Modelos de aprendizaje automático basados en Análisis Formal de Conceptos

Adam Lei Yi Chen Abolacio

Dirigido por D. Joaquín Borrego Díaz

Introducción

- Crear modelos de aprendizaje automático para clasificación basados en FCA.
- Análisis de base STEM, Luxenburger, contextos formales y retículo de conceptos.
- Objetivo: modelos explicables.

Estructura de la presentación

- Preliminares.
- Base STEM.
- Base Luxenburger.
- Limitaciones FCA.
- Modelo fijo FCA.
- Modelo incremental FCA.
- Extracción del conocimiento de modelos.
- Discretización de variables.
- Conclusiones.

Contextos formales

- En FCA representaremos todos los conjuntos de datos como contextos formales.
- En esencia $C = (O, A, I)$

	Fluvial	Litoral	Océano
Carpa	X		
Escatófagus	X	X	
Sargo		X	X
Dorada		X	X
Anguila	X	X	X

Contextos formales: definiciones adicionales

- Definimos como $S \subseteq A$ al **conjunto de atributos de clasificación**.
- **Conjunto de atributos de entrada**: cualquier conjunto $B \subseteq A$ tal que $B \cap S = \emptyset$.
- Podemos representar cualquier objeto como un conjunto de atributos de entrada
- Objetivo: dado un conjunto de entrada el modelo devuelve un conjunto $Y \subseteq S$.

Derivación en FCA

- Dado un contexto formal $C = (O, A, I)$:
  - O: objetos, A: atributos, $I \subseteq O \times A$
- Operaciones de **derivación**:
  - Para $X \subseteq O$:

$$X' = \{a \in A \mid \forall o \in X,\ (o, a) \in I\}$$

    (atributos comunes a los objetos de X)
  - Para $Y \subseteq A$:

$$Y' = \{o \in O \mid \forall a \in Y,\ (o, a) \in I\}$$

    (objetos que poseen todos los atributos de Y)
- **Doble derivación**: $X'' = (X')'$, $Y'' = (Y')'$ *(operador de clausura)*

Soporte y confianza de implicaciones

Sea $L \equiv P \rightarrow Q$ una implicación.

Soporte

$$supp(L) = \frac{|(P \cup Q)'|}{|O|}$$

Confianza

$$conf(L) = \frac{|(P \cup Q)'|}{|P'|}$$

Base STEM

Definición

- Base Duquenne-Guigues o base canónica.
- Conjunto de implicaciones tipo $P \rightarrow Q$.

Propiedades clave

- Completitud
- Irreducibilidad
- Conjunto más pequeño posible en número de implicaciones.

Cálculo

- Next-Closure

Base STEM como clasificador

- Base STEM como sistema de disparo de reglas.
- Conjunto S de atributos de clasificación.
- Implicaciones en forma de cláusulas de Horn.
- Los objetos son representados como conjuntos de hechos Y (conjunto de entrada).
- A partir de un conjunto de hechos inicial, llegar a un nuevo conjunto mediante disparo de reglas que contenga algún atributo de clasificación.
- Complejidad del algoritmo $\theta(n^2)$: cuadrática respecto del número de implicaciones.
- Sistema rápido y compacto.

Limitaciones base STEM: ambiguedad

Si existe **ambiguedad** con Y en el contexto formal, la base STEM nunca podrá deducir la clasificación de una entrada Y.

Hay **ambiguedad** en un contexto formal si dados dos objetos $N, M \in O$ comparten los mismos atributos de entrada pero diferente atributo de clasificación:

$$\exists N, M \in O[(N' \setminus S = M' \setminus S) \wedge (N' \cap S \neq M' \cap S)]$$

No se puede generar una regla lógica válida que permite clasificar N y M en este contexto formal.

Limitaciones base STEM: generalización

Hipótesis del mundo cerrado (CWA): la base STEM no puede clasificar ejemplos que no estén representados en el contexto formal de la que se extrajo.

Base STEM con implicaciones lógicamente válidas pero sin **soporte**

Dado un conjunto de entrada X: $X' = \emptyset \rightarrow X'' = A$

Ejemplo: $A = \{\text{par}, \text{impar}, \text{compuesto}, \text{primo}\}$

$$\text{par} \wedge \text{impar} \rightarrow A$$

Conclusión: base STEM sólo puede clasificar ejemplos vistos anteriormente sin ambiguedad.

Bases de Luxenburger: definición

- Conjunto compacto y no redundante de reglas de asociación ($P \rightarrow Q$) válidas.
- Una regla de asociación es válida si está por encima de un umbral de soporte $\delta \in [0, 1]$ y confianza $\gamma \in [0, 1]$.
- Base STEM extendida $\gamma = 1$.
- Una base de Luxenburger por cada valor de soporte y confianza.

Bases de Luxenburger: ambiguedad

- Ajustando la confianza γ, podemos clasificar ejemplos ambiguos.
- Idea: dado un conjunto de entrada X (si es ambiguo), ajustar la confianza γ descendiendo paulatinamente con una tasa de descenso hasta que $\mathcal{L}(C, \delta, \gamma)$ pueda clasificar X.

Bases de Luxenburger: inestabilidad

- Al descender la confianza se observa que la salida de $\mathcal{L}(C, \delta, \gamma)$ se vuelve inestable al variar levemente γ.
- No podemos asegurar que no aparezcan reglas que nos lleven a conclusiones incoherentes.

Bases de Luxenburger: conclusiones

- Debido a la inestabilidad, las bases de Luxenburger no resuelven la clasificación con datos ambiguos.
- Las bases de Luxenburger no tienen la capacidad de **generalización** (CWA).

Sistema de nomenclatura derivado de FCA

$X \rightarrow$ Identificación
- Visto anteriormente
  - No ambiguo
    - STEM
    - Luxenburger
  - Ambiguo
- No visto

Caso no ambiguo

Retículo de conceptos

Diremos que Y es no ambiguo respecto de C si el mayor concepto que contiene a Y, que es: (Y', Y''), contiene exactamente un único atributo de clasificación:

$$|Y'' \cap S| = 1$$

Clasificación

La clasificación de Y es $Y'' \cap S$.

Caso ambiguo

Retículo de conceptos

Diremos que Y es ambiguo respecto de C si: el mayor concepto que contiene a Y, que es (Y', Y''), no contiene ningún atributo de clasificación.

$$Y'' \cap S = \emptyset$$

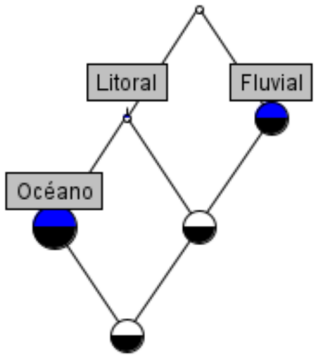
Caso no visto

Retículo de conceptos

Diremos que Y es un ejemplo no visto respecto de C si: el mayor concepto que contiene a Y, que es: (Y', Y'') contiene todos los atributos de clasificación $Y'' = A$:

$$Y'' \cap S = S$$

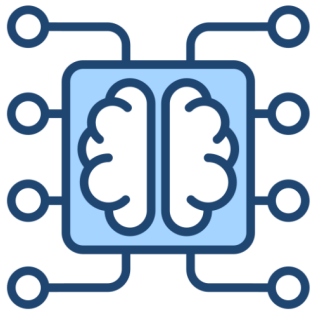
Modelo fijo: idea



+

	Fluvial	Litoral	Océano
Carpa	X		
Escatófagus	X	X	
Sargo		X	X
Dorada		X	X
Anguila	X	X	X

=



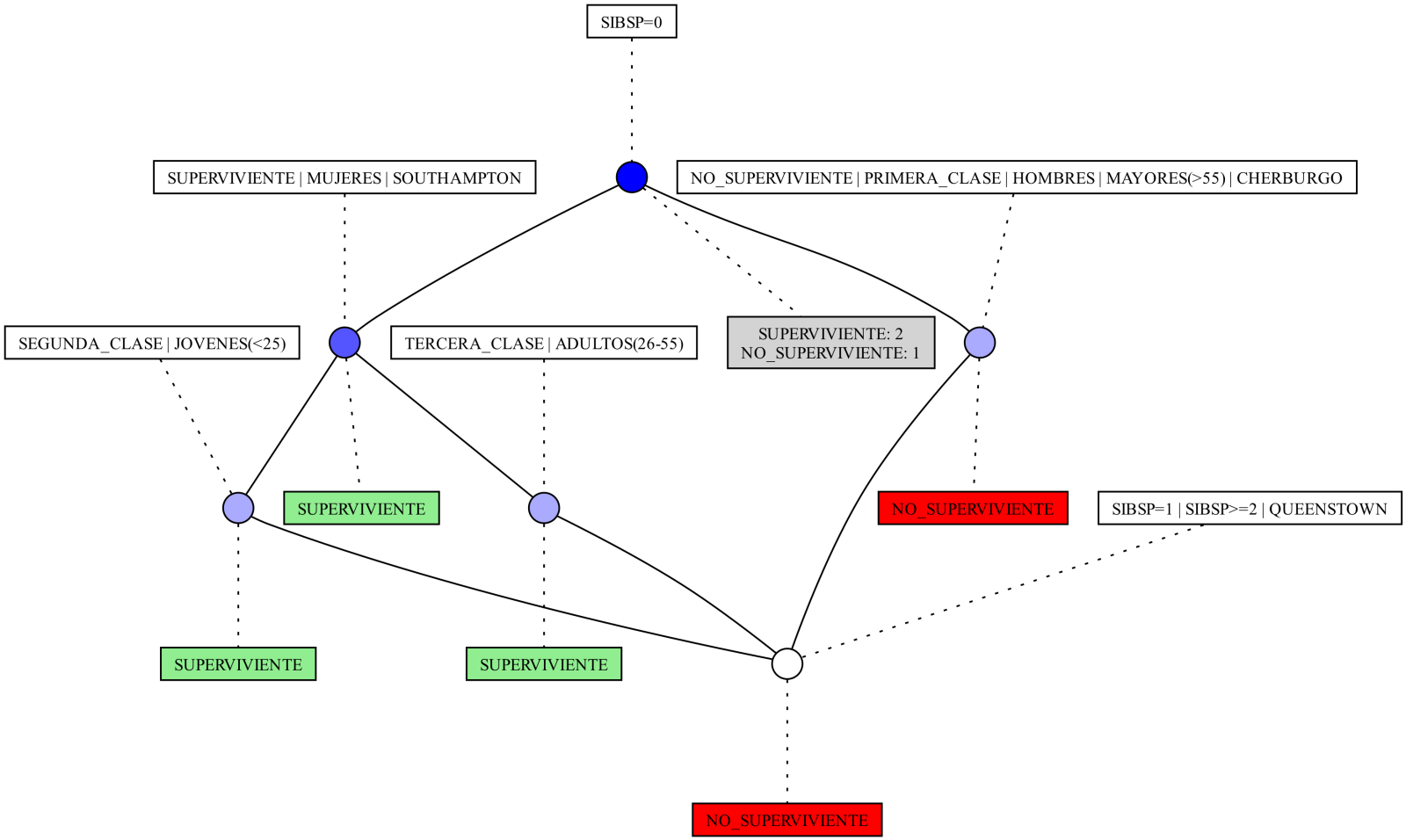
Caso ambiguo

Dado un conjunto de entrada X, calcular el mayor concepto que contiene a X: (X', X'').

Si X es **ambiguo** respecto del contexto formal C, entonces: calcular la distribución de probabilidad de X' en función del conjunto de atributos de clasificación S.

Se proponen dos modos de devolver la clasificación final: **modo determinista** y **modo probabilístico**.

Caso ambiguo: idea gráfica

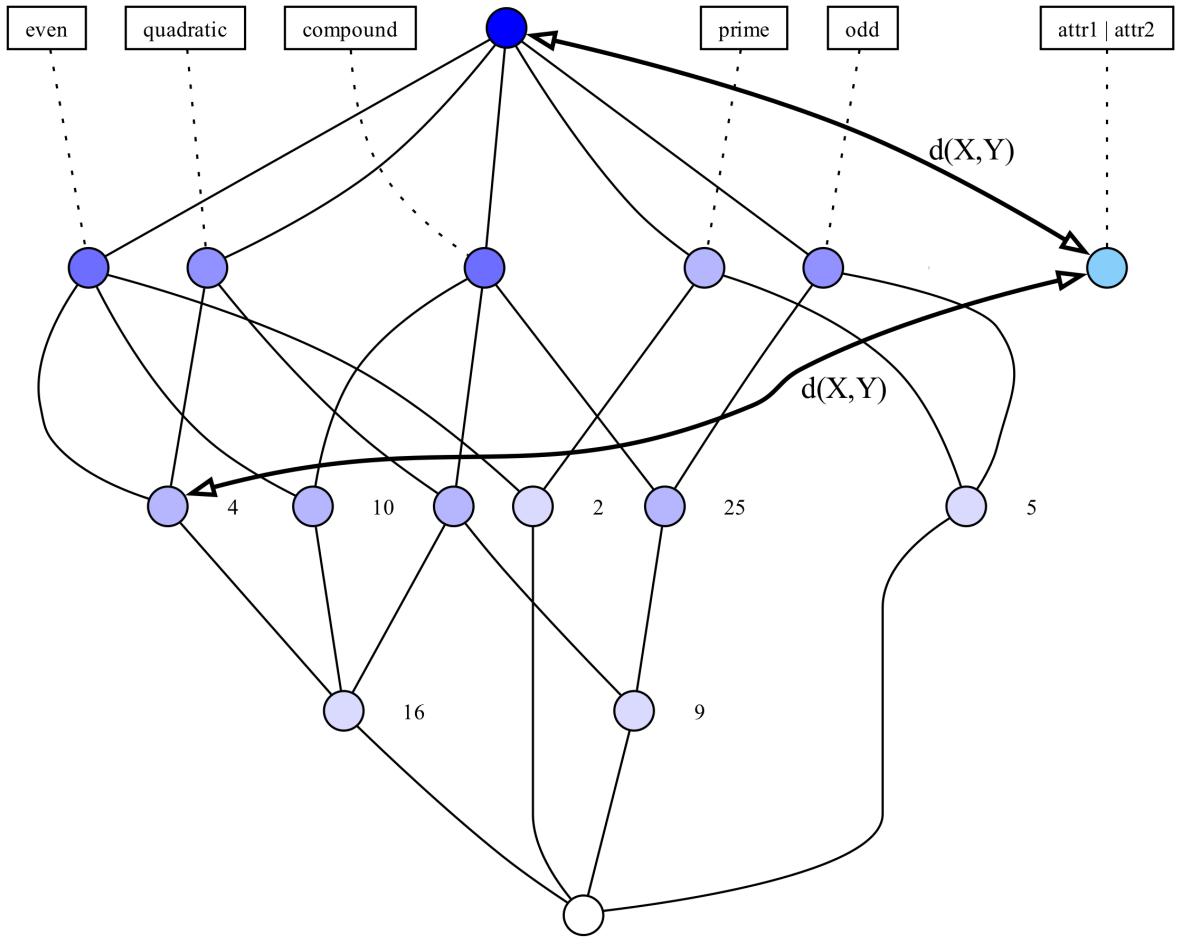


Caso no visto

Dado un conjunto de entrada X y una función de similitud $d : A \times A \to \mathbb{R} \in [0, 1]$, calcular el mayor concepto que contiene a $X : (X', X'')$.

Si X es un caso no visto respecto del contexto formal C, entonces: calcular mediante d el **concepto más similar** entre X y todos los conceptos formales de C.

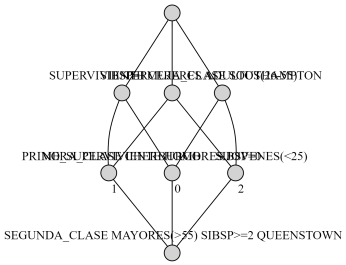
Caso no visto: idea gráfica



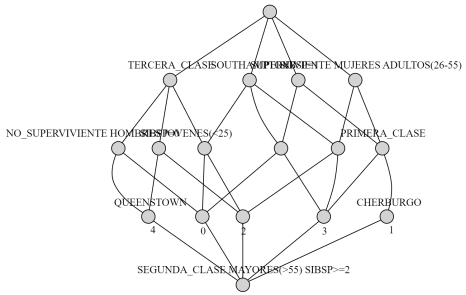
Modelo fijo: conclusiones

- Modelo con capacidad de generalización.
- Explicable.
- Sobreajuste al contexto formal de entrenamiento.
- Cálculo del retículo con complejidad exponencial.

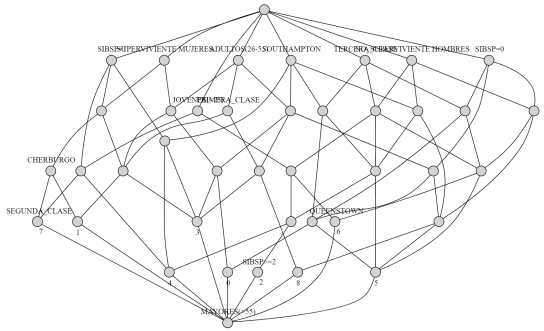
Modelo incremental: idea



Epoch 1



Epoch 2



Epoch 3

Modelo incremental

- El modelo empieza con su contexto formal $C = (O, A, I)$ con el conjunto de objetos vacío $O = \emptyset$.
- En cada época se genera un nuevo contexto formal C_i modificando el contexto anterior C_{i-1}.
- Entrenamiento por épocas con tasa de aprendizaje η y contexto formal de validación.
- Hipótesis: cuanto más se aproxime el retículo del modelo al retículo real mejor rendimiento tendrá.

Modelo incremental: entrenamiento

$$X \begin{cases} \text{No visto} \rightarrow \text{añadir } X \text{ a } C. \\ \text{No ambiguo} \begin{cases} \text{Acierto} \rightarrow \text{no hacer nada.} \\ \text{Fallo} \rightarrow \text{añadir } X \text{ a } C. \end{cases} \\ \text{Ambiguo} \begin{cases} \text{Acierto} \rightarrow \text{Añadir } |X'| \cdot \eta \text{ objetos de la clase de } X \\ \text{Fallo} \begin{cases} \text{Añadir } |X'| \cdot \eta \text{ objetos de la clase de } X \\ \text{Eliminar } |X'| \cdot \eta \text{ objetos de la clase predicha.} \end{cases} \end{cases} \end{cases}$$

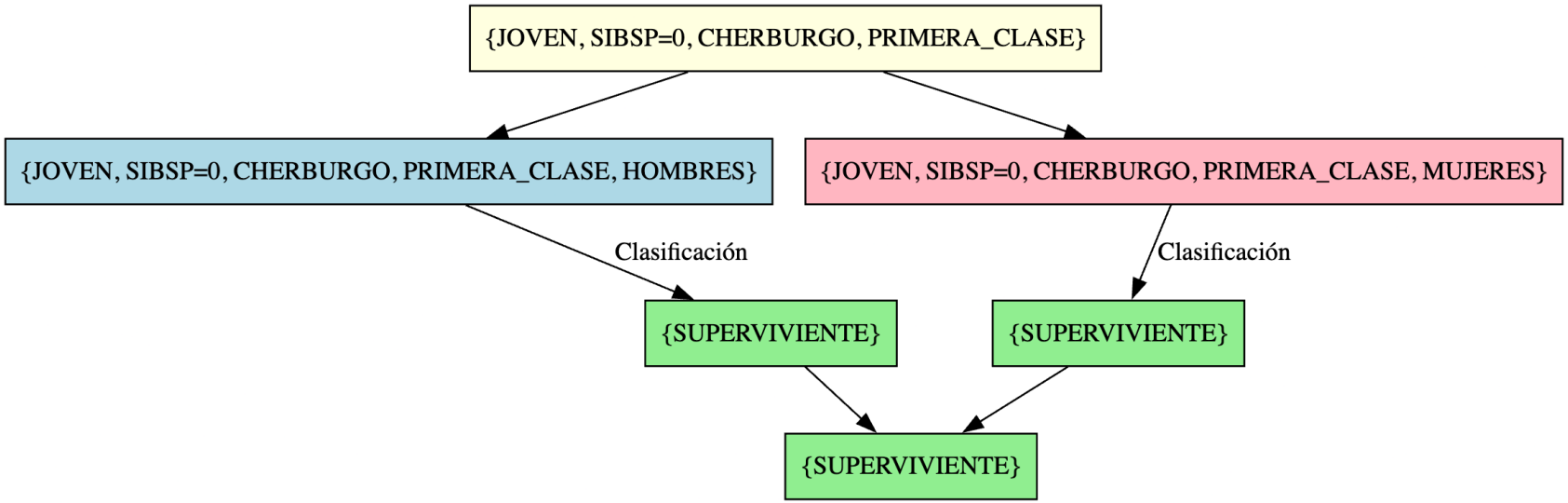
Modelos de caja negra

- Extracción del conocimiento de modelos de caja negra.
- Extracción del ***monster context***.
- Modelo fijo con *monster context*.
- Extraer base STEM del *monster context*.

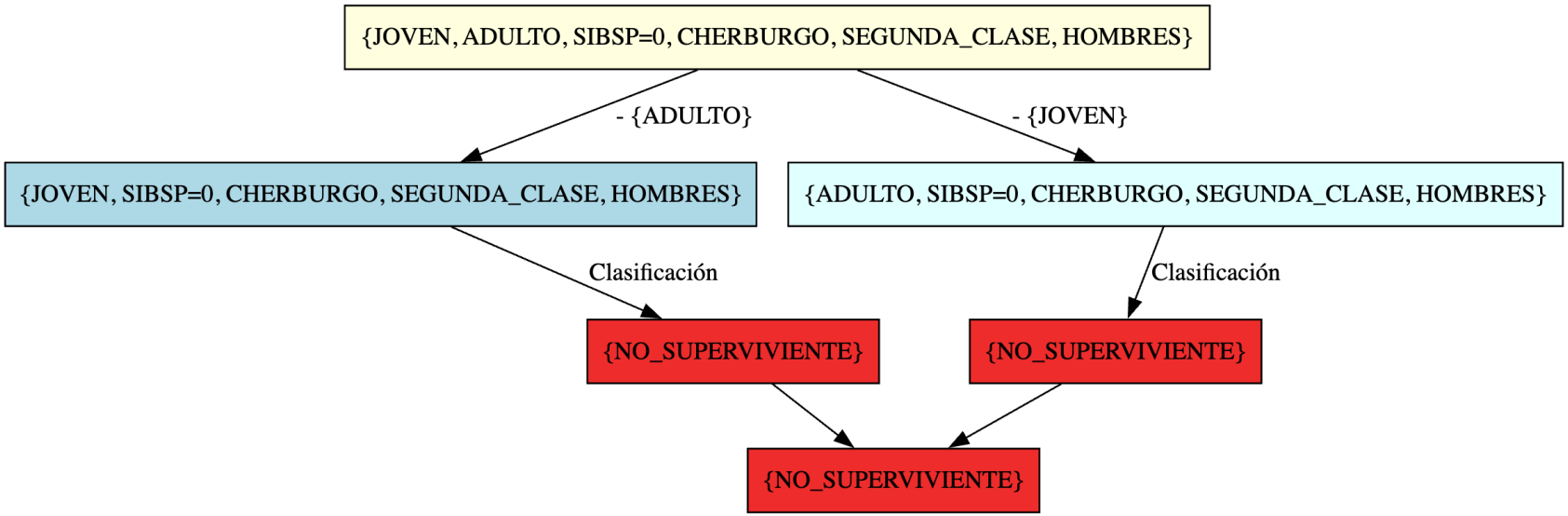
Base STEM del *monster context*

- La base STEM del *monster context* puede clasificar cualquier conjunto de entrada si es **completo**.
- Imposibilidad de clasificar conjuntos de entrada **incompletos**.
- Imposibilidad de clasificar conjuntos de entrada **incoherentes**.

Conjuntos de entrada incompletos



Conjuntos de entrada incoherentes



Métodos

- Los modelos presentados solo pueden trabajar con conjuntos de datos binarios.
- Emplear métodos de discretización de variables continuas **explicables**.
- Compresión de la información.
  - Distribuciones paramétricas y estadístico suficiente minimal.
  - Árboles de decisión discretizadores.
  - *Agglomerative Information Bottleneck*

Resultados

- Datasets binarios mediante Árboles de decisión.
- Modo determinista.
- Medida de inclusión

$$d(A, B) = \frac{|A \cap B|}{|A|}$$

- Hiperparámetros por defecto (*Scikit-learn*)

Resultados Titanic

Modelo	T/T	T/V	T+V/T+V	T+V/Test
Logistic Regression	0.8074	0.8047	0.8061	0.7886
Decision Tree	0.8446	0.7811	0.8381	0.7852
Random Forest	0.8446	0.7778	0.8381	0.7852
Support Vector Classification	0.8243	0.7946	0.8297	0.7987
KNN (k=5)	0.8311	0.7980	0.8179	0.8154
KNN (k=7)	0.8311	0.7946	0.8212	0.8020
KNN (k=10)	0.8108	0.7845	0.8179	0.8054
Naive Bayes	0.7939	0.7879	0.7808	0.8054
MLP Classifier	0.8412	0.7811	0.8347	0.7919
FCA Iterativo	0.8412	0.7879	0.8331	0.8020
FCA Fijo	0.8412	0.7811	0.8347	0.7986

Resultados Car Evaluation

Modelo	Accuracy	Recall	F1
Logistic Regression Multinomial	0.9201	0.9201	0.9230
Decision Tree	1.0000	1.0000	1.0000
Random Forest	1.0000	1.0000	1.0000
Support Vector Classification	0.9890	0.9890	0.9889
KNN (k=5)	0.9410	0.9410	0.9429
KNN (k=7)	0.9462	0.9462	0.9491
KNN (k=10)	0.9543	0.9543	0.9566
Naive Bayes	0.8027	0.8027	0.7864
MLP Classifier	1.0000	1.0000	1.0000
FCA Iterativo	1.0000	1.0000	1.0000
FCA Fijo	1.0000	1.0000	1.0000

Resultados Dataset Iris

Modelo	T/T	T/V	T+V/T+V	T+V/Test
Logistic Regression Multinomial	0.9556	0.9333	0.9500	0.9667
Decision Tree	0.9667	0.9000	0.9583	1.0000
Random Forest	0.9667	0.9333	0.9583	1.0000
Support Vector Classification	0.9667	0.9333	0.9583	1.0000
KNN (k=5)	0.9556	0.9333	0.9500	1.0000
KNN (k=7)	0.9556	0.9333	0.9417	1.0000
KNN (k=10)	0.9333	0.9667	0.9417	1.0000
Naive Bayes	0.9333	0.9667	0.9417	0.9667
MLP Classifier	0.9667	0.9333	0.9583	0.9667
FCA Iterativo	0.9667	0.9000	0.9583	1.0000
FCA Fijo	0.9667	0.9000	0.9583	1.0000

Resultados Dataset Heart Disease

Modelo	T/T	T/V	T+V/T+V	T+V/Test
Logistic Regression	0.8785	0.8197	0.8884	0.8525
Decision Tree	0.9945	0.7869	0.9959	0.8361
Random Forest	0.9945	0.8033	0.9959	0.8197
Support Vector Classification	0.9227	0.7869	0.9339	0.8689
KNN (k=5)	0.8729	0.8361	0.8471	0.8361
KNN (k=7)	0.8674	0.8525	0.8430	0.8525
KNN (k=10)	0.8453	0.7869	0.8388	0.8525
Naive Bayes	0.7403	0.7541	0.8347	0.9016
MLP Classifier	0.9945	0.7213	0.9959	0.7869
FCA Iterativo	0.9945	0.8361	0.9959	0.8197
FCA Fijo	0.9945	0.8361	0.9959	0.8197

Conclusiones

- Modelos FCA con rendimientos similares a *Random Forest*.
- Base STEM como clasificador eficiente y compacto.
- Inestabilidad de las bases de Luxenburger.
- Desarrollo de librerías.

Modelos de aprendizaje automático basados en Análisis Formal de Conceptos

Adam Lei Yi Chen Abolacio

Dirigido por D. Joaquín Borrego Díaz