UNIVERSIDAD DEL CEMA
UCEMA

Machine Learning y Data Analytics

HW 7 - Grupo 1

Ian Amighini
Julieta Brey
Lorenzo Nastri
Camila Sobrino
Matias Rodriguez Brun

Profesor Titular: Sergio Pernice

Consigna

1) A partir del Jupyter notebook “VarCovar”, explicar cual es el teorema de la teoría de probabilidades que nos permite estimar valores esperado, varianzas, covarianzas, etc. a partir de datos empíricos.

2) Explicar en detalle por que, para calcular valores esperados de las variables, y sus varianzas, es suficiente tener las distribuciones marginales, pero para calcular la covarianza es imprescindible usar la distribución conjunta.

3) Generar un Jupyter notebook similar, pero en el que la variable aleatoria X puede tener 3 valores diferentes (x0, x1, x2) e Y cuatro valores diferentes (y0, y1, y2, y3). Pero antes de eso explicar por qué el número de eventos diferentes es 12, y por lo tanto la distribución de probabilidades conjunta tiene 12 probabilidades diferentes pij, i=0,1,2, j=0,1,2,3 (de las cuales 11 son independientes, y la doceava tiene que ser tal que la suma de las 12 de 1 (y todas sean positivas). Notar que la matriz varianza covarianza sigue siendo de 2x2, porque hay dos variables aleatorias.

4) Generar un Jupyter notebook similar, pero con 3 variables aleatorias, X, Y, Z, cada una de las cuales puede tomas dos valores diferentes. Pero antes de eso explicar por qué el número de eventos diferentes es 8 y por lo tanto la distribución de probabilidades conjunta tiene 8 probabilidades diferentes pijk, i=0,1, j=0,1, k=0,1 (de las cuales 7 son independientes, y la octava tiene que ser tal que la suma de las 8 de 1 (y todas sean positivas). Notar que la matriz varianza covarianza ahora es de 3x3, porque hay tres variables aleatorias.

Teorema de la Teoría de Probabilidades

En problemas de estadística aplicada, a menudo contamos sólo con datos muestrales de una población desconocida. Para inferir cantidades poblacionales como la esperanza E[X], la varianza Var(X) o la covarianza Cov(X,Y), empleamos estimadores basados en promedios y sumas de cuadrados de la muestra. El soporte teórico que valida esta práctica es la **Ley de los Grandes Números (LGN)**, que garantiza la convergencia de estos estimadores hacia los valores verdaderos a medida que *n* crece.

Ley de los Grandes Números

Sea $\{X_i\}_{i=1}^{\infty}$ una sucesión de variables aleatorias independientes e idénticamente distribuidas (i.i.d.) con $E[X_i] = \mu$ y $\mathrm{Var}(X_i) < \infty$. Entonces, para el promedio muestral

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i,$$

se cumple

$$\bar{X}_n \xrightarrow[n\to\infty]{} \mu \quad \text{(convergencia en probabilidad)}.$$

Más aún, si $g : \mathbb{R} \to \mathbb{R}$ verifica $E[|g(X)|] < \infty$, también

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i) \xrightarrow[n\to\infty]{} E[g(X)].$$

Estimadores de Momentos

A partir de la LGN se derivan naturalmente los siguientes estimadores:

1. **Esperanza**

$$\widehat{E[X]} = \bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{p} E[X].$$

2. **Varianza**

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2 \xrightarrow{p} \mathrm{Var}(X).$$

El divisor $n-1$ corrige el sesgo introducido por el uso de $\bar{X}_n$.

3. **Covarianza**

$$\widehat{\mathrm{Cov}}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)\,(Y_i - \bar{Y}_n) \ \xrightarrow{p}\ \mathrm{Cov}(X,Y).$$

Aunque la LGN garantiza la convergencia de los estimadores, el Teorema Central del Límite (TCL) describe la velocidad de esta convergencia y la distribución asintótica de Xn. Bajo condiciones razonables, lo que permite construir intervalos de confianza y realizar contrastes de hipótesis.

Gracias a la Ley de los Grandes Números y su extensión a funciones de la muestra, los estadísticos muestrales Xn, S^2 y Cov(X,Y) son **consistentes**, es decir, convergen al verdadero valor poblacional.

Distribuciones Marginales y Conjunta en el Cálculo de Momentos

Momentos univariados a partir de la distribución marginal

Sea X una variable aleatoria con función de densidad (o masa) marginal fX(x).

- **Esperanza: sólo interviene la marginal, pues promediamos X según su propia distribución.**

$$E[X] \ = \ \int_{-\infty}^{\infty} x\, f_X(x)\, dx.$$

- **Varianza: nuevamente, basta conocer fXf para medir la dispersión de X.**

$$\mathrm{Var}(X) \ = \ E\big[(X - E[X])^2\big] \ = \ \int_{-\infty}^{\infty} (x - E[X])^2\, f_X(x)\, dx.$$

Conclusión:

Para cualquier momento de orden k de X, $E[X^k] = \int x^k\, f_X(x)\, dx,$ es suficiente disponer de la función marginal fX.

Covarianza y la necesidad de la distribución conjunta

Para dos variables X y Y, la covarianza se define como

$$\mathrm{Cov}(X,Y) \;=\; E\big[(X - E[X])\,(Y - E[Y])\big] \;=\; E[XY] \;-\; E[X]\,E[Y]$$

- Los términos E[X]y E[Y] se obtienen de las marginales fX y fY, respectivamente.
- El núcleo del cálculo es → $E[XY] \;=\; \iint x\,y\,f_{X,Y}(x,y)\,dx\,dy,$ donde fX,Y(x,y) es la **densidad conjunta** de (X,Y).

Razón fundamental:

El valor medio de XY pondera la probabilidad de que simultáneamente X≈x y Y≈y. Esa probabilidad –es decir, la dependencia entre X y Y– **no** puede inferirse a partir de fX y fY por separado, excepto en el caso especial en que X e Y sean independientes (donde fX,Y=fX fY).

Implicaciones prácticas para el notebook "VarCovar"

Momentos univariados

- Se calculan con muestras de una sola variable: $\bar{X}_n \to E[X]$ y $S^2 \to \mathrm{Var}(X).$
- Sólo requieren la distribución marginal empírica de X

Covarianza

El estimador $\widehat{\mathrm{Cov}}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)\,(Y_i - \bar{Y}_n)$ explota la frecuencia empírica de los pares (Xi,Yi).

Es imprescindible emplear la **distribución conjunta** (o datos emparejados) para capturar la relación lineal entre ambas variables.

Resumen:

- **Distribución marginal → basta para esperanzas y varianzas univariadas.**

- **Distribución conjunta → necesaria para covarianzas y demás momentos multivariados.**

Jupyter NoteBook 1

Explicación de los 12 eventos

Cuando consideramos dos variables discretas, X con tres posibles valores (0, 1 y 2) e Y con cuatro (0, 1, 2 y 3), cada combinación de un valor de X con uno de Y constituye un suceso único y mutuamente excluyente: por ejemplo, “X=0 y Y=0”, “X=0 y Y=1”, …, hasta “X=2 y Y=3”. En total hay 3×4=12 de estos sucesos elementales, de modo que la distribución conjunta debe asignarles una probabilidad pij distinta para cada par (i,j). Es decir, disponemos de doce probabilidades p00,p01,…,p23. Sin embargo, para que esto sea una distribución válida, esas doce magnitudes deben sumar exactamente 1 (y ser no negativas), lo que impone una sola ecuación de normalización:

$$\sum_{i=0}^{2}\sum_{j=0}^{3} p_{ij} = 1.$$

Debido a esta restricción global, bastan once parámetros libres: uno puede elegir arbitrariamente once de los pij (siempre comprobando que queden positivos), y la duodécima probabilidad queda obligatoriamente determinada para que la suma total sea uno. Así, la tabla de probabilidades conjunta es de dimensión 3×4=12, pero sólo once de sus entradas son independientes; la última se calcula como

$$p_{2,3} = 1 - \sum_{\substack{0\le i\le 2,\ 0\le j\le 3\\ (i,j)\ne(2,3)}} p_{ij}.$$

NoteBook

✓ 0s [1]

```
import numpy as np

# Definir los valores posibles
x_vals = np.array([0, 1, 2])    # x0, x1, x2
y_vals = np.array([0, 1, 2, 3]) # y0, y1, y2, y3
```

```python
# 3. Distribución conjunta de ejemplo - Generamos una matriz 3x4 de valores aleatorios y la normalizamos.

P = np.random.rand(len(x_vals), len(y_vals))
P = P / P.sum()  # Normalizar a suma 1
print("Distribución conjunta P:")
print(P)
```

```
Distribución conjunta P:
[[0.02956629 0.13923808 0.0564028  0.05831484]
 [0.05747919 0.15613068 0.14529197 0.11599674]
 [0.05612531 0.02214805 0.1209679  0.04233816]]
```

```python
# 4. Distribuciones marginales
# - p_X(i) = Σ_j P[i,j]
# - p_Y(j) = Σ_i P[i,j]

# Marginal de X (suma por filas)
p_x = P.sum(axis=1)
# Marginal de Y (suma por columnas)
p_y = P.sum(axis=0)

print("\nMarginal de X:", p_x)
print("Marginal de Y:", p_y)
```

```
Marginal de X: [0.283522   0.47489859 0.24157942]
Marginal de Y: [0.14317079 0.31751681 0.32266267 0.21664973]
```

```python
#5. Cálculo de esperanzas

# E[X] = Σ_i x_i p_X(i)
# E[Y] = Σ_j y_j p_Y(j)


E_X = np.dot(x_vals, p_x)
E_Y = np.dot(y_vals, p_y)
print(f"E[X] = {E_X:.4f}")
print(f"E[Y] = {E_Y:.4f}")
```

```
E[X] = 0.9581
E[Y] = 1.6128
```

```python
#6. Cálculo de varianzas
# Var(X) = Σ_i (x_i - E[X])^2 p_X(i)
# Var(Y) = Σ_j (y_j - E[Y])^2 p_Y(j)

Var_X = np.dot((x_vals - E_X)**2, p_x)
Var_Y = np.dot((y_vals - E_Y)**2, p_y)
print(f"Var(X) = {Var_X:.4f}")
print(f"Var(Y) = {Var_Y:.4f}")
```

```
Var(X) = 0.5233
Var(Y) = 0.9569
```

```python
#7. Cálculo de la covarianza

# Cov(X,Y) = E[XY] - E[X]E[Y], donde
# E[XY] = Σ_i Σ_j x_i y_j P[i,j]

# E[XY]
E_XY = sum(x * y * P[i, j] for i, x in enumerate(x_vals) for j, y in enumerate(y_vals))
Cov_XY = E_XY - E_X * E_Y
print(f"Cov(X,Y) = {Cov_XY:.4f}")
```

```
Cov(X,Y) = 0.0318
```

```python
# 8. Matriz varianza-covarianza

# Construimos la matriz 2x2 - Notar que la matriz varianza covarianza sigue siendo de 2x2, porque hay dos variables aleatorias.

Sigma = np.array([[Var_X, Cov_XY], [Cov_XY, Var_Y]])
print("\nMatriz Var-Cov:")
print(Sigma)
```

```
Matriz Var-Cov:
[[0.52334223 0.03175477]
 [0.03175477 0.95691915]]
```

Jupyter NoteBook 2

En el caso de tres variables aleatorias discretas X,Y,Z, cada una con **dos posibles valores** (por ejemplo 0 y 1), el **espacio muestral conjunto** está formado por todos los tripletes (xi,yj,zk)

con $i \in \{0,1\}, \quad j \in \{0,1\}, \quad k \in \{0,1\}.$

El número total de combinaciones es el producto del número de valores de cada variable:

$$\underbrace{2}_{\#X} \times \underbrace{2}_{\#Y} \times \underbrace{2}_{\#Z} = 8.$$

Ocho probabilidades conjuntas p ijk

A cada uno de esos 8 eventos (X=i,Y=j,Z=k) le asignamos una probabilidad

pijk = P(X=i, Y=j, Z=k) con índices i,j,k ∈ {0,1}. En total hay **8** de ellas:

$$\{p_{000},\ p_{001},\ p_{010},\ p_{011},\ p_{100},\ p_{101},\ p_{110},\ p_{111}\}.$$

Para que {pijk} formen una **distribución de probabilidad** válida, deben cumplirse dos axiomas:

1. **No negatividad** $p_{ijk} \geq 0 \quad \forall i, j, k.$

2. **Normalización** $$\sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} p_{ijk} = 1.$$

Grados de libertad

- Partimos de 8 variables p ijk
- La **condición de suma uno** es una ecuación lineal que las une.
- Por tanto, hay **8 – 1 = 7 grados de libertad:** podemos elegir libremente 7 de las probabilidades, y la octava **queda determinada** automáticamente para que la suma sea 1.

¿Por qué importa?

- Cada pijk cuantifica la probabilidad de un evento conjunto (X, Y, Z).
- Conocer las 7 primeras y aplicar la normalización **asegura que la distribución completa sea coherente.**
- Este esquema es la base para definir la **distribución conjunta** y, a partir de ella, derivar marginales, esperanzas, varianzas, covarianzas y la matriz var–cov 3×3.

```
[8]  import numpy as np

     # 1. Definir los valores posibles para cada variable
     x_vals = np.array([0, 1])
     y_vals = np.array([0, 1])
     z_vals = np.array([0, 1])
```

```
# 2. Generar distribución conjunta aleatoria y normalizar
P = np.random.rand(2, 2, 2)
P /= P.sum()
print("Distribución conjunta P:", P)
```

```
Distribución conjunta P: [[[0.17524946 0.15783242]
  [0.16183824 0.11335305]]

 [[0.15583243 0.07961889]
  [0.02664769 0.12962782]]]
```

```
# 3. Calcular distribuciones marginales
p_x = P.sum(axis=(1, 2))  # marginal de X
p_y = P.sum(axis=(0, 2))  # marginal de Y
p_z = P.sum(axis=(0, 1))  # marginal de Z
print("Marginal X:", p_x)
print("Marginal Y:", p_y)
print("Marginal Z:", p_z)
```

```
Marginal X: [0.60827316 0.39172684]
Marginal Y: [0.56853321 0.43146679]
Marginal Z: [0.51956782 0.48043218]
```

```
# 4. Calcular esperanzas
E_X = np.dot(x_vals, p_x)
E_Y = np.dot(y_vals, p_y)
E_Z = np.dot(z_vals, p_z)
print(f"E[X] = {E_X:.4f}")
print(f"E[Y] = {E_Y:.4f}")
print(f"E[Z] = {E_Z:.4f}")
```

```
E[X] = 0.3917
E[Y] = 0.4315
E[Z] = 0.4804
```

```
# 5. Calcular varianzas
Var_X = np.dot((x_vals - E_X)**2, p_x)
Var_Y = np.dot((y_vals - E_Y)**2, p_y)
Var_Z = np.dot((z_vals - E_Z)**2, p_z)
print(f"Var(X) = {Var_X:.4f}")
print(f"Var(Y) = {Var_Y:.4f}")
print(f"Var(Z) = {Var_Z:.4f}")
```

```
Var(X) = 0.2383
Var(Y) = 0.2453
Var(Z) = 0.2496
```


```
# 5. Calcular varianzas
Var_X = np.dot((x_vals - E_X)**2, p_x)
Var_Y = np.dot((y_vals - E_Y)**2, p_y)
Var_Z = np.dot((z_vals - E_Z)**2, p_z)
print(f"Var(X) = {Var_X:.4f}")
print(f"Var(Y) = {Var_Y:.4f}")
print(f"Var(Z) = {Var_Z:.4f}")
```

```
Var(X) = 0.2383
Var(Y) = 0.2453
Var(Z) = 0.2496
```


Código

1_HW7_VarCovar.ipynb