimaciones de los pesos w
m4 = blm4.location
S4 = blm4.dispersion
```

Al graficar la media `m4` y la covarianza `S4` de las estimaciones de los pesos, veremos que se ajustan bien a los valores reales.



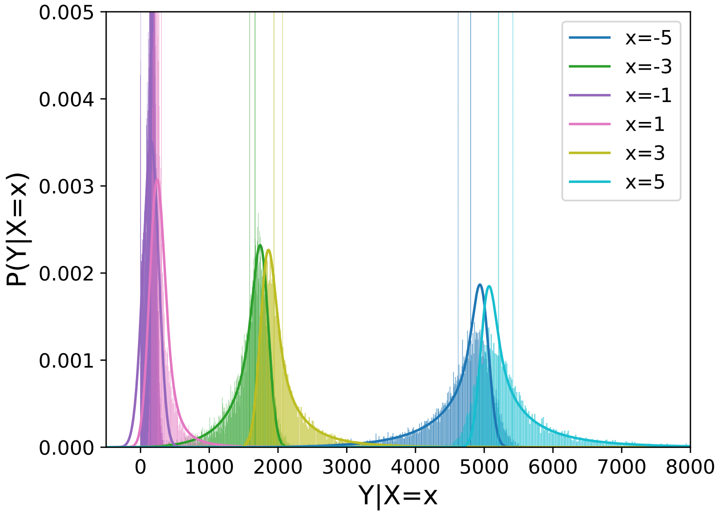
Ahora que sabemos que nuestras estimaciones son correctas, vamos a calcular numéricamente la distribución de probabilidad de la variable objetivo y dada la intervención $\mathrm{do}(x)$, $P(y|\mathrm{do}(x))$, mediante la *adjustment formula*.

```
def p4_Y_doX(y,x,blm4, z_grilla = np.arange(-4,4,0.01)):
    res = 0
    dz = z_grilla[1]-z_grilla[0]
    for z in z_grilla:
        pz = normal(0,1).pdf(z)
        mu_y_xz, S_y_xz, py_xz = blm4.predict(
          X=np.array([1,x**2,x*z**2,z**4]).reshape((1,4)),
          y=np.array([y]), variance=True)
        res += (pz*py_xz)*dz
    return res
```

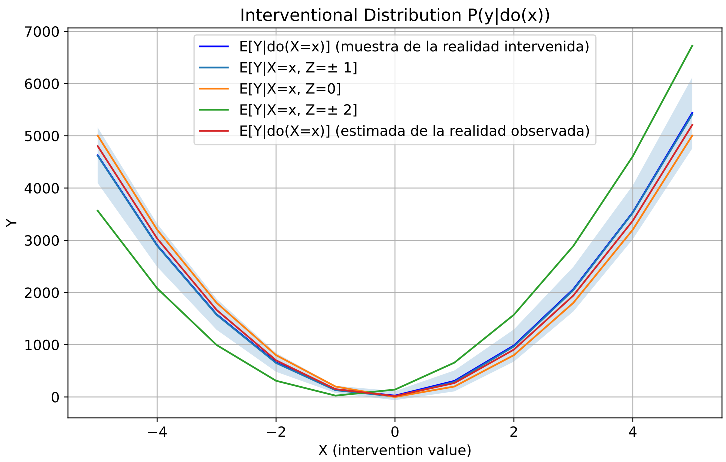
Para asegurarnos que la estimación del efecto causal que obtenemos mediante el modelo lineal y la *adjustment formula* es correcta, vamos a contrastarla contra datos simulados del modelo intervenido.

```
def Y_doX(x, N=10000):
    # Generar una muestra P(y|do(x))
    Z, X, M, Y = simular4(N, do_x=x)
    return Y
```

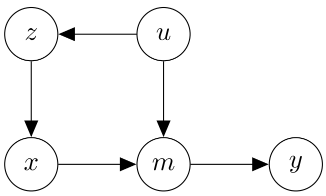
En la siguiente figura graficamos los histogramas `Y_doX` y las estimaciones `p4_Y_doX`.



Se puede ver que las estimaciones son buenas, pero no perfectamente. Esta imperfección es esperable en tanto la relación entre las variables es no lineal (en el anexo discutimos algunos detalles de esto). Sin embargo, es una estimación suficientemente buena para estimar el efecto causal. En la siguiente figura graficamos la media del efecto causal calculada de una muestra de la realidad intervenida, la estimaciones del efecto causal general estimada de la realidad observada, y la estimación de los efecto causales z-específicos.



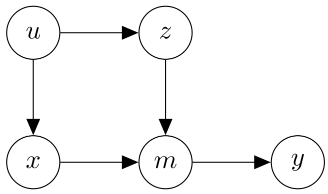
2.5. Realidad causal 5




```
U5s = np.random.uniform(-1.5,1.5, size=N)
Z5s =  U5s**2 + np.random.normal(size=N,scale=1)
X5s = 2*Z5s + np.random.normal(size=N,scale=1)
```

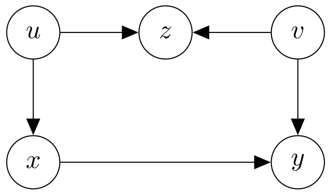
```
M5s = -2*U5s + 10*X5s  + np.random.normal(size=N,scale=1)
Y5s = -1 + 2*M5s + np.random.normal(size=N,scale=1)
```

2.6. Realidad causal 6



```
U6s = np.random.uniform(-1.5,1.5, size=N)
Z6s =  U6s**2 + np.random.normal(size=N,scale=1)
X6s = 2*U6s + np.random.normal(size=N,scale=1)
M6s = -2*Z6s + 10*X6s  + np.random.normal(size=N,scale=1)
Y6s = -1 + 2*M6s**2 + np.random.normal(size=N,scale=1)
```

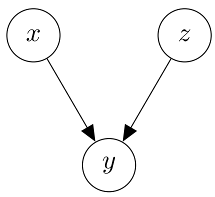
2.7. Realidad causal 7



```
U7s = np.random.uniform(-1.5,1.5, size=N)
V7s = np.random.uniform(-1.5,1.5, size=N)
Z7s =  U7s**2 + V7s + np.random.normal(size=N,scale=1)
X7s = 2*U7s + np.random.normal(size=N,scale=1)
M7s = -2*Z7s + 10*X7s  + np.random.normal(size=N,scale=1)
Y7s = -1 + 2*M6s**2 + V7s + np.random.normal(size=N,scale=1)
```

2.8. Realidad causal 8

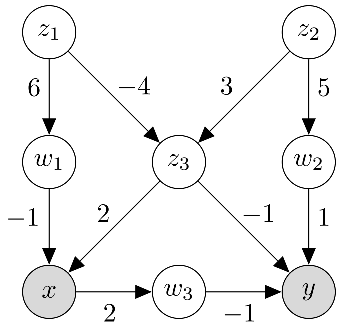
Comparar modelo controlando y no controlando por z.



```
Z8s = np.random.uniform(-1.5,1.5, size=N)
X8s = np.random.uniform(-1.5,1.5, size=N)
Y8s = 2*X8s**2 + 2*Z8s + np.random.normal(size=N,scale=1)
```

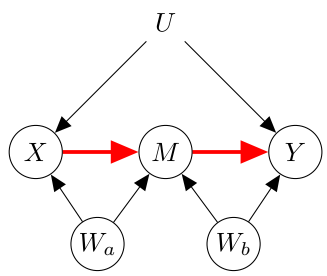
2.9. Realidad causal complejo

Estimar el efecto causal en el siguiente modelo.




```
N = 1000
z1 = np.random.uniform(-3,3, size=N)
w1 = 3*z1 + np.random.normal(size=N,scale=1)
z2 = np.random.uniform(-3,3, size=N)
w2 = 2*z2 + np.random.normal(size=N,scale=1)
z3 = -2*z1 + 2*z2 + np.random.normal(size=N,scale=1)
x = -1*w1 + 2*z3 + np.random.normal(size=N,scale=1)
w3 = 2*x + np.random.normal(size=N,scale=1)
y = 2 - 1*w3 - z3 + w2 + np.random.normal(size=N,scale=1)
```

2.10. Realidad causal frontdoor lineal



En este ejemplo no será posible aplicar el criterio backdoor porque consideraremos que la variable U es no observable.

```
def simular10(N=5000, do_x=None):
    # Si hay intervencion, usar x = do_x, sino generarlo del modelo original
    U = 4 + np.random.normal(size=N, scale=2)
    Wa = -1 + np.random.normal(size=N, scale=1)
    Wb = 3 + np.random.normal(size=N, scale=1)
    if do_x is None:
        X = 4*U + 1*Wa + np.random.normal(size=N,scale=1)
    else:
        X = np.full(N, do_x)
    M = -1 + 5*X + (-2)*Wa + 2*Wb + np.random.normal(size=N,scale=1)
    Y = 3 + 2*M + (-10)*Wb + np.random.normal(size=N,scale=1)
    return Wa, Wb, X, M, Y

```

```
N=5000
Wa10s, Wb10s, X10s, M10s, Y10s = simular10(N)
```

Calcular primero los efectos causales parciales aplicando el criterio backdoor.

$$P(y|\mathrm{do}(m)) \qquad\qquad P(m|\mathrm{do}(x)) \tag{8}$$

Y luego integrar los efectos causales de m para estimar el efecto causal total de x en y.

$$P(y|\mathrm{do}(x)) = \sum_m P(y|\mathrm{do}(m))P(m|\mathrm{do}(x)) \tag{9}$$

2.11. Efecto de X en M

Podemos hacer la regresión de X en M ajustando solo por W_a

```
PHI10_M_X = np.concatenate([
    np.ones(N).reshape(N, 1),
    (X10s).reshape(N, 1),
    (Wa10s).reshape(N, 1)]
, axis=1)

blm10_M_X= BayesianLinearModel(basis=lambda x: x)
blm10_M_X.update(PHI10_M_X, Y10s.reshape(N,1))
print(blm10_M_X.evidence())
print(blm10_M_X.location)
```

```
>>> blm10_M_X.evidence()
-16441.719813762465
>>> blm10_M_X.location
array([[-16.84956395],
       [  9.99958098],
       [ -4.11269564]])
```

En este caso la evidencia es -16441. En cambio, si ajustamos por W_a y W_b.

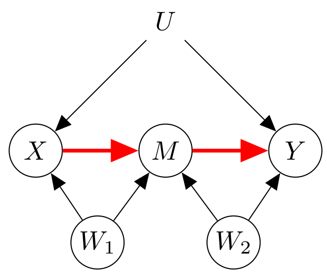
```
PHI10_M_X = np.concatenate([
    np.ones(N).reshape(N, 1),
    (X10s).reshape(N, 1),
    (Wa10s).reshape(N, 1),
    (Wb10s).reshape(N, 1)],
axis=1)

blm10_M_X= BayesianLinearModel(basis=lambda x: x)
blm10_M_X.update(PHI10_M_X, Y10s.reshape(N,1))
blm10_M_X.evidence()
blm10_M_X.location
```

```
>>> blm10_M_X.evidence()
-11116.067560259055
>>> blm10_M_X.location
array([[ 0.93175645],
       [10.00528902],
       [-4.02241328],
       [-6.01307222]])
```

La evidencia mejora significativamente y los coeficientes son los esperados. Continuar....

2.12. Realidad causal frontdoor no-lineal



En este ejemplo no será posible aplicar el criterio backdoor porque consideraremos que la variable U es no observable. Además, la relación entre las variables es no lineal.

```python
def simular11(N=5000, do_x=None):
    # Si hay intervencion, usar x = do_x, sino generarlo del modelo original
    U = 4 + np.random.normal(size=N, scale=2)
    Wa = -1 + np.random.normal(size=N, scale=1)
    Wb = 3 + np.random.normal(size=N, scale=1)
    if do_x is None:
        X = 4*U + 1*Wa**3 + np.random.normal(size=N,scale=1)
    else:
        X = np.full(N, do_x)
    M = -1 + 5*X + (-2)*Wa + 2*Wb + np.random.normal(size=N,scale=1)
    Y = 3 + 2*M**3 + (-10)*Wb + np.random.normal(size=N,scale=1)
    return Wa, Wb, X, M, Y
```