Unidad III - Ingeniería de características

Germán Braun

Facultado de Informática - Universidad Nacional del Comahue

german.braun@fi.uncoma.edu.ar

19 de septiembre de 2025

Agenda

Machine Learning en Práctica (cont'd)

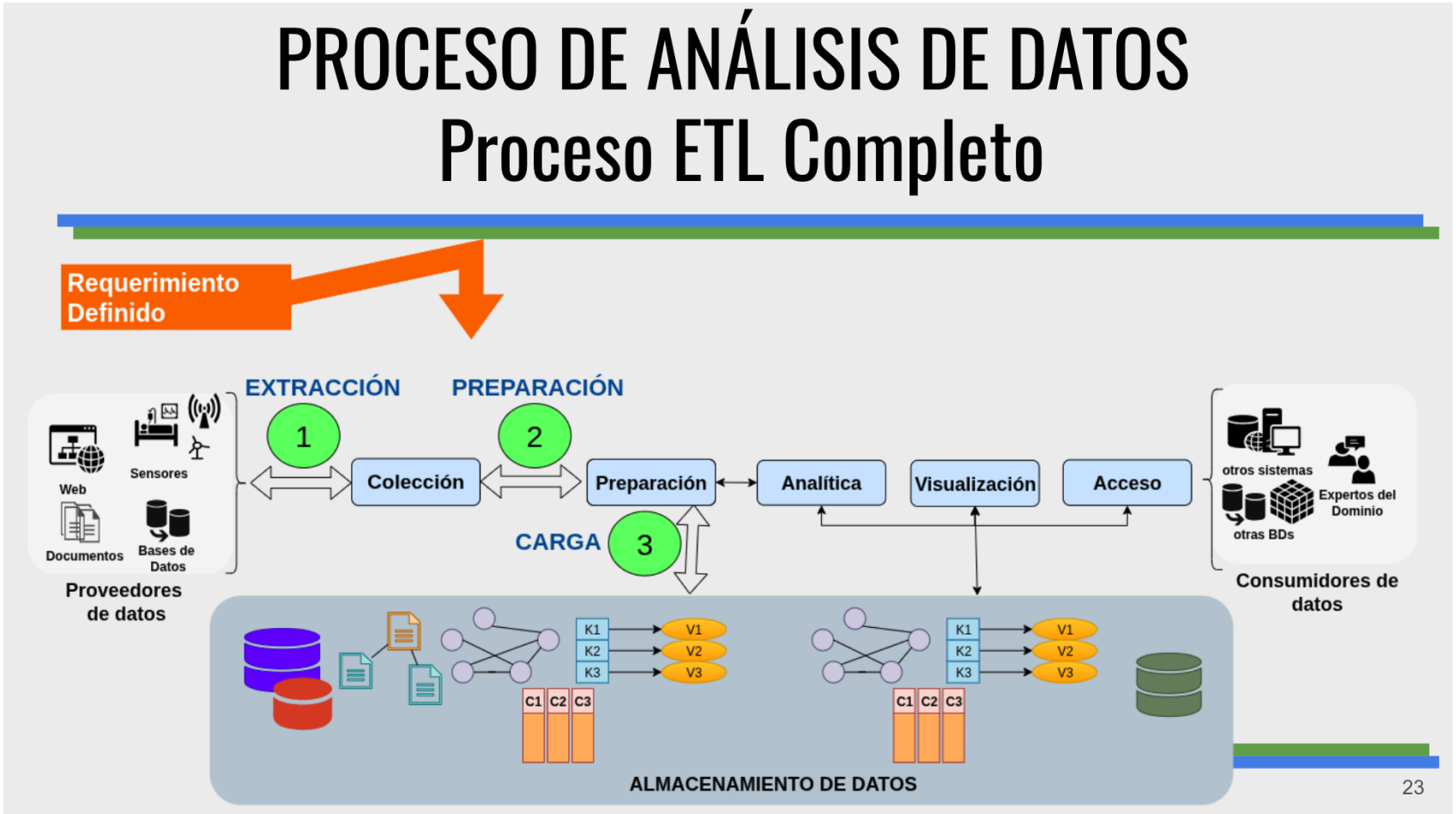
¡Recordatorio!

El aprendizaje automático es un proceso de prueba y error.

ML en la práctica (recap)

¡Recordatorio!

- Entender el dominio, conocimiento previo y metas.
- Pre-procesar datos (intregar, seleccionar, limpiar, dividir dataset)
- Entrenar modelos (comenzando por el más simple posible)
- Interpretar resultados
- Consolidar y deplegar conocimiento descubierto
- Ciclar sobre estos pasos anteriores

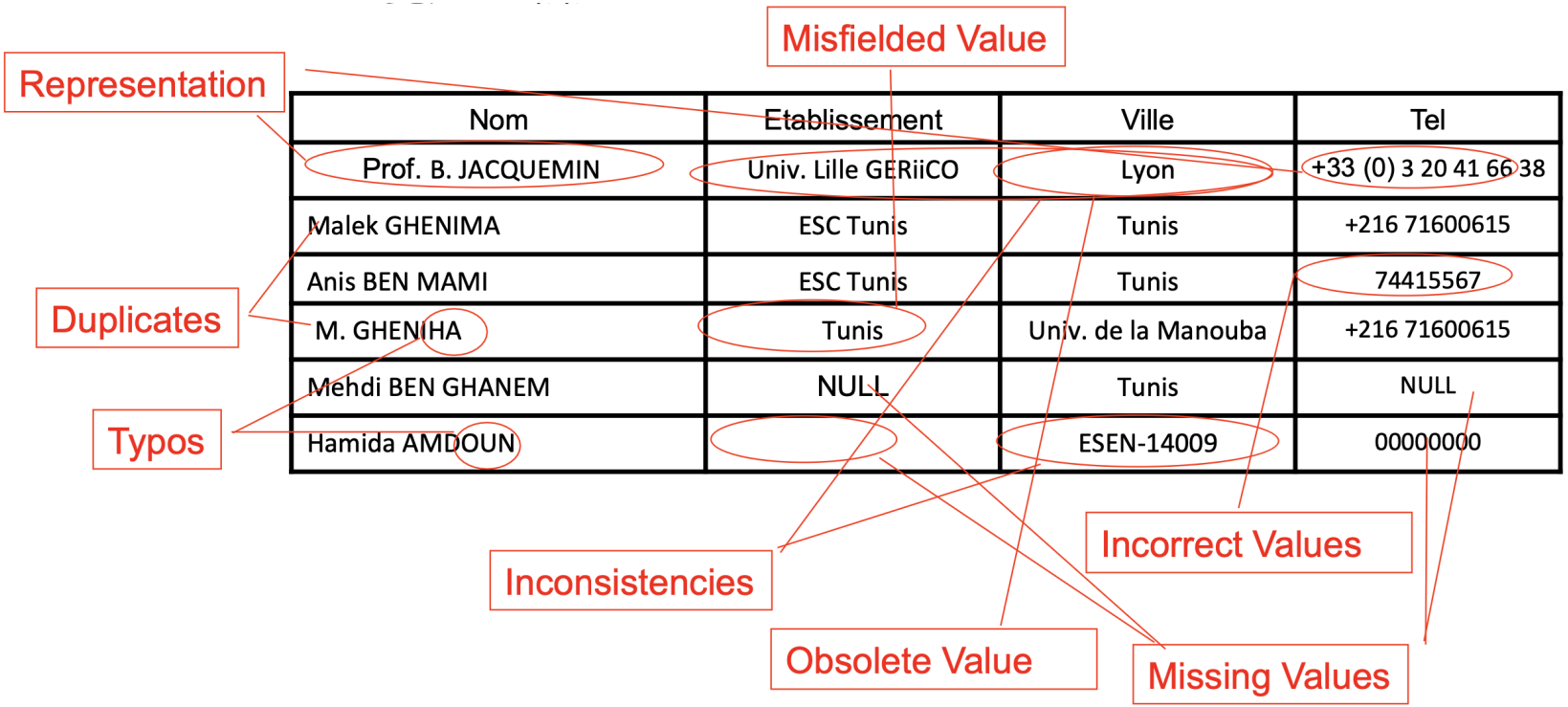


(*) del curso EXTRACCIÓN, PREPARACIÓN Y ALMACENAMIENTO DE LOS DATOS. Créditos: Agustina Buccella

ML en la práctica

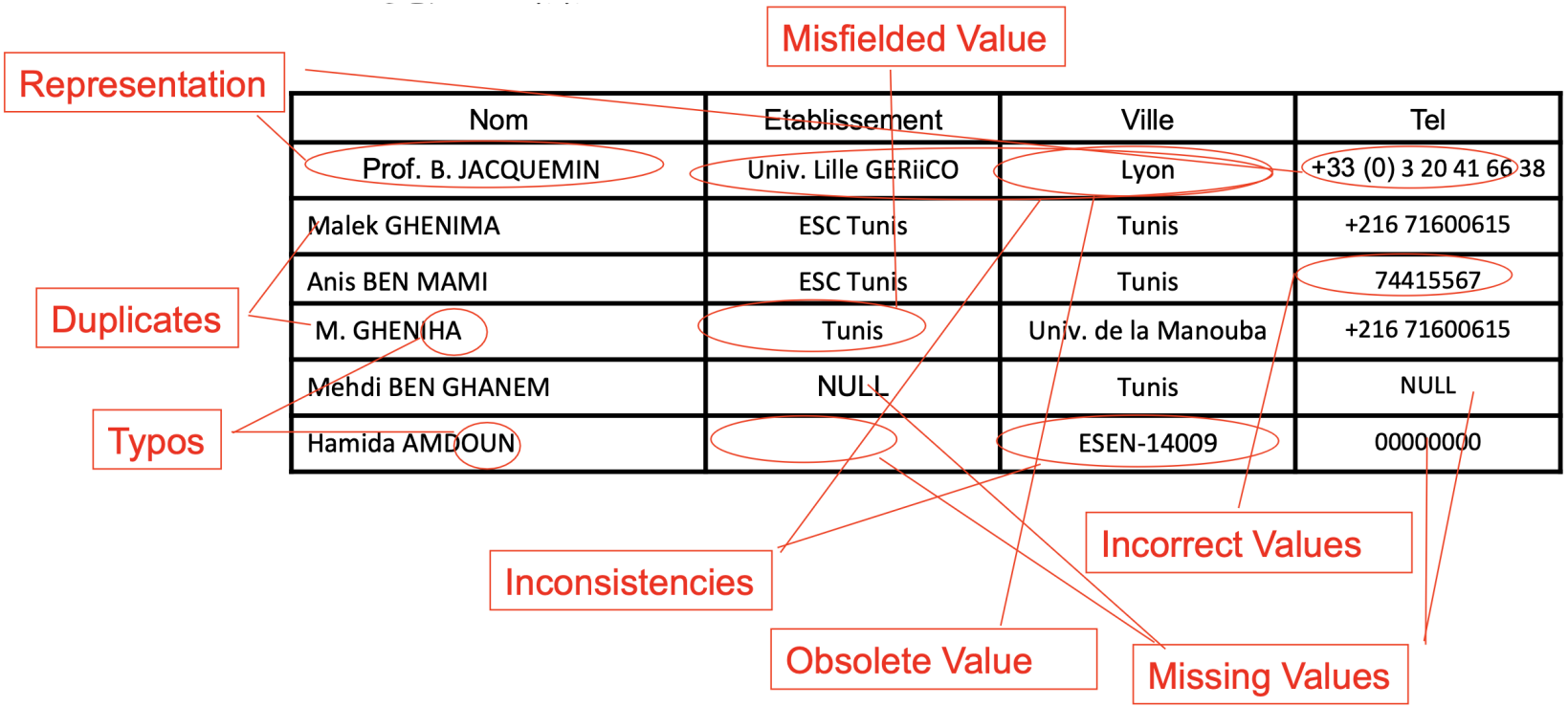
Data are messy!

Data are messy!



Nom	Etablissement	Ville	Tel
Prof. B. JACQUEMIN	Univ. Lille GERiiCO	Lyon	+33 (0) 3 20 41 66 38
Malek GHENIMA	ESC Tunis	Tunis	+216 71600615
Anis BEN MAMI	ESC Tunis	Tunis	74415567
M. GHENIHA	Tunis	Univ. de la Manouba	+216 71600615
Mehdi BEN GHANEM	NULL	Tunis	NULL
Hamida AMDOUN		ESEN-14009	00000000

Data are messy!



Nom	Etablissement	Ville	Tel
Prof. B. JACQUEMIN	Univ. Lille GERiiCO	Lyon	+33 (0) 3 20 41 66 38
Malek GHENIMA	ESC Tunis	Tunis	+216 71600615
Anis BEN MAMI	ESC Tunis	Tunis	74415567
M. GHENIHA	Tunis	Univ. de la Manouba	+216 71600615
Mehdi BEN GHANEM	NULL	Tunis	NULL
Hamida AMDOUN		ESEN-14009	00000000

(*) Créditos de la imagen: Machine Learning-Based Data Cleaning (Laure Berti-Equille)

Selección de atributos

En la práctica

El aprendizaje automático aplicado es básicamente ingeniería de características (Andrew Ng)

Selección de atributos

Por simplicidad, vamos a suponer que nuestros datasets ya fueron preprocesados y están en *"buena forma"* para procesar las entradas de los algoritmos de aprendizaje

Selección de atributos

Por simplicidad, vamos a suponer que nuestros datasets ya fueron preprocesados y están en *"buena forma"* para procesar las entradas de los algoritmos de aprendizaje

Por lo tanto, nos enfocaremos en como preparar las entradas para los algoritmos y en como seleccionar atributos que sean **relevantes** para el entrenamiento de los modelos.

Selección de atributos

Por simplicidad, vamos a suponer que nuestros datasets ya fueron preprocesados y están en *"buena forma"* para procesar las entradas de los algoritmos de aprendizaje

Por lo tanto, nos enfocaremos en como preparar las entradas para los algoritmos y en como seleccionar atributos que sean **relevantes** para el entrenamiento de los modelos.

Selección de atributos (cont'd)

- ... **Reducción de la dimensionalidad**

Selección de atributos (cont'd)

o ... **Reducción de la dimensionalidad**

La **ingeniería de atributos** es el proceso en el cual se preparan y curan los conjuntos de atributos para alimentar a los algoritmos de aprendizaje.

Selección de atributos (cont'd)

o ... **Reducción de la dimensionalidad**

La **ingeniería de atributos** es el proceso en el cual se preparan y curan los conjuntos de atributos para alimentar a los algoritmos de aprendizaje.

La **selección de atributos** es parte de este proceso, y está abocada a la elección de los atributos que puedan tener el mayor impacto en el modelo

Selección de atributos (cont'd)

o ... **Reducción de la dimensionalidad**

La **ingeniería de atributos** es el proceso en el cual se preparan y curan los conjuntos de atributos para alimentar a los algoritmos de aprendizaje.

La **selección de atributos** es parte de este proceso, y está abocada a la elección de los atributos que puedan tener el mayor impacto en el modelo

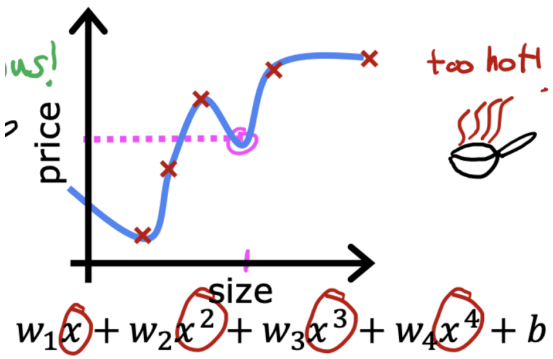
La mejor manera de seleccionar atributos es manualmente... y basado en un conocimiento profundo del dominio y del significado de dichos atributos. Sin embargo, métodos automáticos también pueden ser útiles

Selección de atributos (cont'd)

- mejora la **performance** de los algoritmos de aprendizaje

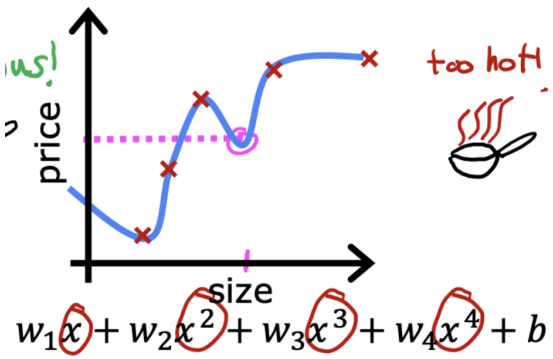
Selección de atributos (cont'd)

- mejora la **performance** de los algoritmos de aprendizaje
- produce una representación más compacta y entendible, focalizando sobre las variables más relevantes
- reduce el ***overfitting*** →



Selección de atributos (cont'd)

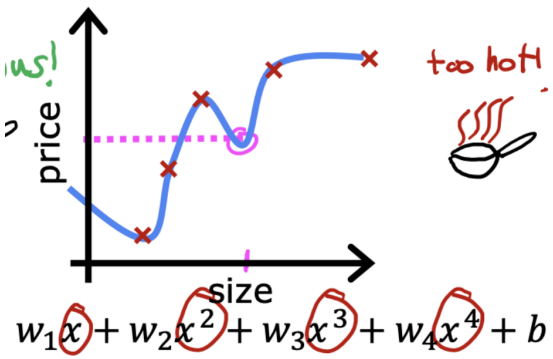
- mejora la **performance** de los algoritmos de aprendizaje
- produce una representación más compacta y entendible, focalizando sobre las variables más relevantes
- reduce el ***overfitting*** →



- reduce tiempo de aprendizaje

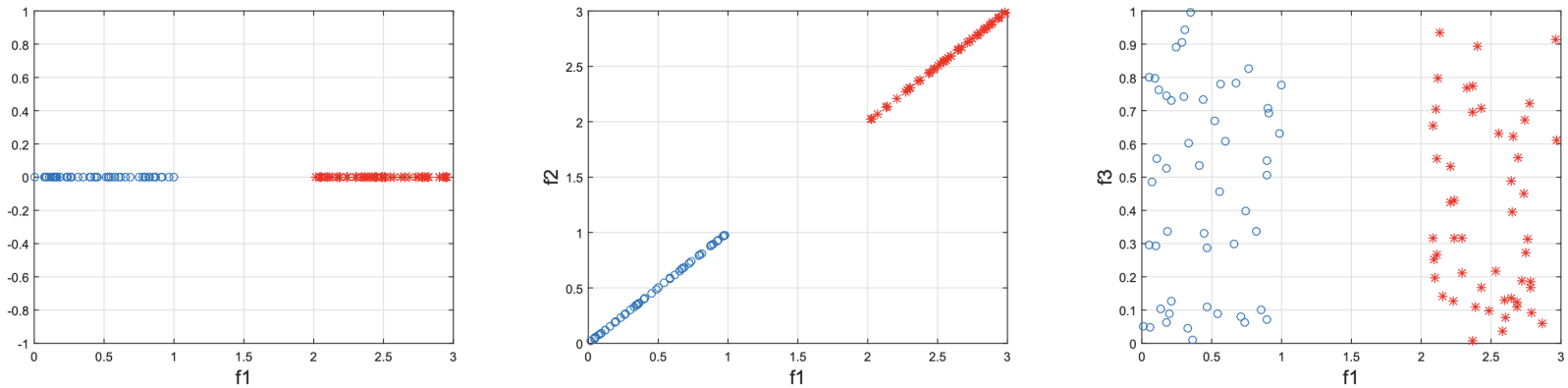
Selección de atributos (cont'd)

- mejora la **performance** de los algoritmos de aprendizaje
- produce una representación más compacta y entendible, focalizando sobre las variables más relevantes
- reduce el ***overfitting*** →



- reduce tiempo de aprendizaje
- incrementa interoperabilidad y facilita implementación

Relevancia de un atributo



(a) relevant feature f_1 (b) redundant feature f_2 (c) irrelevant feature f_3

Figura 1.1: Image de [2]

a) discrimina dos clases >> aporta información

b) es redudante ya que f_2 está fuertemente correlacionado con f_1 >> aporta información pero ya contenida en otras

c) no puede clusterizar >> no aporta información *útil*

Métodos de Selección

Aprendizaje Supervisado

Debido a que los atributos ya están identificados, el objetivo es identificar aquellos atributos de entrada con mayor impacto sobre la variable objetivo $\longrightarrow$ **correlación** (no implica causalidad!)

Métodos de Selección

Aprendizaje Supervisado

Debido a que los atributos ya están identificados, el objetivo es identificar aquellos atributos de entrada con mayor impacto sobre la variable objetivo → **correlación** (no implica causalidad!)

- Filtros

Métodos de Selección

Aprendizaje Supervisado

Debido a que los atributos ya están identificados, el objetivo es identificar aquellos atributos de entrada con mayor impacto sobre la variable objetivo → **correlación** (no implica causalidad!)

- Filtros
- Wrapper

Métodos de Selección

Aprendizaje Supervisado

Debido a que los atributos ya están identificados, el objetivo es identificar aquellos atributos de entrada con mayor impacto sobre la variable objetivo → **correlación** (no implica causalidad!)

- Filtros
- Wrapper
- Embebido

Métodos de Selección - Filtros

Estos modelos dependen de criterios estadísticos aplicados a los datos tales como *distancia, dependencia, consistencia, correlación*. Son **independiente** del algoritmo de aprendizaje y computacionalmente eficientes.

Métodos de Selección - Filtros

Estos modelos dependen de criterios estadísticos aplicados a los datos tales como *distancia, dependencia, consistencia, correlación*. Son **independiente** del algoritmo de aprendizaje y computacionalmente eficientes.

- **Umbral de varianza** (*Variance threshold / F-test*): remueve atributos con baja varianza, es decir, aquellos mayormente constantes

Métodos de Selección - Filtros

Estos modelos dependen de criterios estadísticos aplicados a los datos tales como *distancia, dependencia, consistencia, correlación*. Son **independiente** del algoritmo de aprendizaje y computacionalmente eficientes.

- **Umbral de varianza** (*Variance threshold / F-test*): remueve atributos con baja varianza, es decir, aquellos mayormente constantes
- **Chi-cuadrado** (*Chi-Square Test*): mide la independencia entre un atributo y la variable target (clase). Solo para atributos categóricos [más]

Métodos de Selección - Filtros

Estos modelos dependen de criterios estadísticos aplicados a los datos tales como *distancia, dependencia, consistencia, correlación*. Son **independiente** del algoritmo de aprendizaje y computacionalmente eficientes.

- **Umbral de varianza** (*Variance threshold / F-test*): remueve atributos con baja varianza, es decir, aquellos mayormente constantes
- **Chi-cuadrado** (*Chi-Square Test*): mide la independencia entre un atributo y la variable target (clase). Solo para atributos categóricos [más]
- **Correlación** (*Correlation Coefficient*): remueve atributos altamente correlacionados, manteniendo solo uno de las involucradas. Puede ser aplicada a atributos numéricos y categóricos.

Métodos de Selección - Filtros

Estos modelos dependen de criterios estadísticos aplicados a los datos tales como *distancia, dependencia, consistencia, correlación*. Son **independiente** del algoritmo de aprendizaje y computacionalmente eficientes.

- **Umbral de varianza** (*Variance threshold / F-test*): remueve atributos con baja varianza, es decir, aquellos mayormente constantes
- **Chi-cuadrado** (*Chi-Square Test*): mide la independencia entre un atributo y la variable target (clase). Solo para atributos categóricos [más]
- **Correlación** (*Correlation Coefficient*): remueve atributos altamente correlacionados, manteniendo solo uno de las involucradas. Puede ser aplicada a atributos numéricos y categóricos.
- **Ganancia de Información** (*Information gain*): cuánto reduce un atributo la incertidumbre sobre la variable que queremos predecir. Puede ser aplicada a atributos numéricos y categóricos. Atributos con valor alto de ganancia impactan en la reducción de incertidumbre sobre la variable target (clase)

Limitaciones

Ignoran potenciales interacciones entre atributos. Además, pueden seleccionar atributos que no mejoran la performance del modelo debido a su independencia del algoritmo de aprendizaje subyacente

Filtros - Spam dataset

LinksCount	SpamWords	FontSize	HourReceived	Clase
3	2	12	8	0
8	9	11	1	1
13	15	12	22	1
7	8	11	23	1
1	1	12	11	0
1	1	12	10	0
0	0	11	11	0
12	12	12	20	1
6	8	12	0	1
8	10	12	0	1
3	2	12	7	0
14	14	11	2	1
11	10	12	0	1
0	0	12	11	0
1	2	11	11	0
2	0	11	10	0
9	10	12	23	1
6	7	11	21	1
8	9	11	20	1
2	2	11	9	0
...	...	...	...	...

Filtros - Spam dataset

LinksCount	SpamWords	FontSize	HourReceived	Clase
3	2	12	8	0
8	9	11	1	1
13	15	12	22	1
7	8	11	23	1
1	1	12	11	0
1	1	12	10	0
0	0	11	11	0
12	12	12	20	1
6	8	12	0	1
8	10	12	0	1
3	2	12	7	0
14	14	11	2	1
11	10	12	0	1
0	0	12	11	0
1	2	11	11	0
2	0	11	10	0
9	10	12	23	1
6	7	11	21	1
8	9	11	20	1
2	2	11	9	0
...	...	...	...	...

- ***F-test*** eliminaría `FontSize` (valores "casi" constantes). `SpamWords`, `LinksCount` y `HourReceived` con valores altos.
- **Correlación** eliminaría `HourReceived` (fuertemente correlacionado con `SpamWords`) ... o podría eliminar `LinksCount` !

Filtros - Spam dataset (cont'd)

LinksCount	SpamWords	FontSize	HourReceived	Clase
3	2	12	8	0
8	9	11	1	1
13	15	12	22	1
7	8	11	23	1
1	1	12	11	0
1	1	12	10	0
0	0	11	11	0
12	12	12	20	1
6	8	12	0	1
8	10	12	0	1
3	2	12	7	0
14	14	11	2	1
11	10	12	0	1
0	0	12	11	0
1	2	11	11	0
2	0	11	10	0
9	10	12	23	1
6	7	11	21	1
8	9	11	20	1
2	2	11	9	0
...	...	...	...	...

- **Chi**^{2} retorna un valor alto `SpamWords` (también para `LinksCount` y `HourReceived`), y casi nulo para `FontSize`
- **Ganacia** retorna valores altos para `SpamWords` y `HourReceived` y casi nulo para `FontSize`. `LinksCount` también alto pero con correlación con `SpamWords`

Selección de atributos (cont'd)

LinksCount	SpamWords	FontSize	HourReceived	Clase
3	2	12	8	0
8	9	11	1	1
13	15	12	22	1
7	8	11	23	1
1	1	12	11	0
1	1	12	10	0
0	0	11	11	0
12	12	12	20	1
6	8	12	0	1
8	10	12	0	1
3	2	12	7	0
14	14	11	2	1
11	10	12	0	1
0	0	12	11	0
1	2	11	11	0
2	0	11	10	0
9	10	12	23	1
6	7	11	21	1
8	9	11	20	1
2	2	11	9	0
...	...	...	...	...

Métodos de Selección - *Wrapper*

Son técnicas que evaluan subconjuntos de atributos de manera iterativa para identificar los más relevantes, según su impacto sobre la performance del modelo. Los atributos con menor impacto son removidos en cada paso del proceso luego que el modelo es nuevamente evaluado.

Métodos de Selección - *Wrapper*

Son técnicas que evaluan subconjuntos de atributos de manera iterativa para identificar los más relevantes, según su impacto sobre la performance del modelo. Los atributos con menor impacto son removidos en cada paso del proceso luego que el modelo es nuevamente evaluado.

- **Eliminación Recursiva con Selección hacia adelante** (*RFE, forward selection*): comienza agregando un atributos en cada iteración

Métodos de Selección - *Wrapper*

Son técnicas que evaluan subconjuntos de atributos de manera iterativa para identificar los más relevantes, según su impacto sobre la performance del modelo. Los atributos con menor impacto son removidos en cada paso del proceso luego que el modelo es nuevamente evaluado.

- **Eliminación Recursiva con Selección hacia adelante** (*RFE, forward selection*): comienza agregando un atributos en cada iteración
- **Eliminación Recursiva con Selección hacia atrás** (*RFE, backward selection*): en caso contrario, remueve la menos significativa en casa paso
- (alternativa) **Eliminación Recursiva con *Floating***: agregar el “mejor” atributo y remover el “peor”

Métodos de Selección - *Wrapper*

Son técnicas que evaluan subconjuntos de atributos de manera iterativa para identificar los más relevantes, según su impacto sobre la performance del modelo. Los atributos con menor impacto son removidos en cada paso del proceso luego que el modelo es nuevamente evaluado.

- **Eliminación Recursiva con Selección hacia adelante** (*RFE, forward selection*): comienza agregando un atributos en cada iteración
- **Eliminación Recursiva con Selección hacia atrás** (*RFE, backward selection*): en caso contrario, remueve la menos significativa en casa paso
- (alternativa) **Eliminación Recursiva con *Floating***: agregar el “mejor” atributo y remover el “peor”
- **Búsqueda estocástica** (*Stochastic search*): mutaciones aleatorias de subconjuntos de atributos

Métodos de Selección - *Wrapper*

Son técnicas que evaluan subconjuntos de atributos de manera iterativa para identificar los más relevantes, según su impacto sobre la performance del modelo. Los atributos con menor impacto son removidos en cada paso del proceso luego que el modelo es nuevamente evaluado.

- **Eliminación Recursiva con Selección hacia adelante** (*RFE, forward selection*): comienza agregando un atributos en cada iteración
- **Eliminación Recursiva con Selección hacia atrás** (*RFE, backward selection*): en caso contrario, remueve la menos significativa en casa paso
- (alternativa) **Eliminación Recursiva con *Floating***: agregar el "mejor" atributo y remover el "peor"
- **Búsqueda estocástica** (*Stochastic search*): mutaciones aleatorias de subconjuntos de atributos

Enfoque híbrido. Por ejemplo, un filtro para reducir el conjunto inicial de atributos, y luego un método *wrapper* para refinar la selección

Limitaciones

RFE puede ser computacionalmente costoso debido a que requiere múltiples iteraciones de entrenamiento y evaluación

Métodos de Selección - *Wrapper*

Son técnicas que evaluan subconjuntos de atributos de manera iterativa para identificar los más relevantes, según su impacto sobre la performance del modelo. Los atributos con menor impacto son removidos en cada paso del proceso luego que el modelo es nuevamente evaluado.

- **Eliminación Recursiva con Selección hacia adelante** (*RFE, forward selection*): comienza agregando un atributos en cada iteración
- **Eliminación Recursiva con Selección hacia atrás** (*RFE, backward selection*): en caso contrario, remueve la menos significativa en casa paso
- (alternativa) **Eliminación Recursiva con *Floating***: agregar el "mejor" atributo y remover el "peor"
- **Búsqueda estocástica** (*Stochastic search*): mutaciones aleatorias de subconjuntos de atributos

Enfoque híbrido. Por ejemplo, un filtro para reducir el conjunto inicial de atributos, y luego un método *wrapper* para refinar la selección

Limitaciones

RFE puede ser computacionalmente costoso debido a que requiere múltiples iteraciones de entrenamiento y evaluación

Ejemplo - Selección hacia adelante

```
# Inicializar lista de features seleccionadas
features_selected = []

# Mientras agregar una feature mejore la metrica del modelo
while mejora_metrica:
    mejor_feature = None
    mejor_score = 0

    # Evaluar cada feature que aun no fue seleccionada
    for feature in features_no_seleccionadas:
        score = evaluar_modelo(features_selected + [feature])
        if score > mejor_score:
            mejor_score = score
            mejor_feature = feature

    # Agregar la mejor feature encontrada
    features_selected.append(mejor_feature)
```

Métodos de Selección - Embebidos

- Integran la selección de atributos en el proceso de entrenamiento, atacando las desventajas de los métodos de filtro y *wrapper*

Métodos de Selección - Embebidos

- Integran la selección de atributos en el proceso de entrenamiento, atacando las desventajas de los métodos de filtro y *wrapper*
- Aseguran que la selección contribuyen a la performance predictiva, mantiendo eficiencia computacional

Métodos de Selección - Embebidos

- Integran la selección de atributos en el proceso de entrenamiento, atacando las desventajas de los métodos de filtro y *wrapper*
- Aseguran que la selección contribuyen a la performance predictiva, mantiendo eficiencia computacional

Ejemplos

- **LASSO** (*Least Absolute Shrinkage and Selection Operator*): técnica de regularización (L1) para regresión lineal

Métodos de Selección - Embebidos

- Integran la selección de atributos en el proceso de entrenamiento, atacando las desventajas de los métodos de filtro y *wrapper*
- Aseguran que la selección contribuyen a la performance predictiva, mantiendo eficiencia computacional

Ejemplos

- **LASSO** (*Least Absolute Shrinkage and Selection Operator*): técnica de regularización (L1) para regresión lineal
- **Métodos basados en árboles** (*Tree-based*): árboles de decisión, random forest, y gradiente

Embebidos - LASSO

- Agrega un término de **penalidad** al error de predicción basado en los valores absolutos de los coeficientes y un parámetro de **regularización** (λ)

Embebidos - LASSO

- Agrega un término de **penalidad** al error de predicción basado en los valores absolutos de los coeficientes y un parámetro de **regularización** (λ)
- Grandes valores de λ incrementan penalidad, *reduciendo la importancia* de algunas características.

Embebidos - LASSO

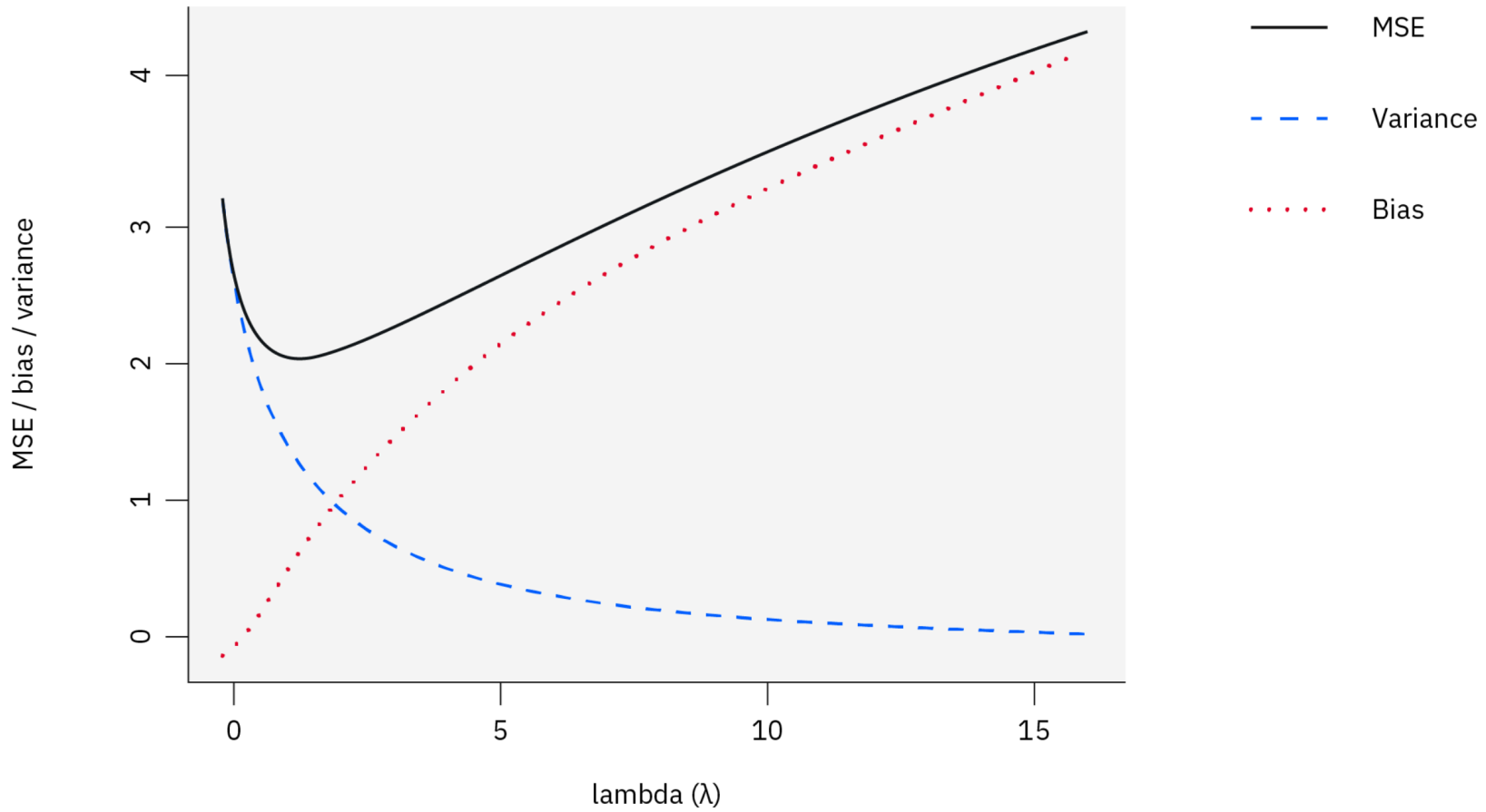
- Agrega un término de **penalidad** al error de predicción basado en los valores absolutos de los coeficientes y un parámetro de **regularización** (λ)
- Grandes valores de λ incrementan penalidad, *reduciendo la importancia* de algunas características.
- Esto resulta en una reducción **automática** de características

Fórmula LASSO (error + penalización)

$$J(w) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |w_j|$$

$$\hat{y}_i = w_0 + \sum_{j=1}^{p} w_j x_{ij}$$

Embebidos - LASSO (cont'd)



(*) imagen de `https://www.ibm.com/think/topics/lasso-regression`

Embebidos - LASSO (cont'd)

Supongamos la función:

$$y = 3x_1 + 0{,}5x_2 + \varepsilon$$

- El algoritmo busca los coeficientes w_1, w_2 que minimicen el error de predicción + la penalización

$$\lambda \sum_i |w_i|$$

Embebidos - LASSO (cont'd)

Supongamos la función:

$$y = 3x_1 + 0{,}5x_2 + \varepsilon$$

- El algoritmo busca los coeficientes w_1, w_2 que minimicen el error de predicción + la penalización

$$\lambda \sum_i |w_i|$$

- Si dos variables son muy correlacionadas entre sí, LASSO suele *apagar* una y dejar la otra.

Embebidos - LASSO (cont'd)

Supongamos la función:

$$y = 3x_1 + 0{,}5x_2 + \varepsilon$$

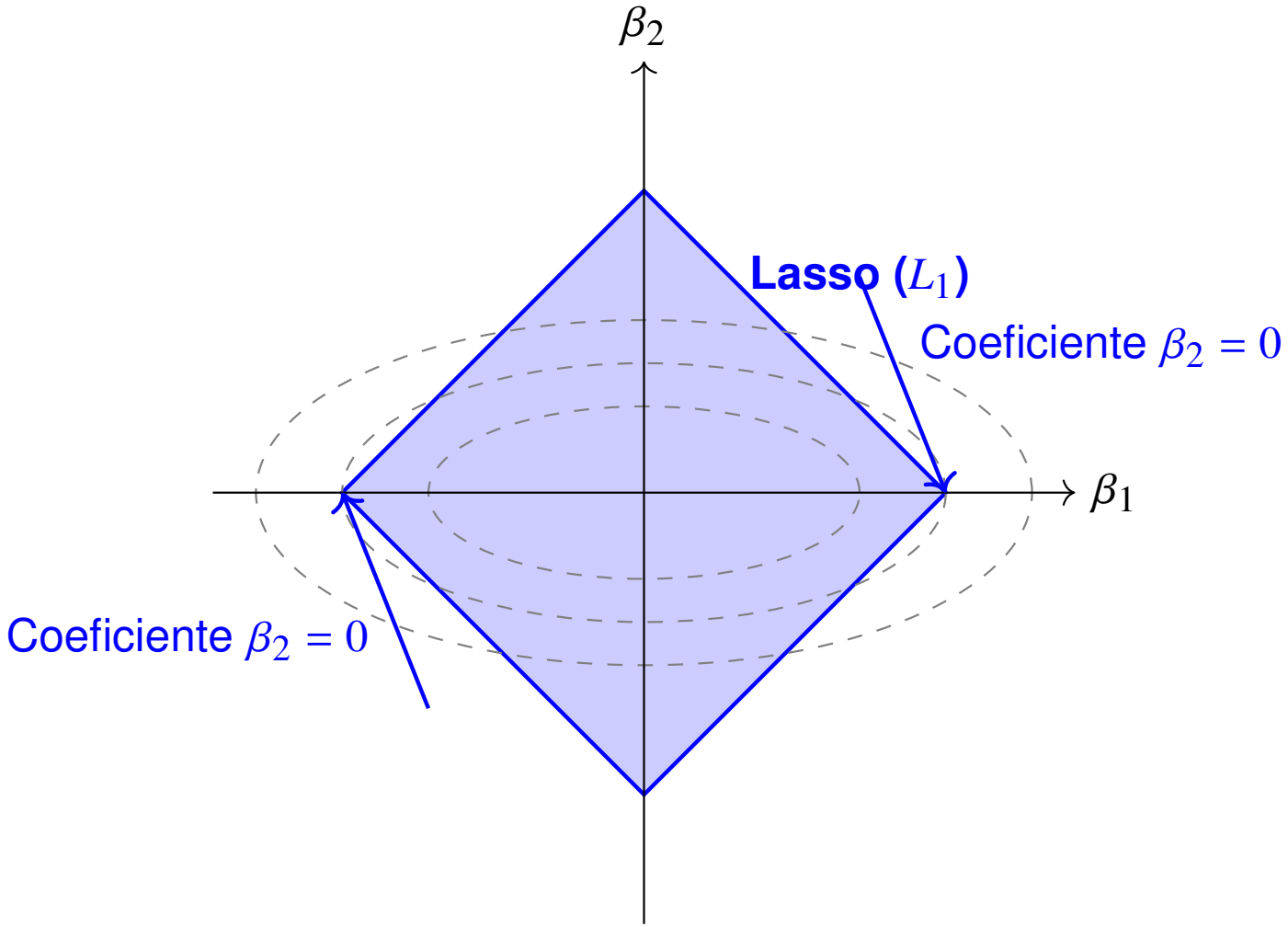
- El algoritmo busca los coeficientes w_1, w_2 que minimicen el error de predicción + la penalización

$$\lambda \sum_i |w_i|$$

- Si dos variables son muy correlacionadas entre sí, LASSO suele *apagar* una y dejar la otra.

$$J(w_1, w_2, b) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (w_1 x_{1,i} + w_2 x_{2,i} + b) \right)^2 + \lambda \left(|w_1| + |w_2| \right)$$

Lasso: Geometría de la penalización



Las esquinas del diamante L_1 favorecen soluciones con coeficientes exactamente cero

Métodos de Selección - Resumen

Table 1. Comparison of Feature Selection Methods in Machine Learning.

Method	Advantages	Limitations	Examples
Filter Methods	Computationally efficient, easy to implement	Ignores feature interactions	Correlation, Chi-square
Wrapper Methods	Considers feature interactions	High computational cost	RFE, Forward Selection
Embedded Methods	Integrated into model training	Model-dependent	LASSO, Tree-based methods

Figura 1.2: Table de [1]

Análisis de componentes principales

Motivación e Intuición

- El análisis de componentes principales (**PCA**, en inglés) es una técnica matemática para reducir la complejidad de los datos, i.e. la **dimensionalidad**.

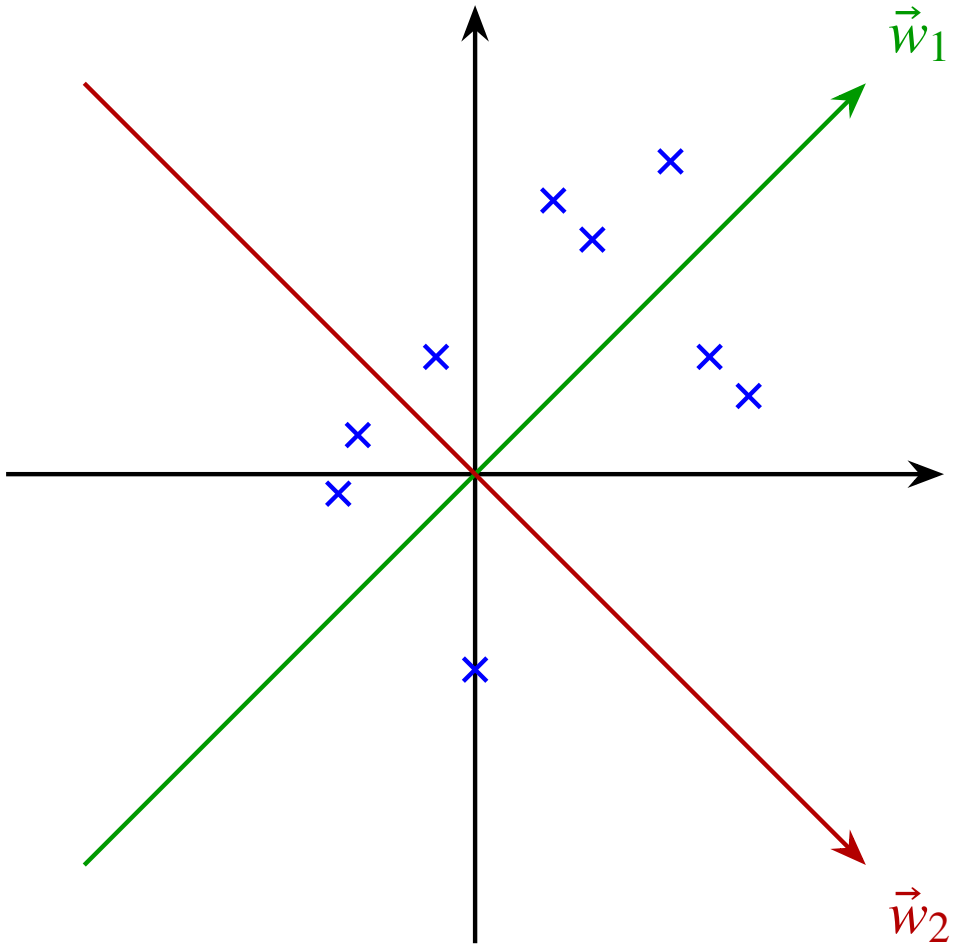
Motivación e Intuición

- El análisis de componentes principales (**PCA**, en inglés) es una técnica matemática para reducir la complejidad de los datos, i.e. la **dimensionalidad**.
- El objetivo es detectar combinaciones lineales que puedan capturar la *mayor* **varianza** en el dataset inicial completo.

Motivación e Intuición

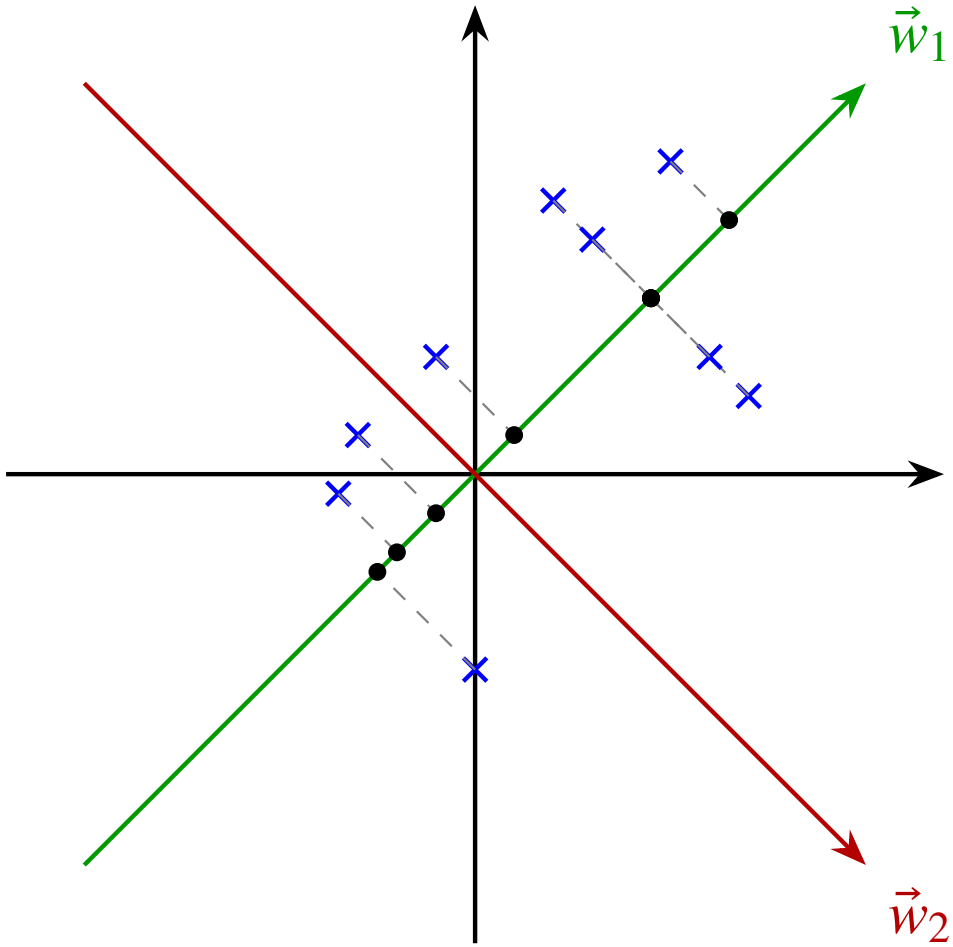
- El análisis de componentes principales (**PCA**, en inglés) es una técnica matemática para reducir la complejidad de los datos, i.e. la **dimensionalidad**.
- El objetivo es detectar combinaciones lineales que puedan capturar la *mayor* **varianza** en el dataset inicial completo.
- Los componentes principales son ortogonales y no están correlacionados entre ellos.

Motivación e Intuición (cont'd)



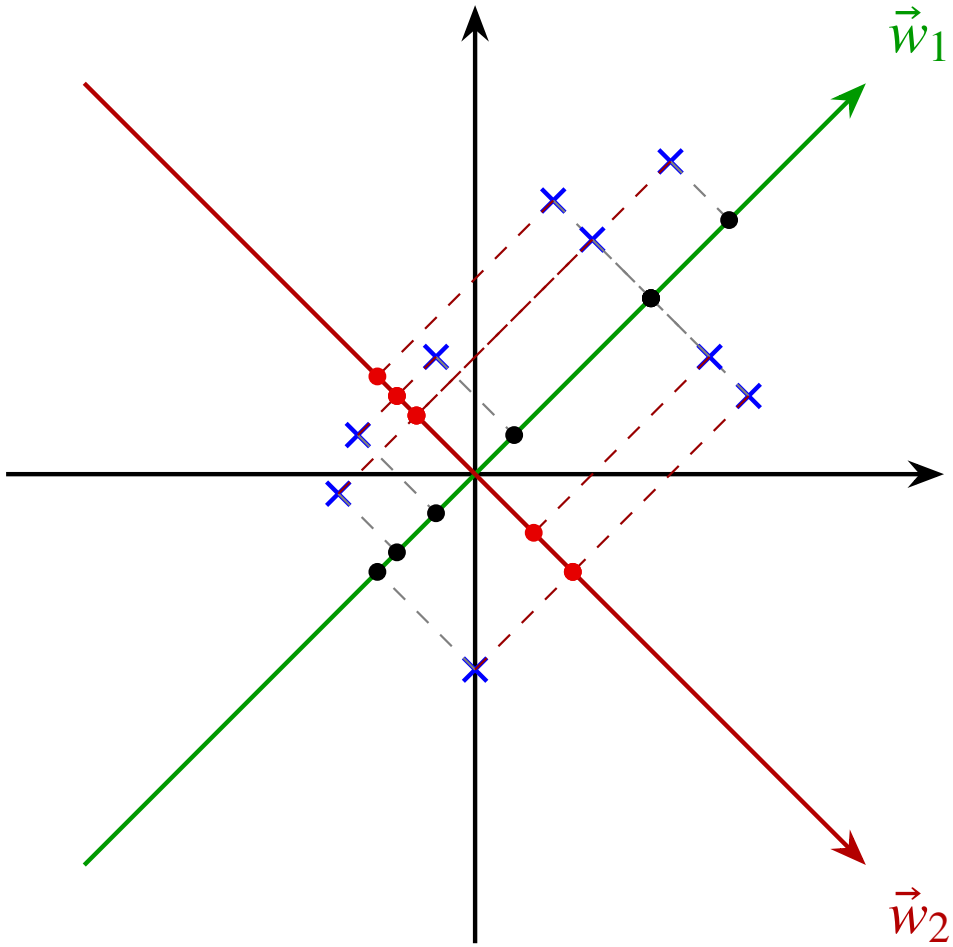
(*) Intuición adaptada de Andrew Ng

Motivación e Intuición (cont'd)



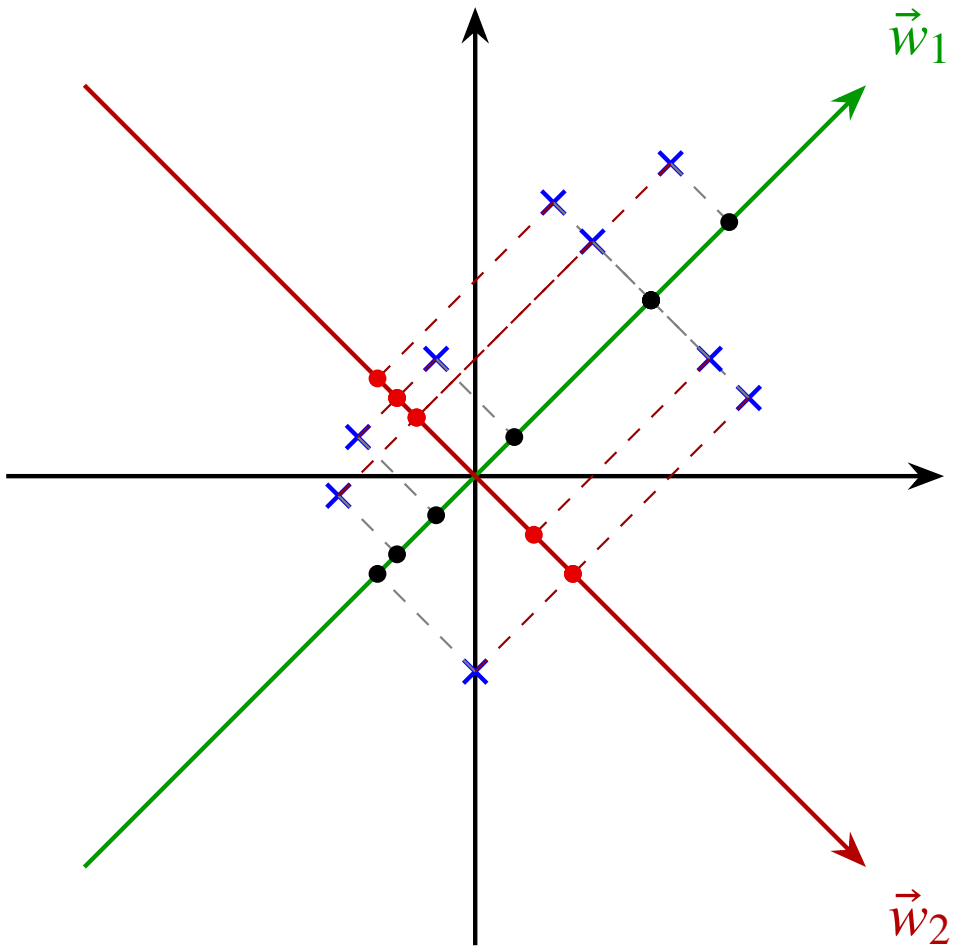
(*) Intuición adaptada de Andrew Ng

Motivación e Intuición (cont'd)



(*) Intuición adaptada de Andrew Ng

Motivación e Intuición (cont'd)

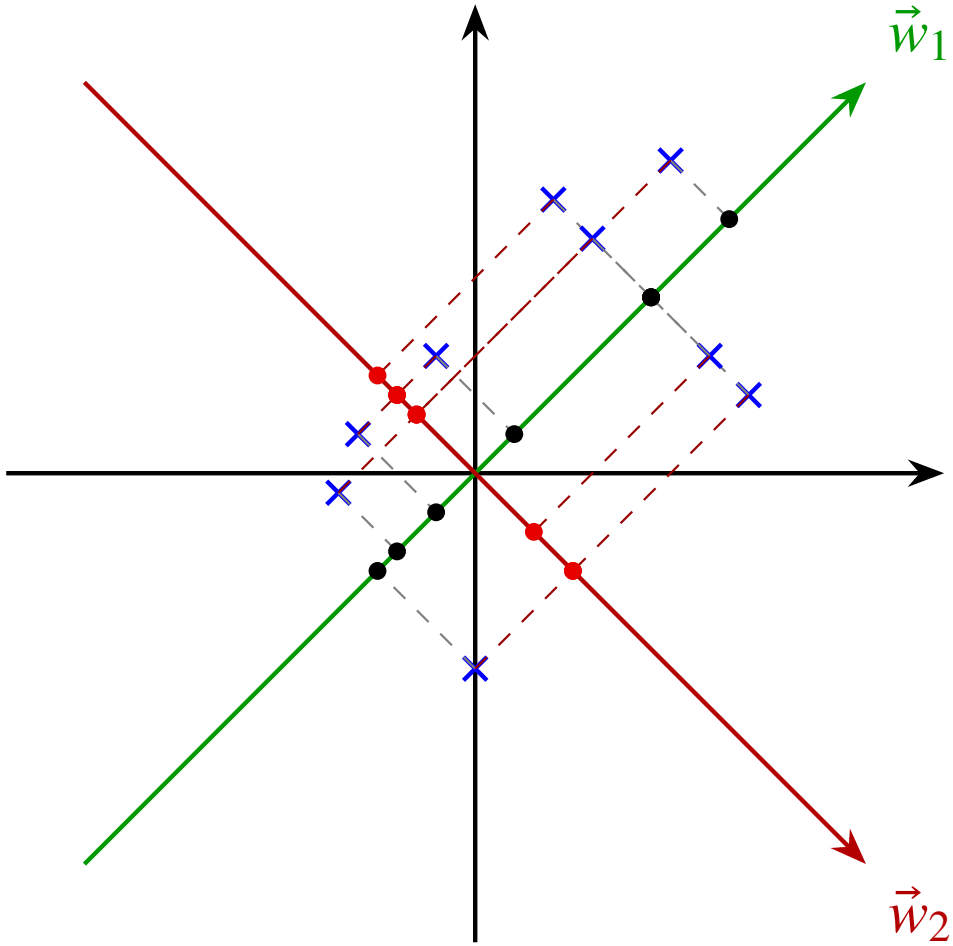


$\vec{w}_1$ *Buen* candidato

$\vec{w}_2$ *Mal* candidato

(*) Intuición adaptada de Andrew Ng

Motivación e Intuición (cont'd)



- *Buen* candidato → maximiza varianza, i.e. puntos sobre $\vec{w}_1$ están a mayor distancia entre si
- *Mal* candidato → menor varianza, i.e. puntos sobre $\vec{w}_2$ están a menor distancia entre si

Motivación e Intuición (cont'd)

Varianza e Información

Mientras mayor es la dispersión de los datos, más información tienen.

PCA - Enfoque

1. Estandarizar los datos y transformarlos a una escala comparable

PCA - Enfoque

1. Estandarizar los datos y transformarlos a una escala comparable
2. Computar la matriz de covarianza

PCA - Enfoque

1. Estandarizar los datos y transformarlos a una escala comparable
2. Computar la matriz de covarianza
3. Encontrar los *eigenvectors* y sus *eigenvalues*

PCA - Enfoque

1. Estandarizar los datos y transformarlos a una escala comparable
2. Computar la matriz de covarianza
3. Encontrar los *eigenvectors* y sus *eigenvalues*
4. Seleccionar y crear el vector de características

PCA - Enfoque

1. Estandarizar los datos y transformarlos a una escala comparable
2. Computar la matriz de covarianza
3. Encontrar los *eigenvectors* y sus *eigenvalues*
4. Seleccionar y crear el vector de características
5. Transformar el dataset original proyectando en los PCA seleccionados.

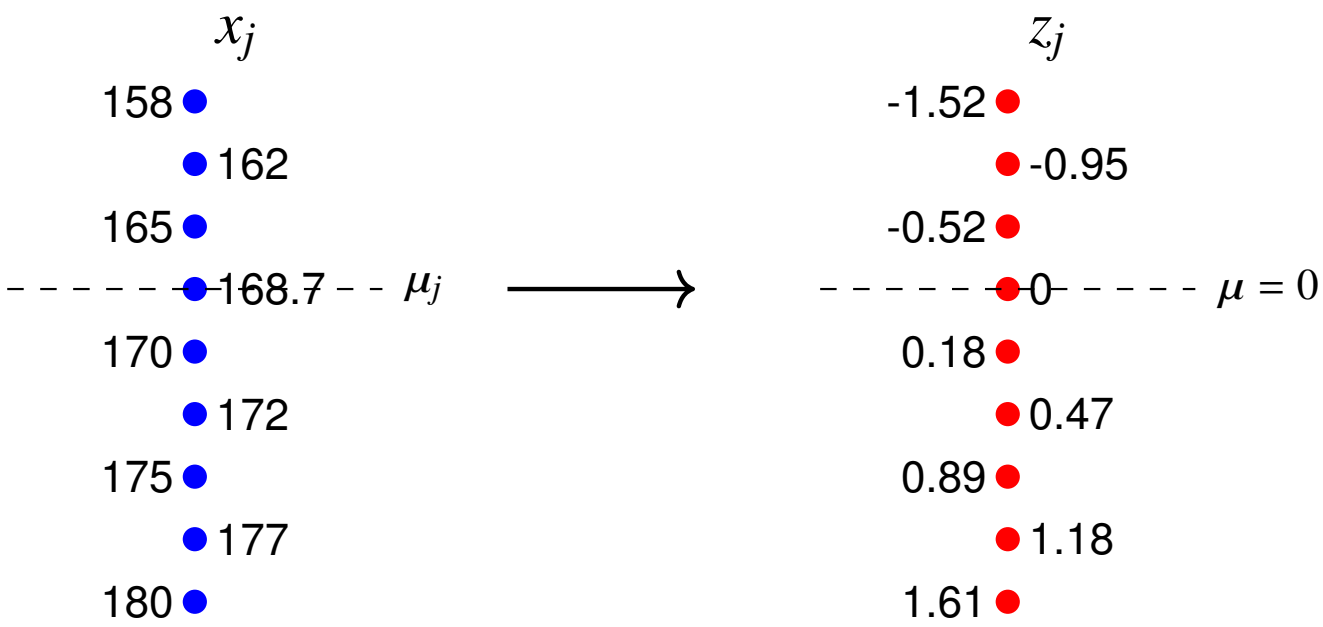
Estandarización

Estandarizar y transformar a una escala comparable

$$z_j^{(i)} = \frac{x_j^{(i)} - \bar{\mu}_j}{\sigma_j}$$

- $x_j^{(i)}$: valor de la observación i en la variable j
- $\bar{\mu}_j = \frac{1}{n} \sum_{i=1}^{n} x_j^{(i)}$: media de la variable j, n son las observaciones
- $\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_j^{(i)} - \bar{\mu}_j)^2}$: desvío estándar poblacional, i.e. dispersión respecto de μ

Estandarización - Ejemplo



$$z_j^{(i)} = \frac{x_j^{(i)} - \bar{\mu}_j}{\sigma} = \frac{158 - 168{,}7}{7{,}06} = \frac{-10{,}7}{7{,}06} = -1{,}52$$

(*) Ejemplo adaptado de

Covarianza

Matriz de Covarianza (cov)

- Permite entender la **correlación** entre variables, es decir, **cuán redundantes son**.
  - si la $cov(x, y)$ es +, ambas x e y crecen o decresen juntas (correlación)
  - si la $cov(x, y)$ es $-$, x crece e y decrece, o viceversa
- Es una matriz (**simétrica**) de $n\ x\ n$, dónde n es el # de dimensiones

Covarianza

Matriz de Covarianza (cov)

- Permite entender la **correlación** entre variables, es decir, **cuán redundantes son**.
  - si la $cov(x, y)$ es +, ambas x e y crecen o decresen juntas (correlación)
  - si la $cov(x, y)$ es $-$, x crece e y decrece, o viceversa
- Es una matriz (**simétrica**) de n x n, dónde n es el # de dimensiones

$$\text{Cov}(X) = \begin{bmatrix} \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) & \text{cov}(x_1, x_3) \\ \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) & \text{cov}(x_2, x_3) \\ \text{cov}(x_3, x_1) & \text{cov}(x_3, x_2) & \text{cov}(x_3, x_3) \end{bmatrix}$$

- $\text{cov}(x_i, x_j)$ = covarianza entre las variables x_i y x_j
- Diagonal = varianzas de cada variable: $\text{cov}(x_i, x_i) = \text{var}(x_i)$
- Matriz simétrica: $\text{cov}(x_i, x_j) = \text{cov}(x_j, x_i)$

Covarianza - Ejemplo

$$\text{cov}(X) = \begin{bmatrix} \text{cov}(x_1, x_1) & \text{cov}(x_1, x_2) & \text{cov}(x_1, x_3) \\ \text{cov}(x_2, x_1) & \text{cov}(x_2, x_2) & \text{cov}(x_2, x_3) \\ \text{cov}(x_3, x_1) & \text{cov}(x_3, x_2) & \text{cov}(x_3, x_3) \end{bmatrix}$$

H (cm)	W (kg)	A (años)	G
170	65	30	1
165	59	25	0
180	75	35	1
175	68	28	1
160	55	22	0
172	70	32	1
168	62	27	0
177	74	33	1
162	58	24	0
158	54	21	0

Sean $x1 = H$ y $x2 = W$. $\bar{\mu}_1 = 168{,}7$ y $\bar{\mu}_2 = 64{,}0$:

$$\text{cov}(x_1, x_2) = \frac{1}{n} \sum_{i=1}^{n} (x_1^{(i)} - \bar{\mu}_1)(x_2^{(i)} - \bar{\mu}_2)$$

$$(170 - 168{,}7)(65 - 64) = 1{,}3 \cdot 1 = 1{,}3$$
$$(165 - 168{,}7)(59 - 64) = (-3{,}7) \cdot (-5) = 18{,}5$$
$$(180 - 168{,}7)(75 - 64) = 11{,}3 \cdot 11 = 124{,}3$$
$$\vdots$$

Sumando y dividiendo entre $n = 10$:

$$\text{cov}(x_1, x_2) \approx 14{,}88$$

Eigenvectors

Eigenvectors y *Eigenvalues*

- *Eigenvectors* de una matriz de covarianza representan las direcciones de máxima varianza en los datos

Eigenvectors

Eigenvectors y *Eigenvalues*

- *Eigenvectors* de una matriz de covarianza representan las direcciones de máxima varianza en los datos
- *Eigenvalues* son coeficientes que dan la cantidad de varianza (importancia del *eigenvector*)

Eigenvectors

Eigenvectors y *Eigenvalues*

- *Eigenvectors* de una matriz de covarianza representan las direcciones de máxima varianza en los datos
- *Eigenvalues* son coeficientes que dan la cantidad de varianza (importancia del *eigenvector*)
- Formalmente:

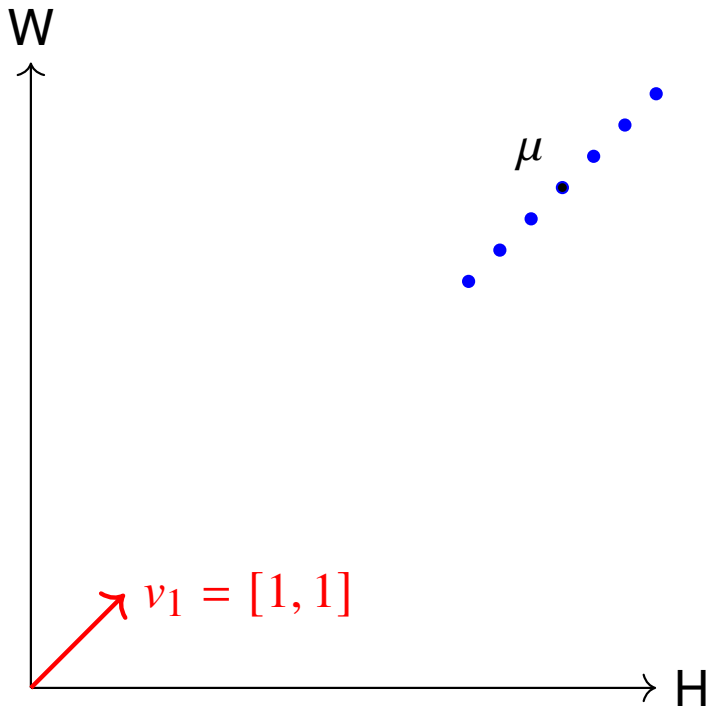
$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad \mathbf{v} \neq 0$$

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = 0$$

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \quad \text{para encontrar } \lambda$$

  - I = matriz identidad $n \times n$
  - λ = eigenvalor
  - v = eigenvector no nulo asociado a λ

Eigenvectors - dirección de máxima varianza



- Los puntos azules representan observaciones en Height (cm) y Weight (kg).
- El vector v_1 muestra la dirección del eigenvector asociado al mayor eigenvalor.
- Proyectando los puntos sobre esta línea obtendremos el primera componente principal (PC1).

Eigenvectors - Ejemplo

Matriz de covarianza simplificada:

$$C = \begin{bmatrix} 1{,}0 & 0{,}98 \\ 0{,}98 & 1{,}0 \end{bmatrix}$$

Paso 1: Eigenvalores λ

$$\det(C - \lambda I) = \begin{vmatrix} 1-\lambda & 0{,}98 \\ 0{,}98 & 1-\lambda \end{vmatrix} = (1-\lambda)^2 - 0{,}98^2 = 0$$

$$\lambda_1 = 1 + 0{,}98 = 1{,}98, \quad \lambda_2 = 1 - 0{,}98 = 0{,}02$$

Paso 2: Eigenvectores v

Para $\lambda_1 = 1{,}98$:

$$(C - \lambda_1 I)v_1 = \begin{bmatrix} -0{,}98 & 0{,}98 \\ 0{,}98 & -0{,}98 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$-0{,}98v_1 + 0{,}98v_2 = 0 \implies v_1 = v_2$$

normalizado: $v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \approx [0{,}707, 0{,}707]$

Para $\lambda_2 = 0{,}02$:

$$(C - \lambda_2 I)v_2 = \begin{bmatrix} 0{,}98 & 0{,}98 \\ 0{,}98 & 0{,}98 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

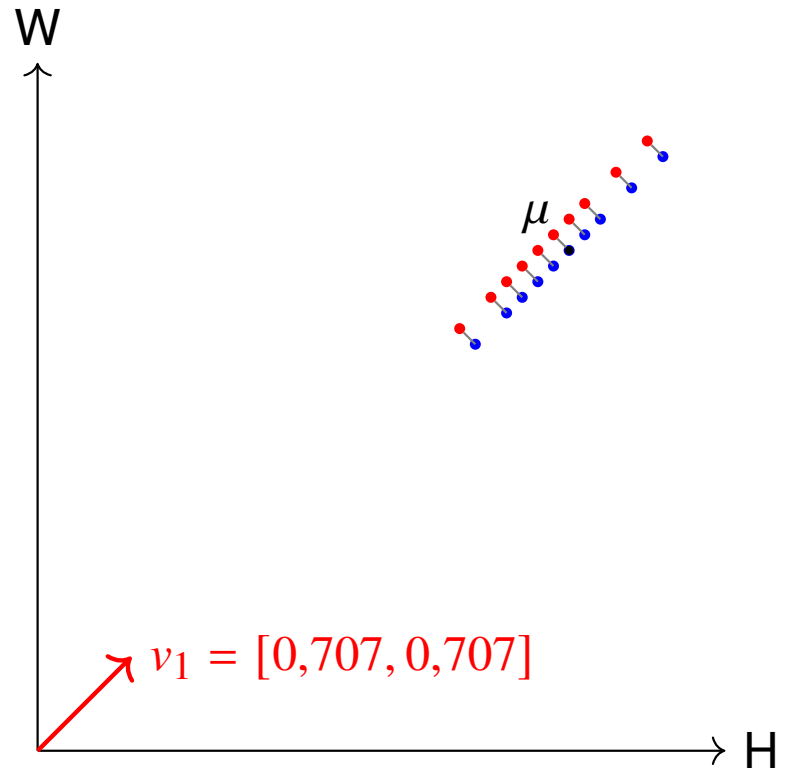
$$0{,}98v_1 + 0{,}98v_2 = 0 \implies v_1 = -v_2$$

normalizado: $v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \approx [0{,}707, -0{,}707]$

Resultado:

Eigenvalor	Eigenvector (normalizado)
1.98	[0.707, 0.707]
0.02	[0.707, -0.707]

Eigenvector y proyección de datos



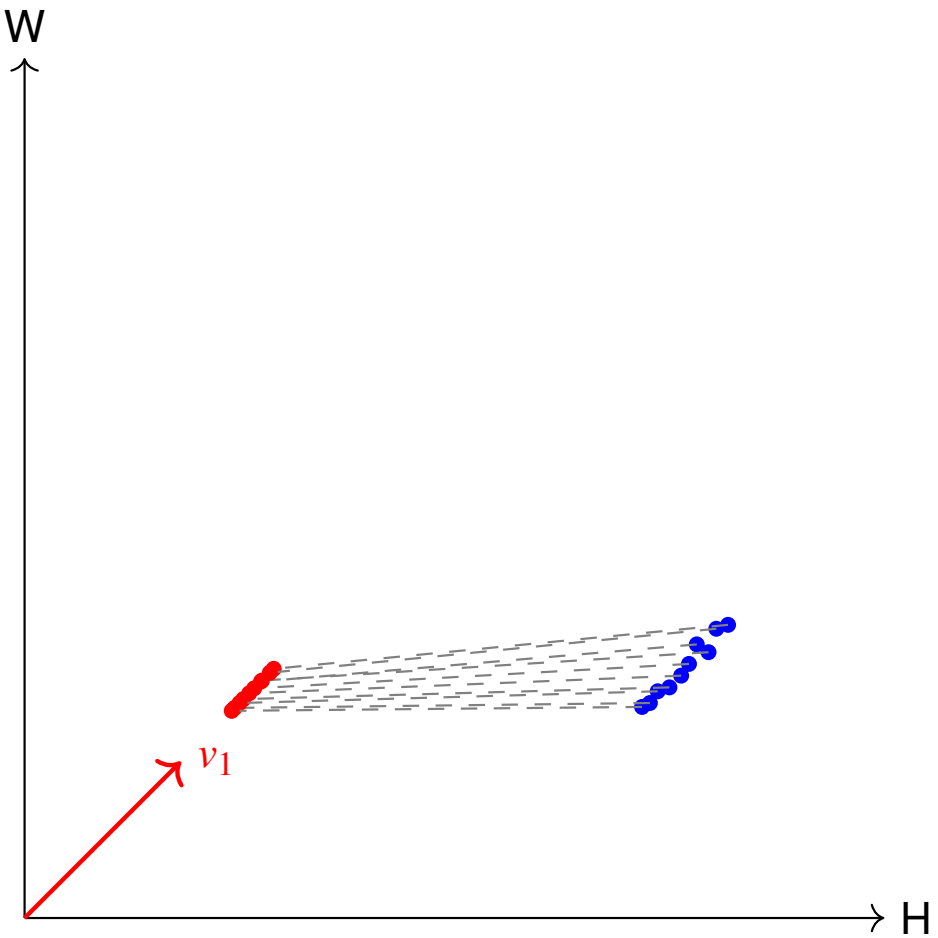
- Puntos azules: observaciones (Height vs Weight)
- v_1: eigenvector normalizado, dirección de máxima varianza (PC1)
- Puntos rojos: proyección de cada observación sobre PC1
- Líneas punteadas grises: muestran cómo se proyecta cada punto

Eigenvector y proyección de datos - Ejemplo

Eigenvector principal:

$$v_1 = \frac{1}{\sqrt{2}}[1,1] \approx [0{,}707, 0{,}707]$$

Height	Weight	PC1 (aprox)
170	65	1.63
165	59	-6.15
180	75	15.77
175	68	7.07
160	55	-12.27
172	70	6.36
168	62	-1.97
177	74	12.62
162	58	-8.49
158	54	-14.85



Consideraciones finales

- La selección de los PCs se realiza a partir del conjunto de *eigenvectors* ordeanos en **orden decreciente** de *eigenvalues* (λ)
  - mayor λ capturan la mayor variación de datos

Consideraciones finales

- La selección de los PCs se realiza a partir del conjunto de *eigenvectors* ordeanos en **orden decreciente** de *eigenvalues* (λ)
  - mayor λ capturan la mayor variación de datos
- Los *eigenvectors* son **perpendiculares** entre si
  - cada nuevo componente explica nueva información no redundante!

Consideraciones finales

- La selección de los PCs se realiza a partir del conjunto de *eigenvectors* ordeanos en **orden decreciente** de *eigenvalues* (λ)
  - mayor λ capturan la mayor variación de datos
- Los *eigenvectors* son **perpendiculares** entre si
  - cada nuevo componente explica nueva información no redundante!
- *criterio*: elegir los primeros k *eigenvectors* que expliquen un % de la varianza acumulada

$$\text{Varianza acumulada}(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{p} \lambda_j} \times 100\,\%$$

PCA en la práctica

```
data = {
    'Height': [170, 165, 180, 175],
    'Weight': [65, 59, 75, 68],
    'Age': [30, 25, 35, 28],
    'Gender': [1, 0, 1, 1]
}
df = pd.DataFrame(data)

X = df.drop(['Gender','Age'], axis=1)

scaler = StandardScaler()
df_scaled_array = scaler.fit_transform(X)

pca = PCA(n_components=2)
X_pca = pca.fit_transform(df_scaled_array)
```

Próximas clases

- Regresión
- Máquinas de soporte vectorial
- Redes Neuronales
- Aprendizaje no supervisado

¡Gracias!

Bibliografía y material de referencia


Harrington, Peter. Machine learning in action. *Simon and Schuster*, 2012.


Alpaydin, Ethem. Introduction to machine learning. 3era Edición *MIT Press*, 2020.


Brett Lantz. Machine Learning with R. *Packt Publishing*, 1997.


Tom M. Mitchell. Machine Learning. *WCB McGraw-Hill*, 1997.


Witten I., Frank E., Hall, M., Pal C.. Data Mining: Practical Machine Learning Tools and Techniques. 4th Edition*WMorgan Kaufmann. Elsevier*, 2017.


Michael A. Nielsen. Neural Networks and Deep Learning. 4th Edition *Determination Press*, 2015.
`http://neuralnetworksanddeeplearning.com`

Bibliografía y material de referencia


Afshine Amidi, Shervine Amidi. CS 229 — Machine Learning.
https://stanford.edu/~shervine/teaching/cs-229/


Andrew Ng. Stanford CS229 - Machine Learning Course.
https://www.youtube.com/playlist?list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU


Andrew Ng. Deep Learning AI.
https://www.deeplearning.ai/resources/


Cheng, Xueyi. A Comprehensive Study of Feature Selection Techniques in Machine Learning Models. SSRN Electronic Journal. 2024


Jundong Li, Kai Cheng, Suhang Wang, Fred Morstatter, Ryan P. Trevino, Jiliang Tang, and Huan Liu.
Feature Selection: A Data Perspective.
Chapman and Hall/CRC, 2017.