Estudio experimental de algoritmos de cálculo de retículos en Análisis Formal de Conceptos

Miguel Ángel Cantarero López

Universidad de Granada

Doble grado en Ingeniería Informática y Matemáticas

2 de diciembre de 2022

Trabajo tutorizado por Nicolás Marín Ruiz y Daniel Sánchez Fernández

Contenidos

1. Introducción
2. Teoría del Análisis Formal de Conceptos
3. Algoritmos
4. Experimentación
5. Conclusiones y vías futuras

Introducción

Introducción

El FCA es un campo de la matemática aplicada que formaliza matemáticamente el significado de la palabra "concepto".

Introducción

El FCA es un campo de la matemática aplicada que formaliza matemáticamente el significado de la palabra "concepto".

- **[1977]:** Retículos matemáticos desde un punto de vista computacional.

Introducción

El FCA es un campo de la matemática aplicada que formaliza matemáticamente el significado de la palabra "concepto".

- **[1977]:** Retículos matemáticos desde un punto de vista computacional.
- **[1981]:** Rudolf Wille formaliza las ideas anteriores y crea el FCA.

Introducción

El FCA es un campo de la matemática aplicada que formaliza matemáticamente el significado de la palabra "concepto".

- **[1977]:** Retículos matemáticos desde un punto de vista computacional.
- **[1981]:** Rudolf Wille formaliza las ideas anteriores y crea el FCA.
- **[1999]:** Se publica el libro de Formal Concept Analysis (Bernhard Ganter y Rudolf Wille).

Introducción

El FCA es un campo de la matemática aplicada que formaliza matemáticamente el significado de la palabra "concepto".

- **[1977]:** Retículos matemáticos desde un punto de vista computacional.
- **[1981]:** Rudolf Wille formaliza las ideas anteriores y crea el FCA.
- **[1999]:** Se publica el libro de Formal Concept Analysis (Bernhard Ganter y Rudolf Wille).
- **[2002]:** S. Kuznetsov y S. Obiedkov publican un artículo comparativo de algoritmos.

Introducción

El FCA es un campo de la matemática aplicada que formaliza matemáticamente el significado de la palabra "concepto".

- **[1977]:** Retículos matemáticos desde un punto de vista computacional.
- **[1981]:** Rudolf Wille formaliza las ideas anteriores y crea el FCA.
- **[1999]:** Se publica el libro de Formal Concept Analysis (Bernhard Ganter y Rudolf Wille).
- **[2002]:** S. Kuznetsov y S. Obiedkov publican un artículo comparativo de algoritmos.
- **[2004]:** Comienzan las ICFCA.

Motivación

Contexto formal

	Cine	Ropa	Alimentación	Pantalla Gigante	Electrónica
Nevada		x	x		x
Neptuno	x	x	x		
Serrallo	x	x	x	x	

Motivación

Contexto formal

	Cine	Ropa	Alimentación	Pantalla Gigante	Electrónica
Nevada		x	x		x
Neptuno	x	x	x		
Serrallo	x	x	x	x	

Conceptos formales

- ({Nevada},{Ropa, Alimentación, Electrónica})
- ({Neptuno, Serrallo},{Cine,Ropa,Alimentación})

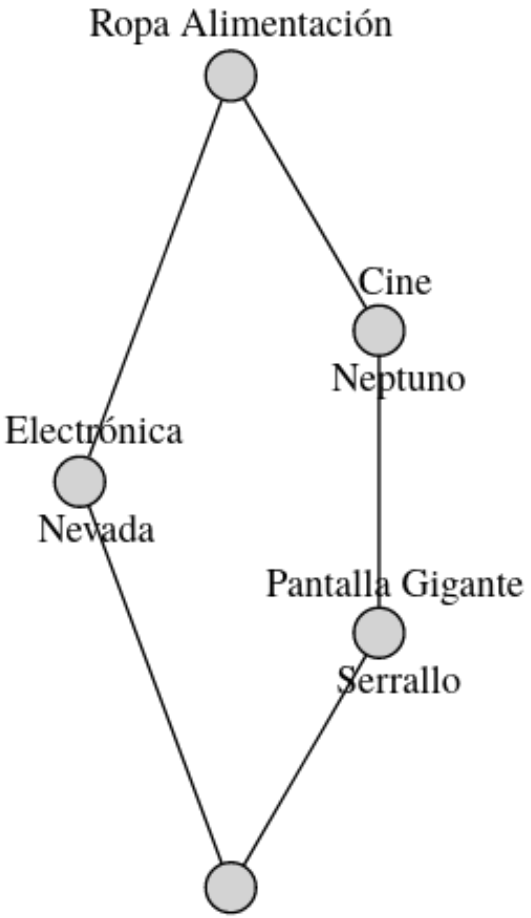
Motivación

Contexto formal

	Cine	Ropa	Alimentación	Pantalla Gigante	Electrónica
Nevada		x	x		x
Neptuno	x	x	x		
Serrallo	x	x	x	x	

Conceptos formales

- ({Nevada},{Ropa, Alimentación, Electrónica})
- ({Neptuno, Serrallo},{Cine,Ropa,Alimentación})



Retículo de conceptos

Objetivos

- Comprender y explicar la Teoría del Análisis Formal de Conceptos.
- Recopilar los algoritmos más importantes que existen en la literatura.
- Implementar los algoritmos y analizar su desempeño mediante una experimentación equitativa.
- Publicar la experimentación para que sea reproducible.

Teoría del Análisis Formal de Conceptos

Contexto formal

Definición (Contexto formal)

Un contexto formal $\mathbb{K} := (\mathcal{G}, \mathcal{M}, \mathcal{I})$ está compuesto por un conjunto $\mathcal{G}$, cuyos elementos se llaman objetos, un conjunto $\mathcal{M}$, cuyos elementos se llaman atributos, y una relación binaria $\mathcal{I} \subseteq \mathcal{G} \times \mathcal{M}$.

Contexto formal

Definición (Contexto formal)

Un contexto formal $\mathbb{K} := (\mathcal{G}, \mathcal{M}, \mathcal{I})$ está compuesto por un conjunto $\mathcal{G}$, cuyos elementos se llaman objetos, un conjunto $\mathcal{M}$, cuyos elementos se llaman atributos, y una relación binaria $\mathcal{I} \subseteq \mathcal{G} \times \mathcal{M}$.

El contexto formal se traduce en la práctica a una tabla cruzada.

Contexto formal

	Cine	Ropa	Alimentación	Pantalla Gigante	Electrónica
Nevada		x	x		x
Neptuno	x	x	x		
Serrallo	x	x	x	x	

Ejemplo 1

$$\mathcal{G} = \{\text{Nevada, Neptuno, Serrallo}\}.$$

$$\mathcal{M} = \{\text{Cine, Ropa, Alimentación, Pantalla Gigante, Electrónica}\}.$$

$\{$(Nevada, Ropa), (Nevada, Alimentación), (Nevada, Electrónica), (Neptuno, Cine), (Neptuno, Ropa), (Neptuno, Alimentación), (Serrallo, Cine), (Serrallo, Ropa), (Serrallo, Alimentación), (Serrallo, Pantalla)$\}$

Operadores de derivación

Definición (Operador de derivación para objetos)

Sea $A \subseteq \mathcal{G}$, $A' := \{m \in \mathcal{M} \mid g\mathcal{I}m \ \ \forall g \in A\}$, $\{\emptyset\}' = \mathcal{M}$

Definición (Operador de derivación para atributos)

Sea $B \subseteq \mathcal{M}$, $B' := \{g \in \mathcal{G} \mid g\mathcal{I}m \ \ \forall m \in B\}$, $\{\emptyset\}' = \mathcal{G}$

Operadores de derivación

Definición (Operador de derivación para objetos)

Sea $A \subseteq \mathcal{G}$, $A' := \{m \in \mathcal{M} \mid g\mathcal{I}m \ \ \forall g \in A\}$, $\{\emptyset\}' = \mathcal{M}$

Definición (Operador de derivación para atributos)

Sea $B \subseteq \mathcal{M}$, $B' := \{g \in \mathcal{G} \mid g\mathcal{I}m \ \ \forall m \in B\}$, $\{\emptyset\}' = \mathcal{G}$

	Cine	Ropa	Alimentación	Pantalla Gigante	Electrónica
Nevada		x	x		x
Neptuno	x	x	x		
Serrallo	x	x	x	x	

Ejemplo 2

$$\{\text{Neptuno}\}' = \{\text{Cine, Ropa, Alimentación}\}$$

$$\{\text{Ropa, Electrónica}\}' = \{\text{Nevada}\}$$

Concepto formal

Definición (Concepto formal)

Un concepto formal de un contexto formal $\mathbb{K} := (\mathcal{G}, \mathcal{M}, \mathcal{I})$ se define como una pareja $(A, B) \subseteq \mathcal{I}$ con $A \subseteq \mathcal{G}$, $B \subseteq \mathcal{M}$, que cumple $A' = B$ y $B' = A$.

Concepto formal

Definición (Concepto formal)

Un concepto formal de un contexto formal $\mathbb{K} := (\mathcal{G}, \mathcal{M}, \mathcal{I})$ se define como una pareja $(A, B) \subseteq \mathcal{I}$ con $A \subseteq \mathcal{G}$, $B \subseteq \mathcal{M}$, que cumple $A' = B$ y $B' = A$.

Ejemplo 4

$(\{\text{Neptuno,Serrallo}\}, \{\text{Cine,Ropa, Alimentación}\})$

	Cine	**Ropa**	**Alimentación**	**Pantalla Gigante**	**Electrónica**
Nevada		x	x		x
Neptuno	**x**	**x**	**x**		
Serrallo	**x**	**x**	**x**	x	

Orden y Retículo de conceptos

Definición (Relación de orden entre conceptos)

Sean (A_1, B_1), (A_2, B_2) conceptos formales, (A_1, B_1) es un subconcepto de (A_2, B_2) si $A_1 \subseteq A_2$ y $B_2 \subseteq B_1$. Cuando esto ocurra escribiremos $(A_1, B_1) \leq (A_2, B_2)$.

Orden y Retículo de conceptos

Definición (Relación de orden entre conceptos)

Sean (A_1, B_1), (A_2, B_2) conceptos formales, (A_1, B_1) es un subconcepto de (A_2, B_2) si $A_1 \subseteq A_2$ y $B_2 \subseteq B_1$. Cuando esto ocurra escribiremos $(A_1, B_1) \leq (A_2, B_2)$.

Definición (Retículo de conceptos)

El conjunto de todos los conceptos formales de $(\mathcal{G}, \mathcal{M}, \mathcal{I})$ ordenados por la anterior relación de orden "$\leq$" definida es un retículo de conceptos y se denota por

$$\underline{\mathfrak{B}}(\mathcal{G}, \mathcal{M}, \mathcal{I}) := (\mathfrak{B}(\mathcal{G}, \mathcal{M}, \mathcal{I}), \leq).$$

Retículo de conceptos

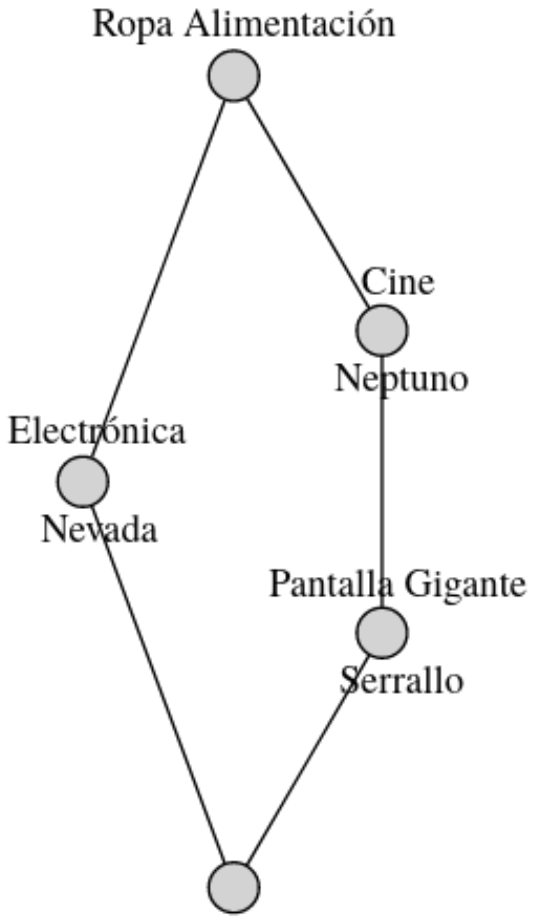
Ejemplo 5

$\underline{\mathfrak{B}}(\mathcal{G}, \mathcal{M}, \mathcal{I})$:={
({Nevada, Neptuno, Serrallo}, {Ropa, Alimentación}),
({Nevada}, {Ropa, Alimentación, Electrónica}),
({Neptuno, Serrallo}, {Cine, Ropa, Alimentación}),
({Serrallo}, {Cine, Ropa, Alimentación, Pantalla}),
({∅},{Cine, Ropa, Alimentación, Pantalla,
Electrónica})
}

Retículo de conceptos

Ejemplo 5

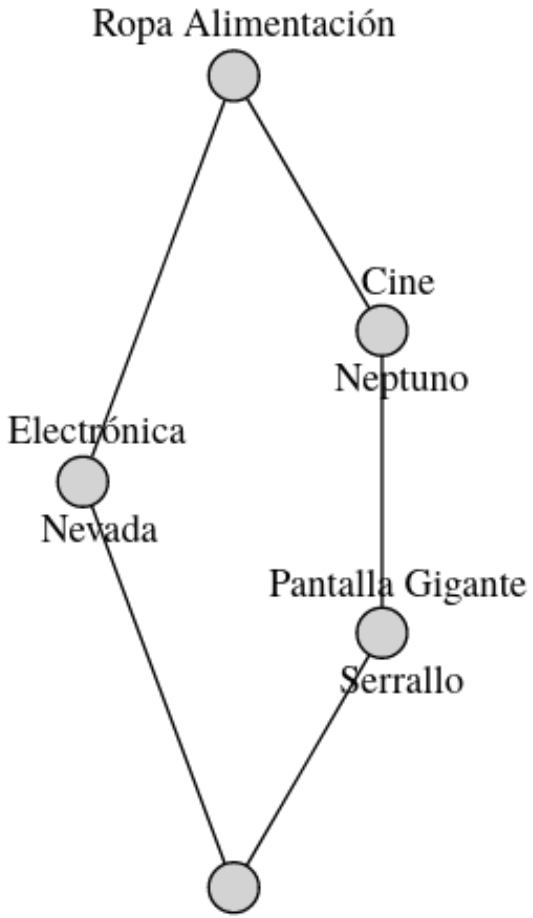
$\underline{\mathfrak{B}}(\mathcal{G}, \mathcal{M}, \mathcal{I}) :=\{$
({Nevada, Neptuno, Serrallo}, {Ropa, Alimentación}),
({Nevada}, {Ropa, Alimentación, Electrónica}),
({Neptuno, Serrallo}, {Cine, Ropa, Alimentación}),
({Serrallo}, {Cine, Ropa, Alimentación, Pantalla}),
({∅},{Cine, Ropa, Alimentación, Pantalla, Electrónica})
}



Retículo de conceptos

Ejemplo 5

$\underline{\mathfrak{B}}(\mathcal{G}, \mathcal{M}, \mathcal{I}) :=\{$
({Nevada, Neptuno, Serrallo}, {Ropa, Alimentación}),
({Nevada}, {Ropa, Alimentación, Electrónica}),
({Neptuno, Serrallo}, {Cine, Ropa, Alimentación}),
({Serrallo}, {Cine, Ropa, Alimentación, Pantalla}),
({∅},{Cine, Ropa, Alimentación, Pantalla, Electrónica})
}



Teorema Básico del FCA

Teorema Básico de los Retículos de Conceptos (Parte I)

El retículo de conceptos de cualquier contexto $(\mathcal{G}, \mathcal{M}, \mathcal{I})$ es un retículo completo. Para cualquier subconjunto arbitrario

$$\{(A_t, B_t) | t \in T\} \subseteq \mathfrak{B}(\mathcal{G}, \mathcal{M}, \mathcal{I}),$$

de conceptos formales, el ínfimo viene dado por

$$\text{ínf}\{(A_t, B_t)_{t \in T}\} = \bigwedge_{t \in T} (A_t, B_t) = (\bigcap_{t \in T} A_t, (\bigcup_{t \in T} B_t)'')$$

y el supremo viene dado por

$$\sup\{(A_t, B_t)_{t \in T}\} = \bigvee_{t \in T} (A_t, B_t) = ((\bigcup_{t \in T} A_t)'', \bigcap_{t \in T} B_t).$$

Teorema Básico del FCA

Teorema Básico de los Retículos de Conceptos (Parte II)

Un retículo completo $\underline{L}$ es isomorfo a $\underline{\mathfrak{B}}(\mathcal{G}, \mathcal{M}, \mathcal{I})$ si y solo si existen funciones $\overline{\gamma} : \mathcal{G} \to L$ y $\overline{\mu} : \mathcal{M} \to L$ tales que $\overline{\gamma}(\mathcal{G})$ es supremo-denso y $\overline{\mu}(\mathcal{M})$ es ínfimo-denso en $\underline{L}$ y

$$g \mathcal{I} m \iff \overline{\gamma}(g) \leq \overline{\mu}(m),$$

En particular $\underline{L} \cong \underline{\mathfrak{B}}(\mathcal{G}, \mathcal{M}, \mathcal{I})$.

Aplicaciones

- Recuperación de información (buscadores).
- Biología.
- Diseño de software.
- Análisis de datos (visualización, preprocesamiento, . . .).

Algoritmos

Tipos

Objetivo: Construir el retículo de conceptos a partir de un contexto formal según la aplicación que se le desee dar.

Tipos

Objetivo: Construir el retículo de conceptos a partir de un contexto formal según la aplicación que se le desee dar.

Por lotes (secuenciales)

- NextClosure (Ganter).
- Algoritmo de Lindig.
- InClose.
- Inherit-Concepts (Berry).
- Algoritmo de Bordat.

Por lotes (paralelos)

- MapReduce Ganter.
- Algoritmo de Krajca.

Tipos

Objetivo: Construir el retículo de conceptos a partir de un contexto formal según la aplicación que se le desee dar.

Por lotes (secuenciales)

- NextClosure (Ganter).
- Algoritmo de Lindig.
- InClose.
- Inherit-Concepts (Berry).
- Algoritmo de Bordat.

Por lotes (paralelos)

- MapReduce Ganter.
- Algoritmo de Krajca.

Incrementales (secuenciales)

- Norris.
- Godin.
- AddIntent.

Algoritmos

- Tratan de evitar calcular varias veces el mismo concepto.
- Heurísticas diferentes desarrolladas por sus autores.
- Basados en teoremas o propiedades desarrolladas por sus autores.

Algoritmos

- Tratan de evitar calcular varias veces el mismo concepto.
- Heurísticas diferentes desarrolladas por sus autores.
- Basados en teoremas o propiedades desarrolladas por sus autores.

Algoritmo de Lindig

- Estrategia "bottom-up".
- Sea (A, B) un concepto formal,

$$S = \{((A \cup \{g\})'', (A \cup \{g\})') | g \notin A\},$$

es el conjunto de todos los vecinos superiores de (A, B).

Implementación

C++:

- <chrono>: *steady_clock::now()*
- <random>: *uniform_int_distribution*

El contenido del trabajo se encuentra disponible en:

`https://github.com/miguecl97/TFG-AlgoritmosFCA` ,

tanto datasets, retículos del resultado, pruebas de caja negra, ...

Experimentación

Conjuntos de datos

Valores:

- Número de atributos: $|\mathcal{M}|$ (lo fijaremos en **100**).

Conjuntos de datos

Valores:

- Número de atributos: $|\mathcal{M}|$ (lo fijaremos en **100**).
- Número de objetos: $|\mathcal{G}|$ (**variando** para cada ejecución de los experimentos).

Conjuntos de datos

Valores:

- Número de atributos: $|\mathcal{M}|$ (lo fijaremos en **100**).
- Número de objetos: $|\mathcal{G}|$ (**variando** para cada ejecución de los experimentos).
- Número medio de atributos que tiene cada objeto: $|g'|$ (**fijo** para cada experimento).

Conjuntos de datos

Valores:

- Número de atributos: $|\mathcal{M}|$ (lo fijaremos en **100**).
- Número de objetos: $|\mathcal{G}|$ (**variando** para cada ejecución de los experimentos).
- Número medio de atributos que tiene cada objeto: $|g'|$ (**fijo** para cada experimento).

Los atributos que posee cada objeto se eligen generando números aleatorios siguiendo una distribución uniforme.

Resultados (Conjuntos artificiales)

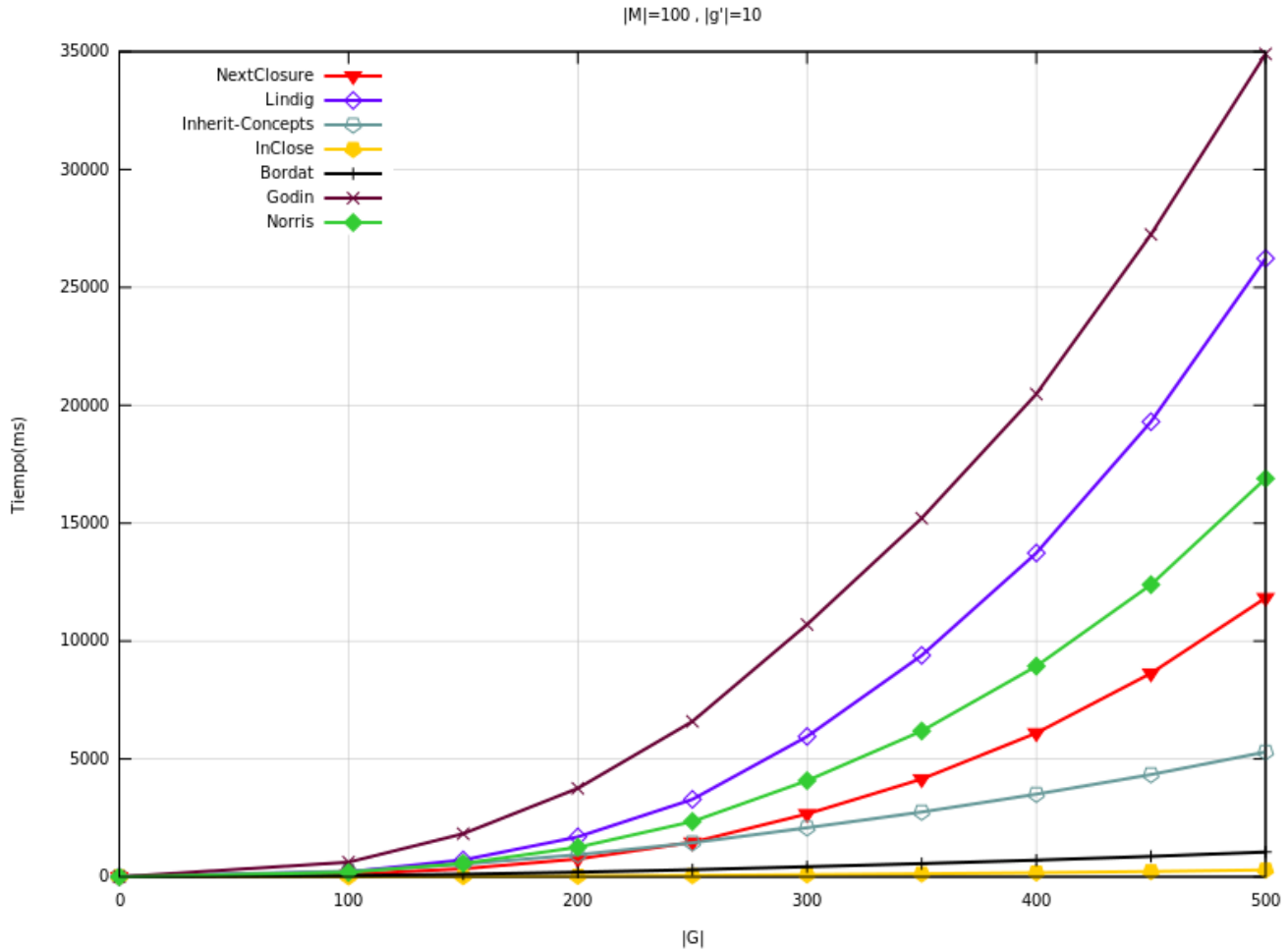


Figura: $|\mathcal{M}| = 100$, $|g'| = 10$, $|\mathcal{G}| = \{100, 200, 300, 400, 500\}$

Resultados (Conjuntos artificiales)

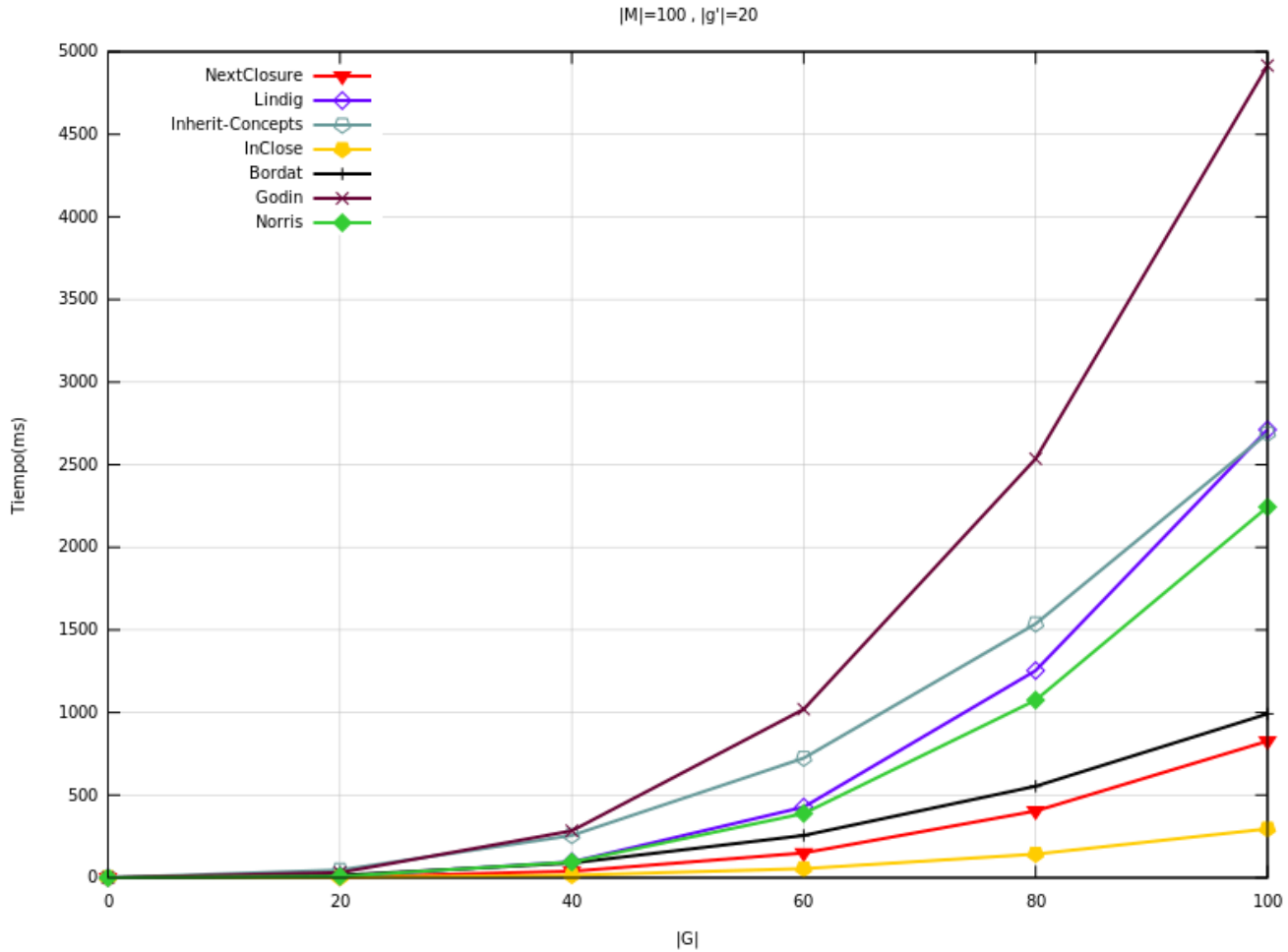


Figura: $|\mathcal{M}| = 100$, $|g'| = 20$, $|\mathcal{G}| = \{20, 40, 60, 80, 100\}$

Resultados (Conjuntos artificiales)

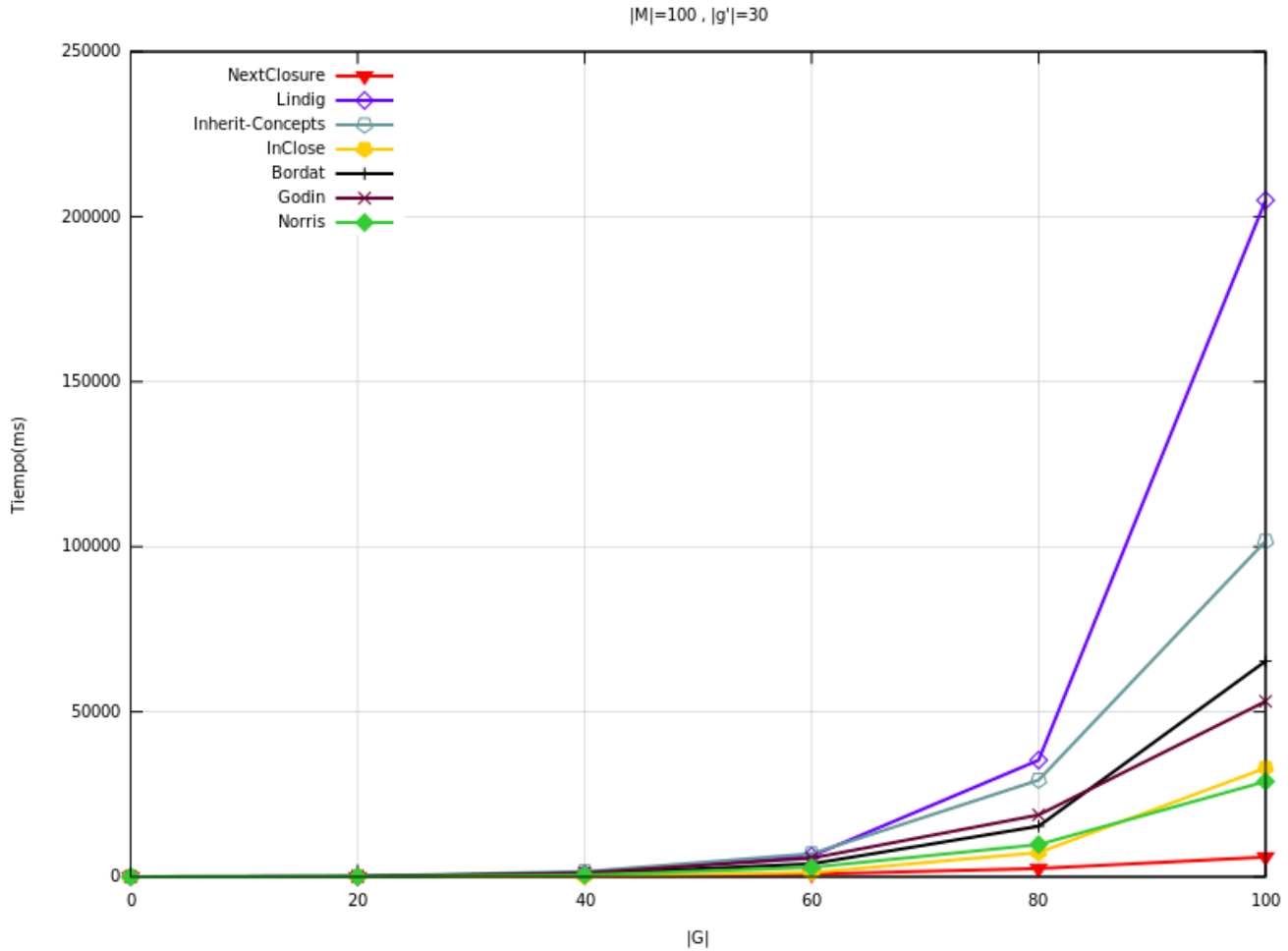


Figura: $|\mathcal{M}| = 100$, $|g'| = 30$, $|\mathcal{G}| = \{20, 40, 60, 80, 100\}$

Resultados (Conjuntos artificiales)

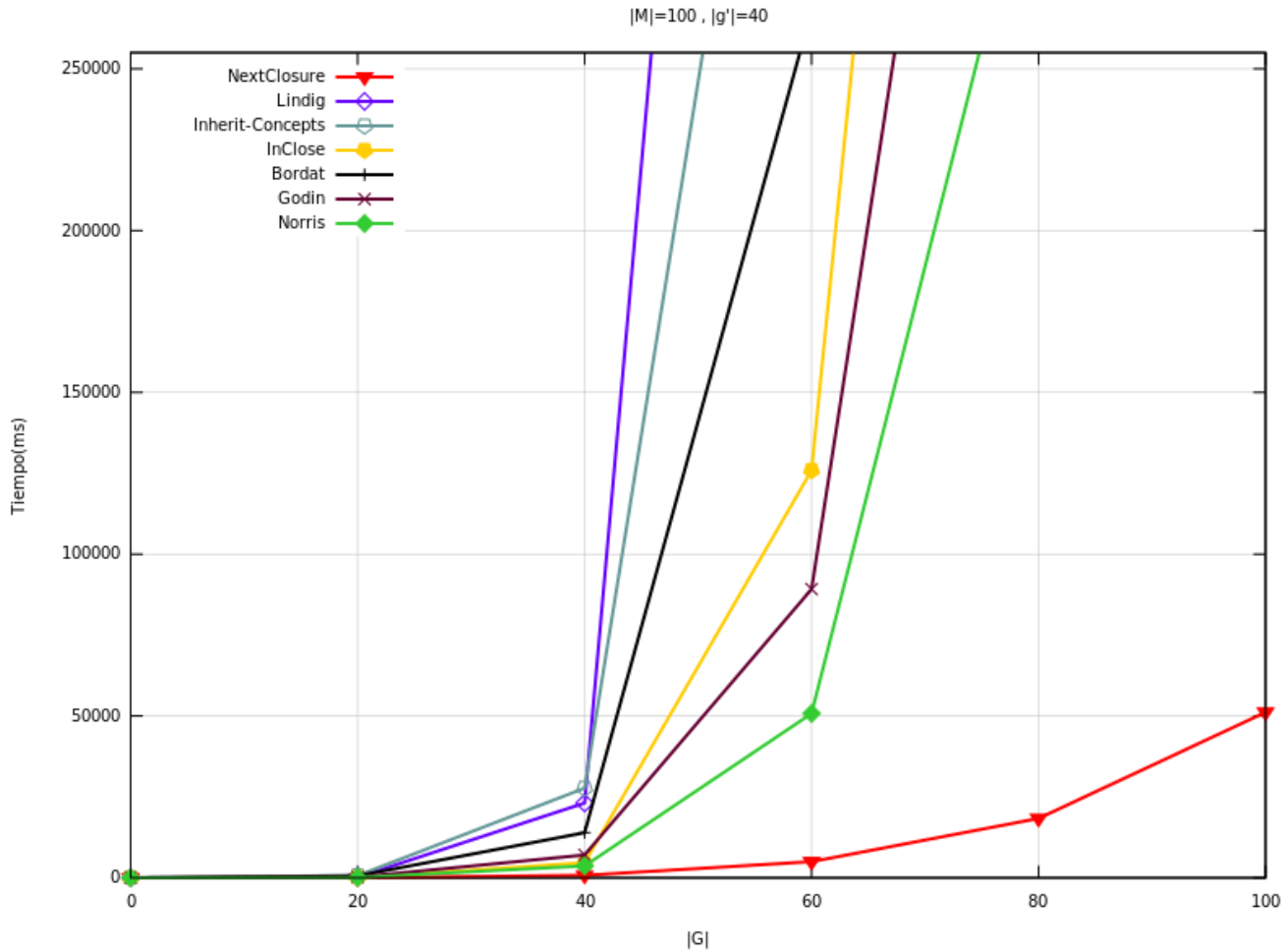
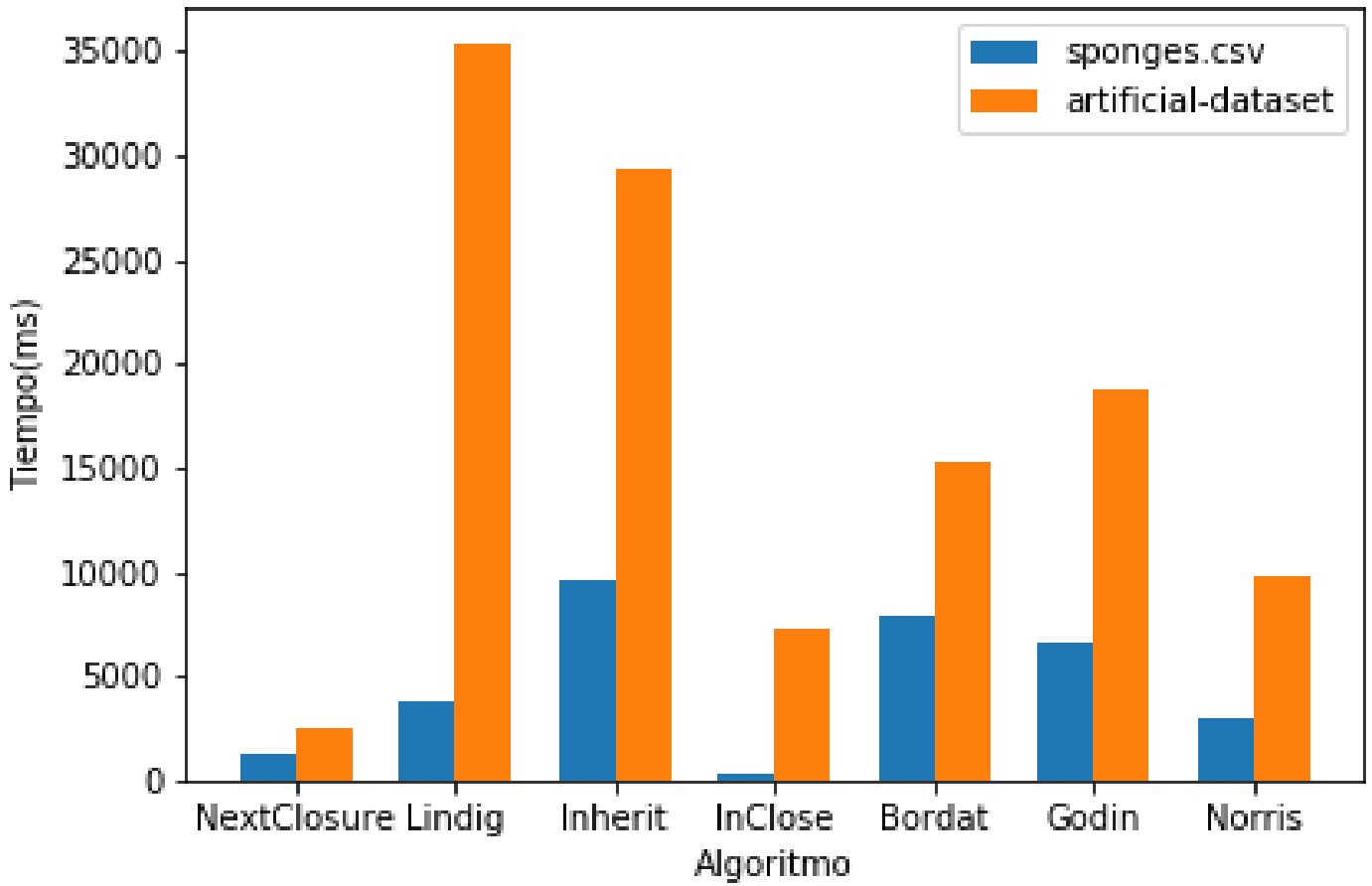


Figura: $|\mathcal{M}| = 100$, $|g'| = 40$, $|\mathcal{G}| = \{20, 40, 60, 80, 100\}$

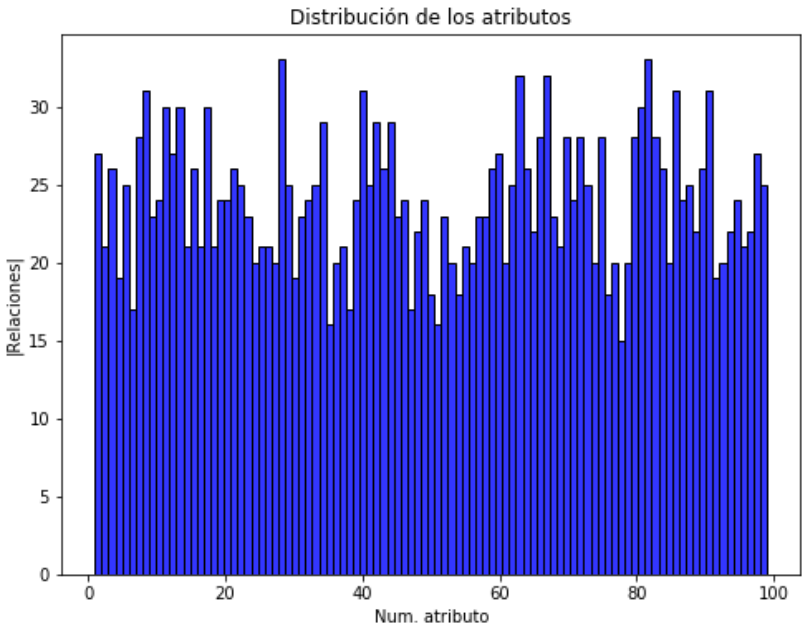
Resultados (Conjuntos de datos reales)

Sponges.csv $(|\mathcal{M}| = 100, |g'| = 29, |\mathcal{G}| = 76)$

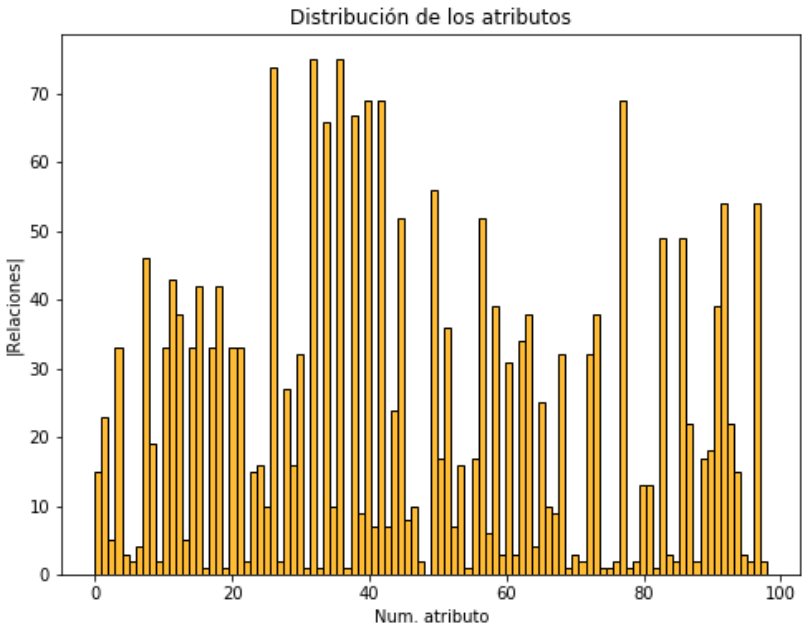


Distribuciones

Artificial



Sponges.csv



Conclusiones y vías futuras

Conclusiones

- Hemos obtenido una imagen del estado de los principales algoritmos que existen para el cálculo de retículos de conceptos.

Conclusiones

- Hemos obtenido una imagen del estado de los principales algoritmos que existen para el cálculo de retículos de conceptos.
- Ante conjuntos de datos cuyas relaciones de los atributos sigan una distribución uniforme y que tengan poca densidad, el algoritmo que menor tiempo de ejecución requiere es el InClose.
- Si el conjunto de datos tiene una densidad elevada el algoritmo que mejor funciona es el NextClosure.
- Si la distribución de los atributos en el conjunto de datos no sigue una distribución uniforme no podemos predecir el desempeño de los algoritmos con este estudio.

Conclusiones

- Hemos obtenido una imagen del estado de los principales algoritmos que existen para el cálculo de retículos de conceptos.
- Ante conjuntos de datos cuyas relaciones de los atributos sigan una distribución uniforme y que tengan poca densidad, el algoritmo que menor tiempo de ejecución requiere es el InClose.
- Si el conjunto de datos tiene una densidad elevada el algoritmo que mejor funciona es el NextClosure.
- Si la distribución de los atributos en el conjunto de datos no sigue una distribución uniforme no podemos predecir el desempeño de los algoritmos con este estudio.
- Hemos visto una aplicación a los conocimientos que hemos aprendido durante el grado de Matemáticas y del grado en Ingeniería Informática, que requiere de ambas competencias para su correcta comprensión y puesta en práctica.

Trabajo futuro

- Repetir la experimentación utilizando otra distribución de las relaciones de los atributos, no necesariamente uniforme, para así tener más casuísticas que permitan predecir el comportamiento de los algoritmos ante otros tipos de conjuntos de datos.

- Paralelizar los algoritmos que mejor rendimiento han tenido como el InClose o el NextClosure y comparar la mejora en tiempo de ejecución.

- Realizar un análisis en profundidad de los algoritmos paralelos que existen en la literatura e incorporarlos al estudio.

Gracias por su atención.

Ejecución incremental

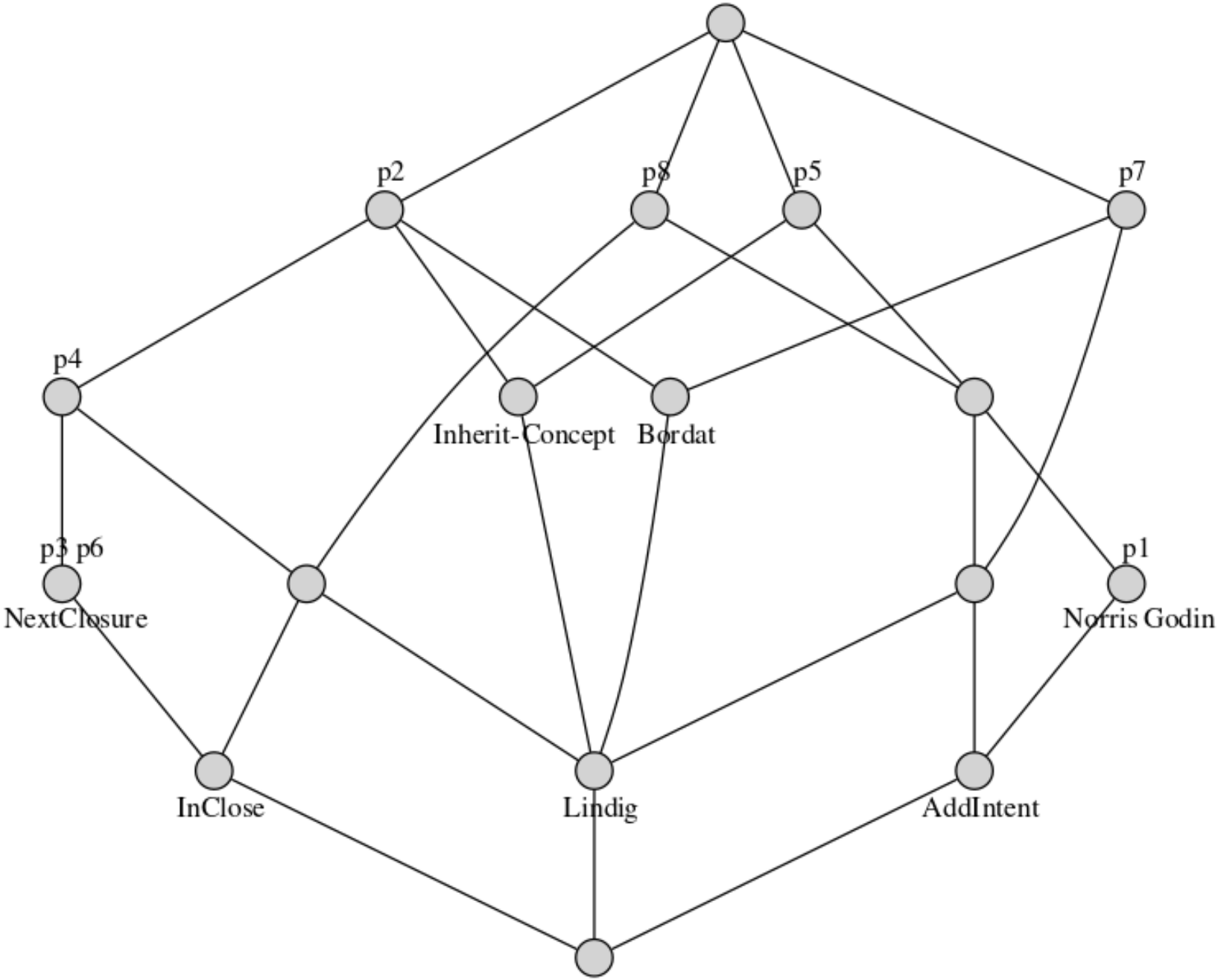
$\|g'\|$	$\|\mathcal{G}\|$	**NextClosure**	**Norris**	**Godin**
40	60	4990	50786	89212,9
40	80	18359,5	327035,3	542232,6
	Tiempo(ms) en añadir 20 objetos (de 60 a 80):	23349,5	327035,3	542232,6

Tabla: Tabla comparativa de tiempos de ejecución en ms para $|\mathcal{M}| = 100$, que tardaría cada algoritmo en añadir 20 objetos al retículo.

$\|g'\|$	$\|\mathcal{G}\|$	**NextClosure**	**Norris**	**NextClosure (incrementalmente)**	**Norris (incrementalmente)**
10	20	1	1,8	1	1,8
10	40	7,6	7,88	8,6	7,88
10	60	26,7	43,88	35,3	43,88
10	80	56,2	95	91,5	95
10	100	107,1	185,1	198,6	185,1
	Tiempo total (ms):			197,1	185,1

Tabla: Tabla comparativa de tiempos de ejecución para una ejecución incremental con $|\mathcal{M}| = 100$ y $|g'| = 10$.

Retículo de propiedades



Diferencia distribución atributos

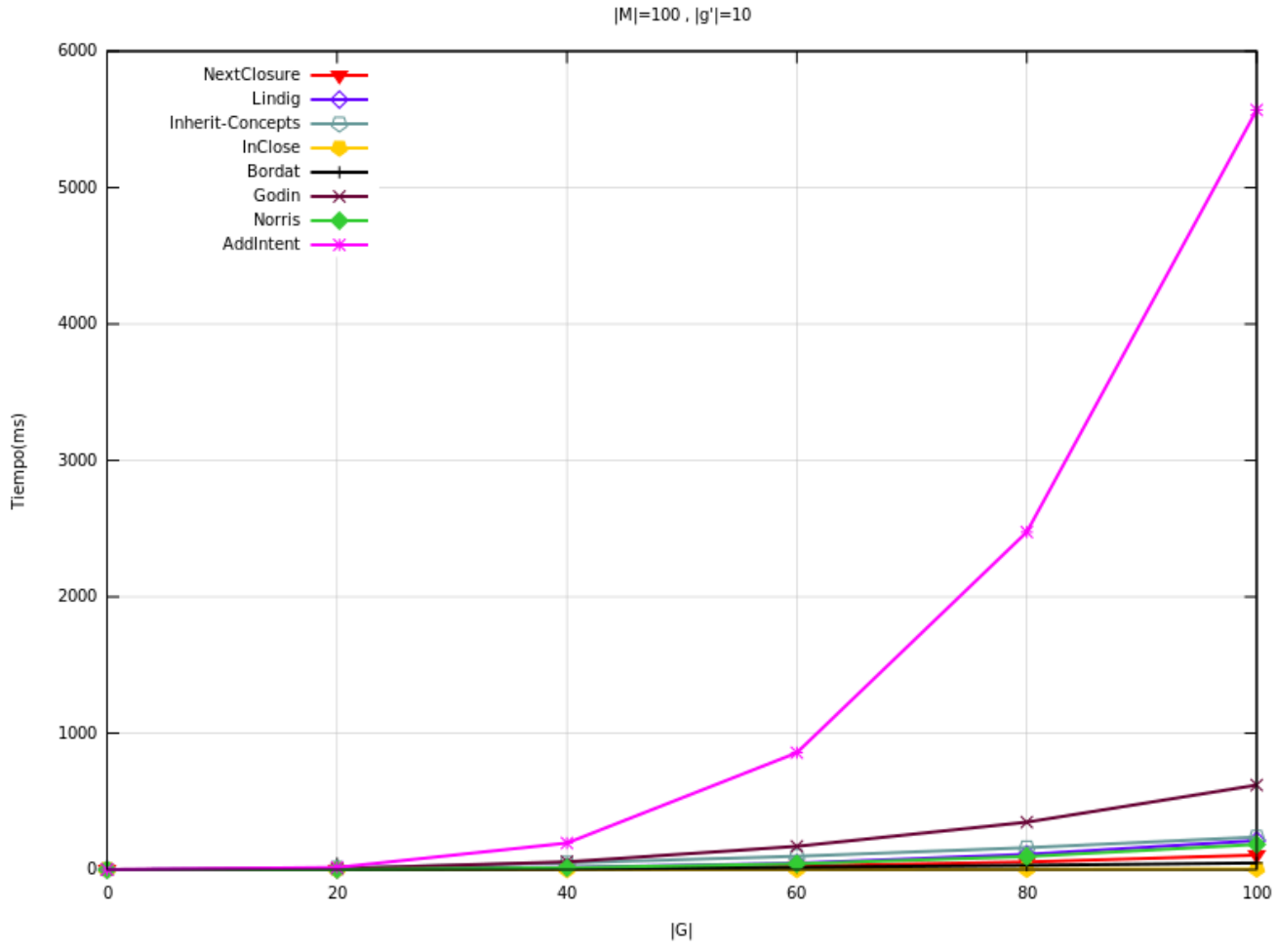
	a1	a2	a3	a4
o1		X	X	
o2	X		X	
o3		X		X

- Concepto 1: ({}, {a1, a2, a3, a4})
- Concepto 2: ({o1}, {a2, a3})
- Concepto 3: ({o2}, {a1, a3})
- Concepto 4: ({o3}, {a2, a4})
- Concepto 5: ({o1, o2}, {a3})
- Concepto 6: ({o1, o3}, {a2})
- Concepto 7: ({o1, o2, o3}, {})

	a1	a2	a3	a4
o1	X	X		
o2	X			
o3	X		X	X

- Concepto 1: ({}, {a1, a2, a3, a4})
- Concepto 2: ({o1}, {a1, a2})
- Concepto 3: ({o3}, {a1, a3, a4})
- Concepto 4: ({o1, o2, o3}, {a1})

Resultados (Conjuntos de datos reales)



Aplicaciones

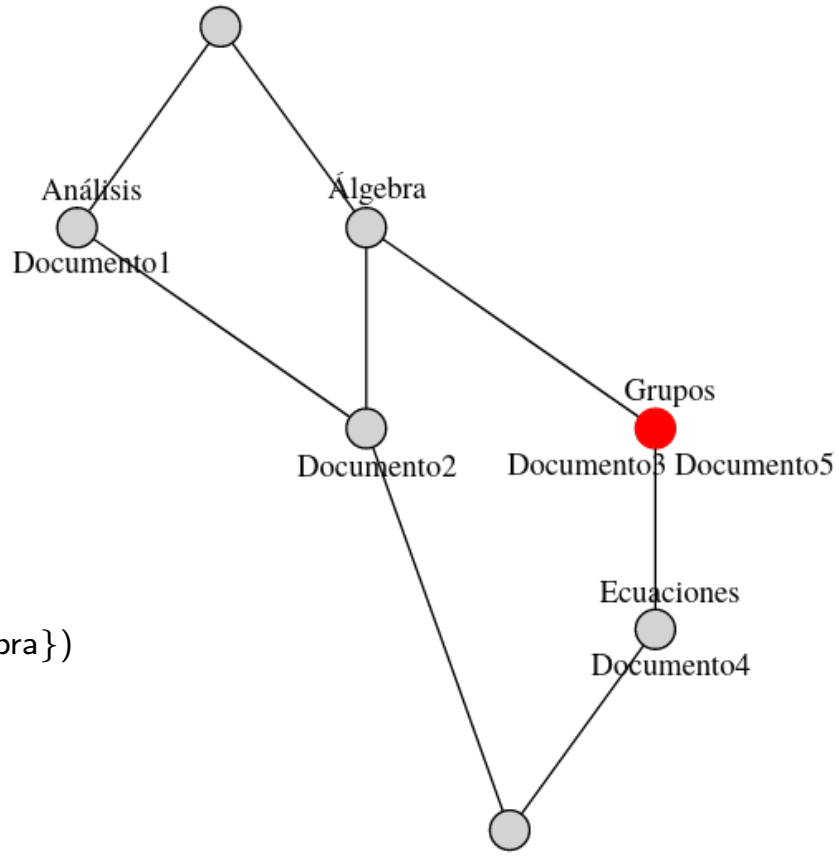
- Recuperación de Información (Buscadores).
- Biología, análisis de datos, preprocesamiento, agrupamiento, visualización, ...

Aplicaciones

- Recuperación de Información (Buscadores).
- Biología, análisis de datos, preprocesamiento, agrupamiento, visualización, ...

	Álgebra	Ecuaciones	Grupos	Análisis
Documento1				x
Documento2	x			x
Documento3	x		x	
Documento4	x	x	x	
Documento5	x		x	

Contexto de documentos, conceptos formales y retículo de conceptos asociado.



({Documento2,Documento3,Documento5,Documento4}, {Álgebra})

({Documento3,Documento4,Documento5},{Álgebra,Grupos})

({Documento4}, {Álgebra,Grupos,Ecuaciones})