

The background of the slide is a complex, abstract composition. It features a network of thin, reddish-brown lines connecting various points, creating a web-like structure. Interspersed among these lines are numerous small, colored dots in shades of green, blue, and orange. The overall aesthetic is technical and data-driven, with a mix of organic and geometric forms. A large, light gray, angular shape frames the central text area.

# Why Constraint-Based Mining?

# Why Constraint-Based Mining?

---

- ❑ Finding **all** the patterns in a dataset **autonomously** – unrealistic!
  - ❑ Too many patterns but not necessarily user-interested!
- ❑ Pattern mining should be an **interactive** process
  - ❑ User directs what to be mined using a **data mining query language** (or a graphical user interface)
- ❑ Constraint-based mining
  - ❑ User flexibility: Provides **constraints** on what to be mined
  - ❑ Optimization: Explores such constraints for efficient mining
  - ❑ **Constraint-based mining**: Constraint-pushing, similar to push selection first in DB query processing

# Constraints in General Data Mining

---

A data mining query can be in the form of a meta-rule or with the following language primitives

- **Knowledge type constraint**
  - Ex.: Classification, association, clustering, outlier finding, ...
- **Data constraint** – using SQL-like queries
  - Ex.: Find products sold together in NY stores this year
- **Dimension/level constraint**
  - Ex.: In relevance to region, price, brand, customer category
- **Rule (or pattern) constraint**
  - Ex.: Small sales (price < \$10) triggers big sales (sum > \$200)
- **Interestingness constraint**
  - Ex.: Strong rules:  $\text{min\_sup} \geq 0.02$ ,  $\text{min\_conf} \geq 0.6$ ,  $\text{min\_correlation} \geq 0.7$

# Meta-Rule Guided Mining

---

- ❑ A meta-rule can contain partially instantiated predicates & constants
  - ❑  $P_1(X, Y) \wedge P_2(X, W) \Rightarrow \text{buys}(X, \text{"iPad"})$
- ❑ The resulting mined rule can be
  - ❑  $\text{age}(X, \text{"15-25"}) \wedge \text{profession}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"iPad"})$
- ❑ In general, (meta) rules can be in the form of
  - ❑  $P_1 \wedge P_2 \wedge \dots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \dots \wedge Q_r$
- ❑ Method to find meta-rules
  - ❑ Find frequent ( $l + r$ ) predicates (based on *min-support*)
  - ❑ Push constants deeply when possible into the mining process
    - ❑ Using constraint-push techniques introduced in this lecture
  - ❑ Also, push `min_conf`, `min_correlation`, and other measures as early as possible (measures acting as constraints)



The background features a complex network of thin, light-colored lines forming a web-like structure. Overlaid on this are various data points: small green dots, larger blue dots, and a cluster of orange and red dots on the left. A horizontal band of purple and pink symbols, including arrows and mathematical-like characters, runs across the upper portion. A white, angular shape serves as a backdrop for the main text.

# Different Kinds of Constraints: Different Pruning Strategies

# Different Kinds of Constraints Lead to Different Pruning Strategies

---

- Constraints can be categorized as
  - **Pattern space pruning** constraints vs. **data space pruning** constraints
- **Pattern space pruning** constraints
  - **Anti-monotonic**: If constraint  $c$  is violated, its further mining can be terminated
  - **Monotonic**: If  $c$  is satisfied, no need to check  $c$  again
  - **Succinct**: If the constraint  $c$  can be enforced by directly manipulating the data
  - **Convertible**:  $c$  can be converted to monotonic or anti-monotonic if items can be properly ordered in processing
- **Data space pruning** constraints
  - **Data succinct**: Data space can be pruned at the initial pattern mining process
  - **Data anti-monotonic**: If a transaction  $t$  does not satisfy  $c$ , then  $t$  can be pruned to reduce data processing effort



The background features a complex, abstract design. It includes a network of thin, reddish-brown lines connecting various points, some of which are colored in shades of green and blue. There are also faint, light-colored geometric shapes and patterns, such as a grid of small plus signs and a series of horizontal lines with small arrows. The overall color palette is muted, with earthy tones and soft pastels.

# Constrained Mining with Pattern Anti-Monotonicity

# Pattern Space Pruning with Pattern Anti-Monotonicity

- Constraint  $c$  is *anti-monotone*

- If an itemset  $S$  **violates** constraint  $c$ , so does any of its superset
- That is, mining on itemset  $S$  can be terminated

- Ex. 1:  $c_1: \text{sum}(S.\text{price}) \leq v$  is **anti-monotone**

- Ex. 2:  $c_2: \text{range}(S.\text{profit}) \leq 15$  is **anti-monotone**

- Itemset  $ab$  violates  $c_2$  ( $\text{range}(ab) = 40$ )
- So does every superset of  $ab$

- Ex. 3.  $c_3: \text{sum}(S.\text{Price}) \geq v$  is **not anti-monotone**

- Ex. 4. Is  $c_4: \text{support}(S) \geq \sigma$  anti-monotone?

- Yes! Apriori pruning is essentially pruning with an anti-monotonic constraint!

TID	Transaction	Item	Profit
10	a, b, c, d, f, h	a	40
20	b, c, d, f, g, h	b	0
30	b, c, d, f, g	c	-20
40	a, c, e, f, g	d	-15
		e	-30
		f	-10
		g	20
		h	5

min\_sup = 2

price(item) > 0



The background of the slide is a complex, abstract composition. It features a network of thin, reddish-brown lines connecting various points, creating a web-like structure. Scattered throughout are numerous small, colored dots in shades of green, blue, and orange. A prominent white diagonal band runs from the top-left towards the bottom-right, serving as a backdrop for the title. On the left side, there is a rectangular inset showing a zoomed-in view of a data cluster with orange and brown points, overlaid with a grid of small white crosses. The overall aesthetic is technical and data-driven.

# Constrained Mining with Pattern Monotonicity

# Pattern Monotonicity and Its Roles

- A constraint  $c$  is *monotone*: If an itemset  $S$  **satisfies** the constraint  $c$ , so does any of its superset
  - That is, we do not need to check  $c$  in subsequent mining
- Ex. 1:  $c_1: \text{sum}(S.\text{Price}) \geq v$  is **monotone**
- Ex. 2:  $c_2: \text{min}(S.\text{Price}) \leq v$  is **monotone**
- Ex. 3:  $c_3: \text{range}(S.\text{profit}) \geq 15$  is **monotone**
  - Itemset  $ab$  satisfies  $c_3$
  - So does every superset of  $ab$

TID	Transaction	Item	Profit
10	a, b, c, d, f, h	a	40
20	b, c, d, f, g, h	b	0
30	b, c, d, f, g	c	-20
40	a, c, e, f, g	d	-15
min_sup = 2 price(item)>0		e	-30
		f	-10
		g	20
		h	5



The background of the slide is a complex, abstract composition. It features a network of thin, light-colored lines forming a mesh or web-like structure. Overlaid on this are various data visualization elements: a grid of small, light-colored plus signs, a series of small, colorful dots (green, blue, yellow) connected by lines, and a large, semi-transparent, light-colored shape that resembles a stylized letter 'A' or a large triangle. The overall color palette is muted, with shades of brown, beige, and light gray, accented by the colors of the data points.

# Constrained Mining with Data Anti-Monotonicity



# Data Space Pruning with Data Anti-Monotonicity

- A constraint  $c$  is **data anti-monotone**: In the mining process, if a data entry  $t$  cannot satisfy a pattern  $p$  under  $c$ ,  $t$  cannot satisfy  $p$ 's superset either

- Data space pruning: Data entry  $t$  can be pruned

- Ex. 1:  $c_1: \text{sum}(S.\text{Profit}) \geq v$  is **data anti-monotone**

- Let constraint  $c_1$  be:  $\text{sum}(S.\text{Profit}) \geq 25$

- $T_{30}: \{b, c, d, f, g\}$  can be removed since none of their combinations can make an  $S$  whose sum of the profit is  $\geq 25$

- Ex. 2:  $c_2: \min(S.\text{Price}) \leq v$  is **data anti-monotone**

- Consider  $v = 5$  but every item in a transaction, say  $T_{50}$ , has a price higher than 10

- Ex. 3:  $c_3: \text{range}(S.\text{Profit}) > 25$  is **data anti-monotone**

TID	Transaction	Item	Profit
10	a, b, c, d, f, h	a	40
20	b, c, d, f, g, h	b	0
30	b, c, d, f, g	c	-20
40	a, c, e, f, g	d	-15
min_sup = 2		e	-30
		f	-10
price(item) > 0		g	20
		h	5

# Data Space Pruning Should Be Explored Recursively

Example.  $c_3: \text{range}(S.\text{Profit}) > 25$

We check b's projected database

But item "a" is infrequent ( $\text{sup} = 1$ )

After removing "a (40)" from  $T_{10}$

$T_{10}$  cannot satisfy  $c_3$  any more

Since "b (0)" and "c (-20), d (-15), f (-10), h (5)"

By removing  $T_{10}$ , we can also prune "h" in  $T_{20}$

b's-proj. DB

TID	Transaction
10	a, c, d, f, h
20	c, d, f, g, h
30	c, d, f, g

TID	Transaction	Item	Profit
10	a, b, c, d, f, h	a	40
20	b, c, d, f, g, h	b	0
30	b, c, d, f, g	c	-20
		d	-15
40	a, c, e, f, g	e	-30
		f	-10
		g	20
		h	5

$\text{min\_sup} = 2$

$\text{price}(\text{item}) > 0$

Constraint:  
 $\text{range}\{S.\text{profit}\} > 25$

b's-proj. DB

TID	Transaction
10	<del>a, c, d, f, h</del>
20	c, d, f, g, <del>h</del>
30	c, d, f, g

Recursive  
Data  
Pruning

b's FP-tree

single branch: cdfg: 2

Only a single branch "cdfg: 2"  
to be mined in b's projected DB

Note:  $c_3$  prunes  $T_{10}$  effectively only after "a" is pruned (by min-sup) in b's projected DB

The background of the slide is a complex, abstract composition. It features a network of thin, reddish-brown lines connecting various points, creating a web-like structure. Interspersed among these lines are numerous small, colored dots in shades of green, blue, and orange. The overall aesthetic is technical and data-driven, with a focus on connectivity and spatial distribution. The text is centered within a white, angular shape that cuts across the middle of the image.

# Constrained Mining with Succinct Constraints



# Succinctness: Pruning Both Data and Pattern Spaces

---

- Succinctness: If the constraint  $c$  can be enforced by directly manipulating the data
- Ex. 1: To find those patterns without item  $i$ 
  - Remove  $i$  from DB and then mine (pattern space pruning)
- Ex. 2: To find those patterns containing item  $i$ 
  - Mine only  $i$ -projected DB (data space pruning)
- Ex. 3:  $c_3: \min(S.Price) \leq v$  is succinct
  - Start with only items whose price  $\leq v$  and remove transactions with high-price items only (pattern + data space pruning)
- Ex. 4:  $c_4: \sum(S.Price) \geq v$  is not succinct
  - It cannot be determined beforehand since sum of the price of itemset  $S$  keeps increasing

The background of the slide is a complex, abstract composition. It features a network of thin, light-colored lines forming a web-like structure. Overlaid on this are various data points and shapes: small green and blue dots, larger orange and red clusters, and a prominent white rectangular area in the center containing the title. The overall color palette is muted, with earthy tones and a touch of vibrant orange and red.

# Constrained Mining with Convertible Constraints

# Convertible Constraints: Ordering Data in Transactions

- Convert tough constraints into (anti-)monotone by proper ordering of items in transactions

- Examine  $c_1$ :  $\text{avg}(S.\text{profit}) > 20$

- Order items in value-descending order

- $\langle a, g, f, b, h, d, c, e \rangle$

- An itemset  $ab$  violates  $c_1$  ( $\text{avg}(ab) = 20$ )

- So does  $ab^*$  (i.e., *ab-projected DB*)

- $C_1$ : **anti-monotone** if patterns grow in the right order!

- Can item-reordering work for Apriori?

- Does not work for level-wise candidate generation!

- $\text{avg}(agf) = 23.3 > 20$ , but  $\text{avg}(gf) = 15 < 20$

		Item	Profit
		a	40
		b	0
		c	-20
		d	-15
		e	-30
		f	10
		g	20
		h	-5
TID	Transaction		
10	a, b, c, d, f, h		
20	b, c, d, f, g, h		
30	b, c, d, f, g		
40	a, c, e, f, g		

min\_sup = 2

price(item) > 0



The background of the slide is a complex, abstract composition. It features a network of red lines connecting various green and blue dots, resembling a graph or a molecular structure. This network is overlaid on a grid of small, light-colored plus signs. The overall color palette is muted, with shades of red, green, blue, and grey. A large, white, angular shape is positioned behind the text, creating a sense of depth and focus.

# Handling Multiple Constraints

# How to Handle Multiple Constraints?

---

- It is beneficial to use multiple constraints in pattern mining
- But different constraints may require potentially conflicting item-ordering
  - If there exists an order  $R$  making both  $c_1$  and  $c_2$  convertible, try to sort items in the order that benefits pruning most
  - If there exists conflict ordering between  $c_1$  and  $c_2$ 
    - Try to sort data and enforce *one constraint* first (which one?)
    - Then enforce the other when mining the projected databases
- Ex.  $c_1$ :  $\text{avg}(S.\text{profit}) > 20$ , and  $c_2$ :  $\text{avg}(S.\text{price}) < 50$ 
  - Sort in profit descending order and use  $c_1$  first (assuming  $c_1$  has more pruning power)
  - For each project DB, sort trans. in price ascending order and use  $c_2$  at mining



The background of the slide is a complex, abstract composition. It features a dark, reddish-brown base with a network of thin, light-colored lines forming a mesh or web-like structure. Scattered throughout are numerous small, colored dots in shades of green, blue, and orange. In the upper left, there is a horizontal band containing faint, stylized symbols and arrows. On the left side, there is a vertical rectangular area with a light blue border, containing a grid of small, colored squares and dots. The overall aesthetic is technical and data-driven.

# Constraint-Based Sequential- Pattern Mining



# Constraint-Based Sequential-Pattern Mining

---

- ❑ Share many similarities with constraint-based itemset mining
- ❑ **Anti-monotonic:** If  $S$  violates  $c$ , the super-sequences of  $S$  also violate  $c$ 
  - ❑  $\text{sum}(S.\text{price}) < 150; \min(S.\text{value}) > 10$
- ❑ **Monotonic:** If  $S$  satisfies  $c$ , the super-sequences of  $S$  also do so
  - ❑  $\text{element\_count}(S) > 5; S \supseteq \{\text{PC}, \text{digital\_camera}\}$
- ❑ **Data anti-monotonic:** If a sequence  $s_1$  with respect to  $S$  violates  $c_3$ ,  $s_1$  can be removed
  - ❑  $c_3: \text{sum}(S.\text{price}) \geq v$
- ❑ **Succinct:** Enforce constraint  $c$  by explicitly manipulating data
  - ❑  $S \supseteq \{\text{i-phone}, \text{MacAir}\}$
- ❑ **Convertible:** Projection based on the sorted value not sequence order
  - ❑  $\text{value\_avg}(S) < 25; \text{profit\_sum}(S) > 160$
  - ❑  $\text{max}(S)/\text{avg}(S) < 2; \text{median}(S) - \text{min}(S) > 5$


# Timing-Based Constraints in Seq.-Pattern Mining

---

- ❑ **Order constraint:** Some items must happen before the other
  - ❑  $\{\text{algebra, geometry}\} \rightarrow \{\text{calculus}\}$  (where “ $\rightarrow$ ” indicates ordering)
  - ❑ Anti-monotonic: Constraint-violating sub-patterns pruned
- ❑ **Min-gap/max-gap constraint:** Confines two elements in a pattern
  - ❑ E.g., mingap = 1, maxgap = 4
  - ❑ Succinct: Enforced directly during pattern growth
- ❑ **Max-span constraint:** Maximum allowed time difference between the 1<sup>st</sup> and the last elements in the pattern
  - ❑ E.g., maxspan (S) = 60 (days)
  - ❑ Succinct: Enforced directly when the 1<sup>st</sup> element is determined
- ❑ **Window size constraint:** Events in an element do not have to occur at the same time: Enforce max allowed time difference
  - ❑ E.g., window-size = 2: Various ways to merge events into elements

# Episodes and Episode Pattern Mining

---

- Episodes and regular expressions: Alternative to seq. patterns
  - Serial episodes:  $A \rightarrow B$
  - Parallel episodes:  $A \mid B$   Indicating partial order relationships
  - Regular expressions:  $(A \mid B)C^*(D \rightarrow E)$
- Methods for episode pattern mining
  - Variations of Apriori/GSP-like algorithms
  - Projection-based pattern growth
    - $Q_1$ : Can you work out the details?
  - $Q_2$ : What are the differences between mining episodes and constraint-based pattern mining?