Lecture 2. White- and black-box models. Causality

Regression models • Decision trees Itemsets mining • Discretization • Clustering • Numerical pattern mining • Tree ensembles • Neural networks

Tatiana Makhalova
MADE, 25.09.2021

Part I. Pattern mining

(or white-box models in unsupervised settings)

Interpretable models in unsupervised settings
Pattern mining

Goal: to discover a small set of non-redundant and interesting patterns that describe almost entirely the whole dataset

Main steps:

- discover the pattern search space (candidates to interesting patterns)

First of all, we need to select a pattern type. Then, all possible patterns of the chosen type make the pattern search space

- selection of the most interesting ones

We consider pattern mining in **binary** and **numerical** data

Pattern mining in binary data

Transactional dataset and its binary representation

t_1	rice, bread, milk
t_2	rice, pasta, milk
t_3	yogurt, juice, apples, pasta
t_4	chips, apples, rice
t_5	pasta, milk, apples
t_6	pasta, juice, bread, yogurt

	rice	bread	milk	pasta	yogurt	juice	apples	chips
t_1	X	X	X					
t_2	X		X	X				
t_3				X	X	X	X	
t_4	X						X	X
t_5			X	X				
t_6		X		X	X	X		

Itemset: any combination of binary attributes

Formal concepts (an intuitive introduction)

One way to restrict the pattern search space is to use the **maximal rectangles** in data filled with crosses

The rectangle is maximal if we cannot add neither column nor a row such that a new rectangle is filled entirely with the crosses

These maximal rectangles are called **formal concepts**

	rice	bread	milk	pasta	yogurt	juice	apples	chips
t_1	X	X	X					
t_2	X		X	X				
t_3				X	X	X	X	
t_4	X						X	X
t_5			X	X				
t_6		X		X	X	X		

Formal concept analysis

A formal context is a triple (*G, M, I*), where *G* is a set of objects, *M* is a set of attributes, and *I* is the incidence relation between two sets, i.e., $(g,m) \in I$ if object *g* has attribute *m*

	rice	bread	milk	pasta	yogurt	juice	apples	chips
t_1	x	x	x					
t_2	x		x	x				
t_3				x	x	x	x	
t_4	x						x	x
t_5			x	x				
t_6		x		x	x	x		

$A' = \{ m \in M \mid gIm \text{ for all } g \in A \}$

$B' = \{ g \in G \mid gIm \text{ for all } m \in B \}$

$\{t_3, t_6\}'$ = {pasta, yogurt, juice}   A' = B

{pasta, yogurt, juice}' = $\{t_3, t_6\}$ B' = A

Formal concept is a pair *(A,B)* where *A' = B*, and *B' = A.* B is called ***closed itemset***

Specificity: for noisy data the number of patterns may grows exponentially

Formal concept analysis. Closed itemsets

For an arbitrary set of attributes C the formal concept is computed as follows:

C’ = B

B’ = C’’ = A

For an arbitrary set of objects D the formal concept is computed as follows:

D’ = A

A’ = D’’ = B

In total there exists $O(2^{\min(|G|,\ |M|)})$ itemsets where *G* and *M* are the set of objects and attributes, respectively

For every formal concept (A,B) the following holds: A’’ = A, B’’ = B

Example. Computing formal concepts

Compute the formal concepts that contain:

1. *attributes*: bread and milk

   {bread, milk}’ = **{t1}**,
   {t1}’ = {bread, milk}’’ = **{rice, bread, milk}**

2. *attributes* milk and pasta

   {milk, pasta}’ = **{t2, t5}**,
   {t2, t5}’ = {milk, pasta}’’ = **{milk, pasta}**

3. *objects*: t1 and t4

   {t1, t4}’ = **{rise}**
   {rise}’ = {t1, t4}’’ = **{t1, t2, t4}**

4. *objects*: t3 and t4

   {t3, t4}’ = **{apples}**
   {apples}’ = {t3, t4}’’ = **{t3, t4}**

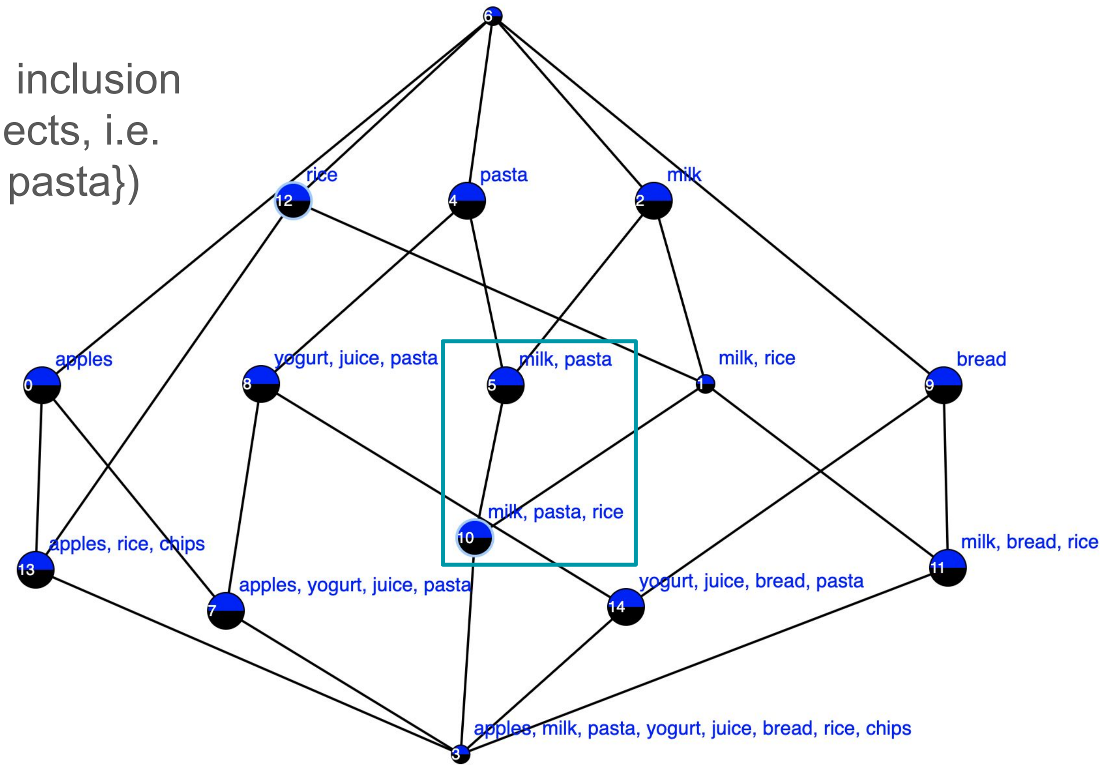
	rice	bread	milk	pasta	yogurt	juice	apples	chips
t1	X	X	X					
t2	X		X	X				
t3				X	X	X	X	
t4	X						X	X
t5			X	X				
t6		X		X	X	X		

Formal concept analysis. Concept lattice

All closed itemsets are ordered by inclusion w.r.t. the set of attributes or a set of objects, i.e. $(\{t_2, t_5\}, \{milk, pasta\}) > (\{t_2\}, \{rice, milk, pasta\})$

The **concept lattice** contains the whole set of formal concepts (closed itemsets)

If we choose closed itemsets as patterns, the concept lattice is the corresponding pattern search space



Tools: LatViz https://latviz.loria.fr/

Itemset mining

Goal: select a small set of interesting and non-redundant itemsets that cover almost entirely the whole dataset

Standard pipeline (pattern mining):

1. frequent (closed) itemset enumeration
2. application of an interestingness measure to each pattern individually

Drawbacks:

there are no rules of thumb for choosing a good threshold

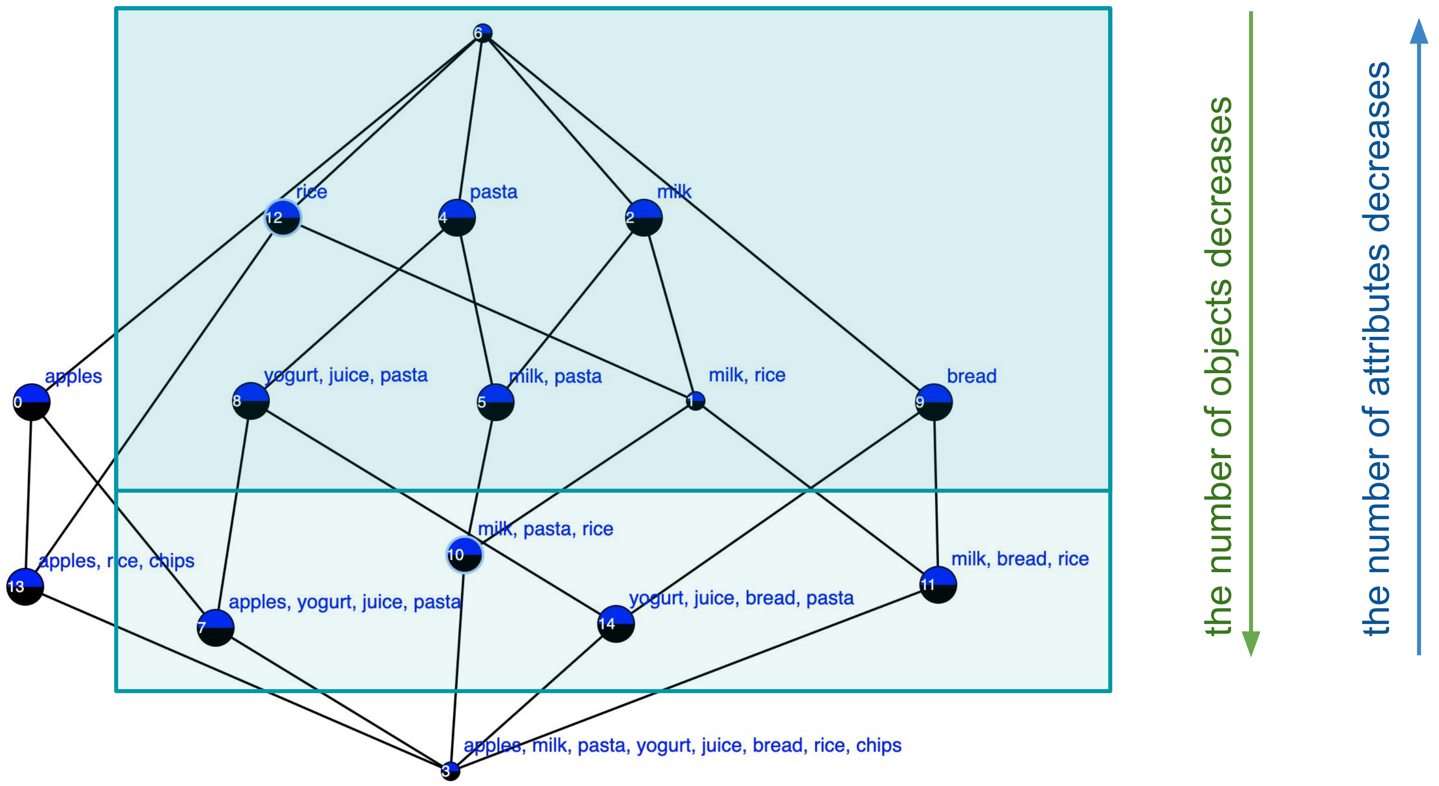
itemsets are highly redundant

cover is not regulated (usually)

itemsets are not interesting

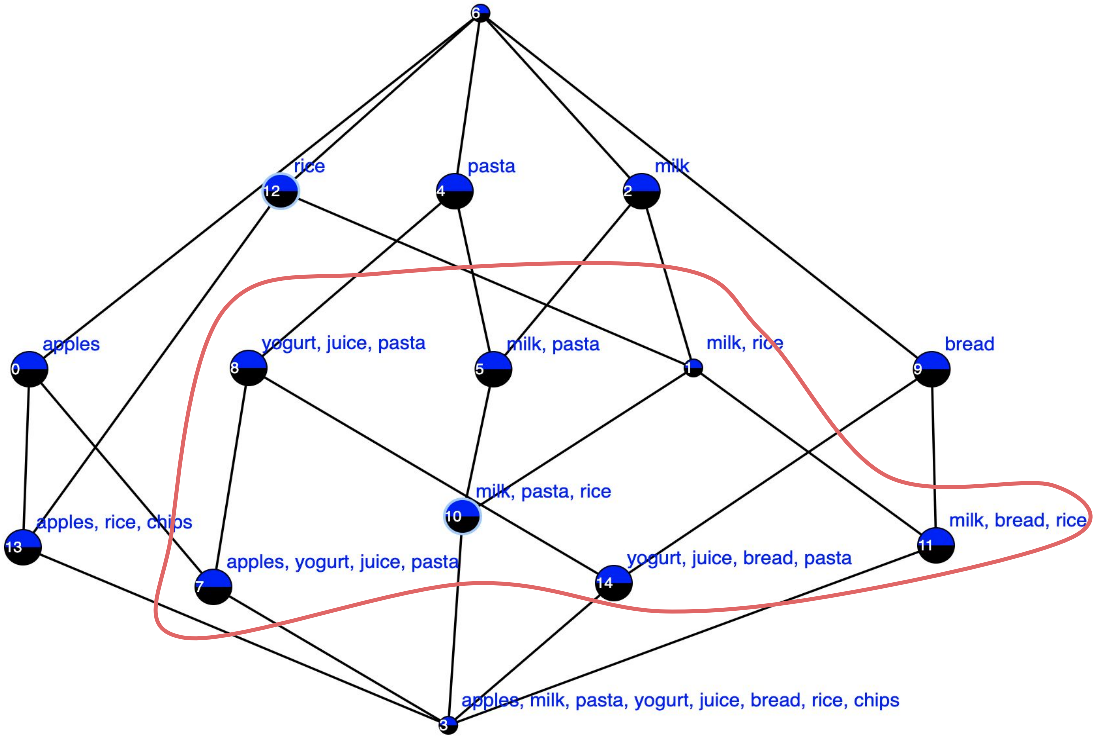
Kuznetsov, S. O., and Makhalova T. "On interestingness measures of formal concepts." *Information Sciences* 442 (2018): 202-219.

Enumeration of frequent closed itemsets



Tools: LatViz https://latviz.loria.fr/

Evaluation of almost the same itemsets



Given a set of enumerated itemsets, the problem with application of interestingness measures is the following:

being applied to almost the same itemsets we will see that **almost the same patterns** will obtain **almost the same interestingness scores =>**

very redundant set of interesting itemsets

Example of a set of itemsets

The first set of itemset: {rise, milk}, {bread}, {milk, pasta}, {apples}, {pasta, yogurt, juice}

- covers almost all crosses (cover rate is 17/19)
- is almost non-overlapping (overlapping rate is 18/17)

The second set of itemset: {rise}, {bread}, {milk}, {milk, pasta}, {pasta}, {pasta, yogurt, juice}, {rice, apples, chips}

- covers almost all crosses (cover rate is 18/19)
- is almost non-overlapping (overlapping rate is 25/18)

	rice	bread	milk	pasta	yogurt	juice	apples	chips
t_1	X	X	X					
t_2	X		X	X				
t_3				X	X	X	X	
t_4	X						X	X
t_5			X	X				
t_6		X		X	X	X		

	rice	bread	milk	pasta	yogurt	juice	apples	chips
t_1	X	X	X					
t_2	X		X	X				
t_3				X	X	X	X	
t_4	X						X	X
t_5			X	X				
t_6		X		X	X	X		

Cover rate = #crosses covered at least once / # total crosses
Overlapping rate = # total area of itemsets / # crosses covered at least once

Evolution of itemset mining

STATIC

MAKE AN ASSUMPTION

statistical hypotheses
interesting measures

MEASURE INFORMATIVENESS

entropy-based measures

DYNAMIC

REGULATE INFORMATIVENESS

measures with
a penalty term

Itemset mining

Goal: select a small set of interesting and non-redundant itemsets that cover almost entirely the whole dataset

Standard pipeline (pattern mining):

1. frequent (closed) itemset enumeration
2. application of an interestingness measure to each pattern individually

Drawbacks:

there are no rules of thumb for choosing a good threshold

itemsets are highly redundant

cover is not regulated (usually)

itemsets are not interesting

Kuznetsov, S. O., and Makhalova T. "On interestingness measures of formal concepts." *Information Sciences* 442 (2018): 202-219.

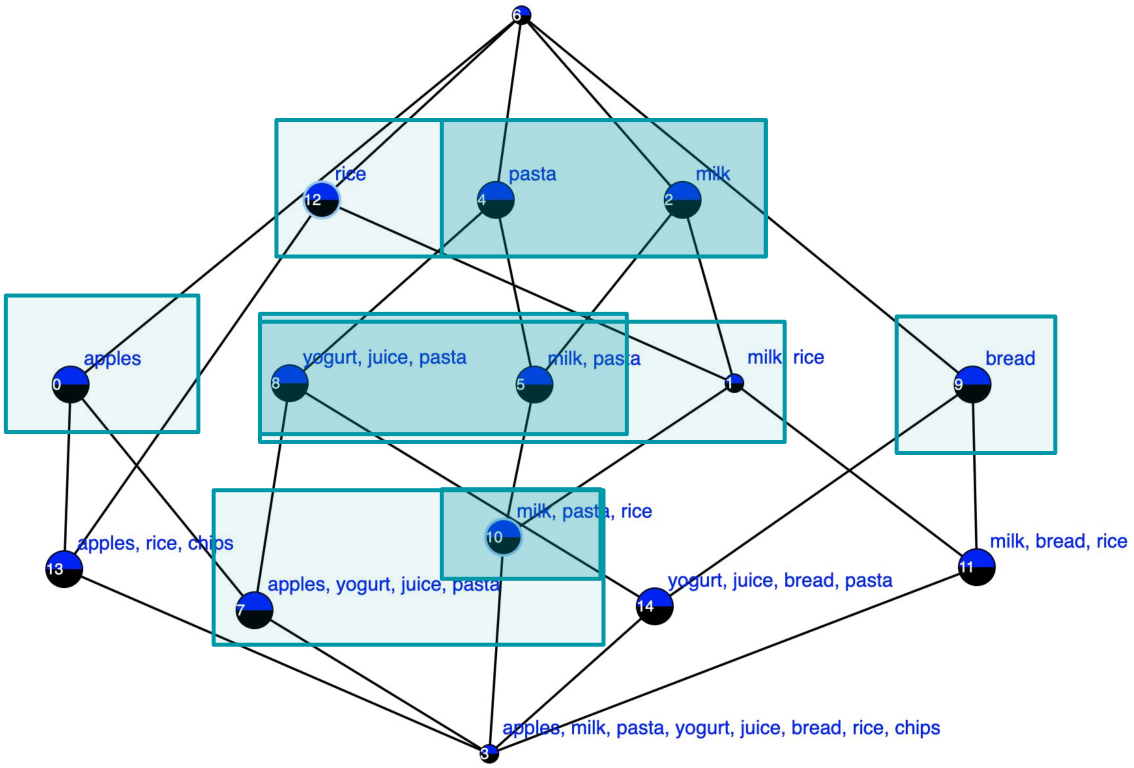
Itemset set mining

Goal: select a small set of interesting and non-redundant itemsets that cover almost entirely the whole dataset

Academic state-of-the-art pipeline (pattern set mining):

1. iterative itemset enumeration
2. application of a measure to a **set** of itemsets

Formal concept analysis. Concept lattice



Tools: LatViz https://latviz.loria.fr/

Minimum description length principle

MDL-based methods are often referred to as “**learning by compression**”

“Learning” ≈ “finding regularities”, thus the more regularities are found, the more the data can be compressed

In MDL the **total description length** is given by

$$L(D,H) = L(H) + L(D|H)$$

where $L(H)$ is the description length of the model (set of patterns) H, in bits, and $L(D|H)$ is the description length of the dataset D encoded with this set of patterns, in bits

Grünwald, Peter D. The minimum description length principle. MIT press, 2007

Minimum description length principle in pattern mining

Itemsets

- Vreeken, J., Van Leeuwen, M., and Siebes, A. (2011). Krimp: mining itemsets that compress. DMKD, 23(1):169–214.
- Smets, K. and Vreeken, J. (2012). Slim: Directly mining descriptive patterns. ICDM, 236–247. SIAM.

Tiles

- Tatti, N. and Vreeken, J. (2008). Finding good itemsets bypacking data. ICDM; 588–597. IEEE.52
- Tatti, N. and Vreeken, J. (2012). Discovering Descriptive Tile Trees – By Mining Optimal Geometric Subtiles. ECML PKDD, LNCS 7523, 9–24. Springer

Boolean matrix factorization

- Miettinen, P. and Vreeken, J. (2014). MDL4BMF: Minimum Description Length for Boolean Matrix Factorization. ACM TKDD,8(4):1–31.

Arbitrary-shaped patterns

- Faas, M. and van Leeuwen, M. (2020) Vouw: Geometric pattern mining using the MDL principle. International Symposium on Intelligent Data Analysis, 158–170. Springer.

Graphs

- Saran, D. and Vreeken, J. (2019). Summarizing dynamic graphs using MDL.

Hyper-rectangles

- Witteveen, J., Duivesteijn, W., Knobbe, A., and Grünwald, P.(2014). Realkrimp – finding hyper intervals that compress with MDL for real-valued data., International Symposium on Intelligent Data Analysis, 368–379. Springer.
- Makhalova, T., Kuznetsov, S. O., & Napoli, A. (2020). Mint: MDL-based approach for Mining INTeresting Numerical Pattern Sets. arXiv preprint arXiv:2011.14843

Discretization

- Nguyen, H.-V., Müller, E., Vreeken, J., and Böhm, K. (2014). Unsupervised interaction-preserving discretization of multivariate data. DMKD, 28(5-6):1366–1397.51

MDL in itemset mining. Slim algorithm

In practice, minimization is performed into **two stages:**

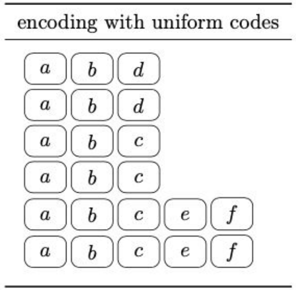
- computing pattern candidates
- selection those that minimize the total description length

Main components of the MDL-based approaches:

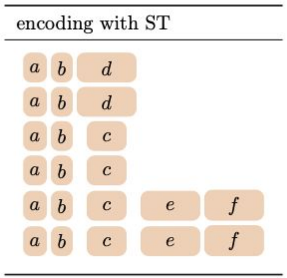
- encoding scheme
- itemset candidate enumeration
- total description length minimization

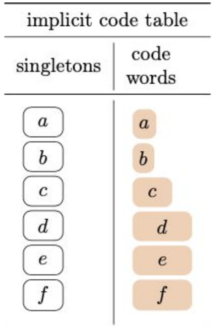
Slim. Encoding scheme

apple, bread, duck
apple, bread, duck
apple, bread, carrot
apple, bread, carrot
apple, bread, carrot, eggs, fish
apple, bread, carrot, eggs, fish



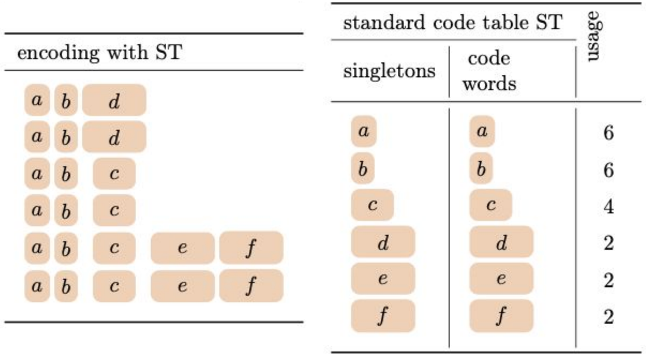
With uniform encoding, each item is encoded by the code words of uniform length





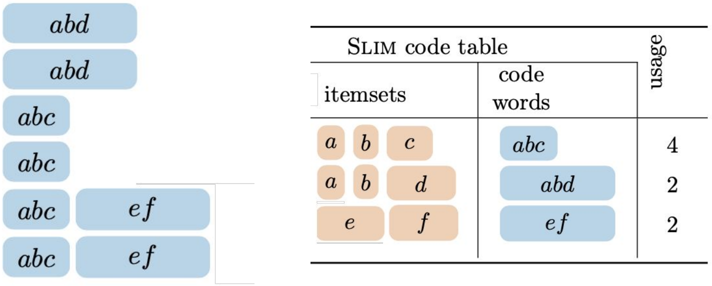
We can gain some bits by using the codewords of variable length (frequent items are encoded with shorter codewords). See Shannon codes for details. But we need to store a dictionary that contains the uniform codes and the associated code words of variable length

Slim. Encoding scheme



Applied to itemset mining, the simplest model is one that contains only singletons in its code table.

The goal is to find itemsets that ensure the shortest total description length



x variable-length code word associated with the singleton x

X variable-length code word associated with the itemset X

Slim. Encoding scheme

$L(D,CT) = L(CT) + L(D|CT)$

SLIM code table: itemsets	code words	usage
a b c	abc	4
a b d	abd	2
e f	ef	2

encoding with the SLIM code table

g_1	abd	
g_2	abd	
g_3	abc	
g_4	abc	
g_5	abc	ef
g_6	abc	ef

$$L(CT) = \underbrace{2 \cdot (l(a) + l(b)) + l(c) + l(d) + l(e) + l(f)}_{\substack{L(abd|ST)+L(abc|ST)+L(ef|ST) \\ \text{the left-hand column length}}} + \underbrace{L(code(abc)) + L(code(abd)) + L(code(ef))}_{\text{the right-hand column length}}.$$

$$L(D|CT) = 2 \cdot \Big(L(code(abd)) + L(code(ef)) \Big) + 4 \cdot L(code(abc)),$$

Slim. Candidate discovery by pairwise merging

Step 1	Candidates				
	b	c	d	e	f
a	**ab**	**ac**	**ad**	**ae**	**af**
b		**bc**	**bd**	**be**	**bf**
c			**cd**	**ce**	**cf**
d				**de**	**df**
e					**ef**
f					

Step 2	Candidates			
	c	d	e	f
ab	**abc**	**abd**	**abe**	**abf**
c		cd	ce	cf
d			de	df
e				ef
f				

Step 3	Candidates		
	ef	c	d
ab	**abef**	abc	abd
ef		**cef**	**cdef**
c			cd
d			

Step 4	Candidates	
	ef	d
abc	**abcdef**	**abcd**
ab	abef	abd
ef		def
d		

Step 5	Cand.
	ef
abc	abcdef
abd	
ef	

At each step, the candidates to the code table are computed by merging the members of a current code table.

In bold, the patterns appearing for the first time in the code table or in the candidate set are highlighted.

patterns from the code table at the current step, a set of candidates selected pattern

Association rule mining vs classification rule mining

transactions
apple, bread, duck
apple, bread, duck
apple, bread, carrot
apple, bread, carrot
apple, bread, carrot, eggs, fish
apple, bread, carrot, eggs, fish

d ➔ ab

e ➔ f

cf ➔ abe

ab ➔ c

ab ➔ d

d ➔ 0

c ➔ 1

cf ➔ 1

Association rule mining

If antecedent then consequent : A → C

- support(A ➔ C) = support(A ∪ C) = P(AC) ∈ [0,1]
- confidence(A ➔ C) = support(A ➔ C) / support(A) = P(C|A) ∈ [0,1]
- lift(A ➔ C) = confidence(A ➔ C) / support(C) = P(AC) / (P(A)P(C)) ∈ [0,inf)
- levarage(A ➔ C) = support(A ➔ C) - support(A)support(C) = P(AC) - P(A)P(C) ∈ [-1,1]

For mining association rules, the Mlxtend Python library can be used

Example. Computing the best rules

transactions
apple, bread, duck
apple, bread, duck
apple, bread, carrot
apple, bread, carrot
apple, bread, carrot, eggs, fish
apple, bread, carrot, eggs, fish

measures	c ➔ ef
support(c ➔ ef) = P(cef) confidence(c ➔ ef) = P(ef | c) lift(c ➔ ef) = P(cef) / (P(c)P(ef)) levarage(c ➔ ef) = P(cef) - P(c)P(ef)	2 / 6 2 / 4 2 * 6 / (2 * 4) = 12 / 8 2 / 6 - 4 / 6 * 2 / 6 = = 1 / 9

Reminder:

support(A ➔ C) = P(AC) ∈ [0,1]
confidence(A ➔ C) = P(AC) / P(A) = P(C|A)
lift(A ➔ C) = P(AC) / (P(A)P(C))
levarage(A ➔ C) = P(AC) - P(A)P(C)

Causal relations in association rules

Associations may not imply causal relationships!

A measure of the risk of experiencing the outcome under study when the antecedent factor is present is given by

$$\Omega_A = \frac{P(C|A)}{P(\overline{C}|A)}$$

Ω_A is the odds *A* that *B* will occur when *A* is present. The odds $\Omega_{\neg A}$ are defined similarly

$$\Omega_{\overline{A}} = \frac{P(C|\overline{A})}{P(\overline{C}|\overline{A})}$$

The odds ratio is given by

$$\omega_D = \frac{\Omega_A}{\Omega_{\overline{A}}} = \frac{supp(AC)supp(\overline{A}\overline{C})}{supp(\overline{A}C)supp(A\overline{C})}$$

Interpretation: $\omega_D = 1$ - exposure A does not affect odds of outcome C
$\omega_D > 1$ - exposure A associated with higher odds of outcome C
$\omega_D < 1$ - exposure A associated with lower odds of outcome C

Example. Computing the best rules

transactions
apple, bread, duck
apple, bread, duck
apple, bread, carrot
apple, bread, carrot
apple, bread, carrot, eggs, fish
apple, bread, carrot, eggs, fish

A ➔ C	c ➔ ef
$P(C \mid A) = P(AC)/P(A)$ $P(\neg C \mid A)$ $P(C \mid \neg A)$ $P(\neg C \mid \neg A)$ $\Omega_A = P(C \mid A) / P(\neg C \mid A)$ $\Omega_{\neg A} = P(C \mid \neg A) / P(\neg C \mid \neg A)$ $\omega = \Omega_A / \Omega_{\neg A}$	

Example. Computing the best rules

transactions
apple, bread, duck
apple, bread, duck
apple, bread, carrot
apple, bread, carrot
apple, bread, carrot, eggs, fish
apple, bread, carrot, eggs, fish

$A \rightarrow C$	$c \rightarrow ef$
$P(C \mid A) = P(AC)/P(A)$ $P(\neg C \mid A)$ $P(C \mid \neg A)$ $P(\neg C \mid \neg A)$ $\Omega_A = P(C \mid A) / P(\neg C \mid A)$ $\Omega_{\neg A} = P(C \mid \neg A) / P(\neg C \mid \neg A)$ $\omega = \Omega_A / \Omega_{\neg A}$	2 / 4 2 / 4 0 2 / 2 = 1 1 0 0

Simpson paradox

	Salary = *low*	Salary = *high*
Gender = m	185	120
Gender = f	65	60

$$\omega_D(m \to low) = \omega_D(w \to high) = \frac{185 \cdot 60}{65 \cdot 120} = 1.42$$

The odds ratio indicates a positive association between “Gender = m” and “Salary = low”

	Salary = *low*	Salary = *high*
Gender = m & College = y	5	20
Gender = f & College = y	15	40

$$\omega_{D_{college=y}}(m \to low) = \omega_D(w \to high)$$
$$= \frac{5 \cdot 40}{15 \cdot 20} = 0.66$$

	Salary = *low*	Salary = *high*
Gender = m & College = n	180	100
Gender = f & College = n	50	20

$$\omega_{D_{college=n}}(m \to low) = \omega_D(w \to high)$$
$$= \frac{180 \cdot 20}{50 \cdot 100} = 0.72$$

Pearl, Judea. "Models, reasoning and inference." *Cambridge, UK: CambridgeUniversityPress* 19 (2000).

Studying the causal relations

Studying the causal relations like A ➔ C, it is important to consider two subsets (groups), with A and without A, such that the instances within these two groups are the most **similar** by **relevant** to C characteristics

Difficulties:

- *Similarity*: What is similarity? Is there any threshold on similarity?
- *Relevance*: Which attributes are relevant for given C?

Example: “man” ➔ “low salary”

Similarity	*Relevance*
Can we consider the age of 25 and 30 years to be similar?	*Possible relevant characteristic*: age, education, domain
What about 25 and 35? 35 and 45?	*Irrelevant characteristics*: hair and eyes color, food preferences

Numerical pattern mining

Goal: to discover a small set of diverse and interesting patterns that cover almost entirely the whole dataset

Pattern is a **subset of the input feature space** that contain similar objects

The chosen type of patterns affect the interetability of the results. Patterns of “simple’’ shape may have a limited descriptiveness

Issues:

- Computationally **expensive search for splitting points** in each interval range (compare with a greedy dichotomous splitting of the decision trees)
- **Pre-discretization** is not always a good solution (in a few slides)
- **Potentially exponential number of patterns**, the approaches for a greedy search for good rules are underdeveloped

Pattern shape

Selecting a particular shape of patterns (pattern type) we may

- restrict the pattern search space
- improve the interpretability of the results

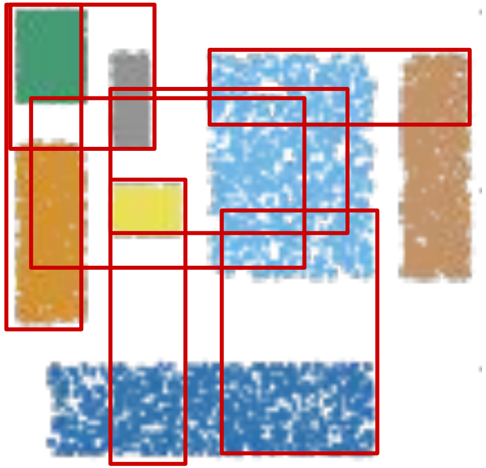
The best pattern shapes w.r.t. the aforementioned goals are

- spheres (are set by their center and radius)
- hyper-rectangles (are set by a tuple of intervals)

Pattern search space

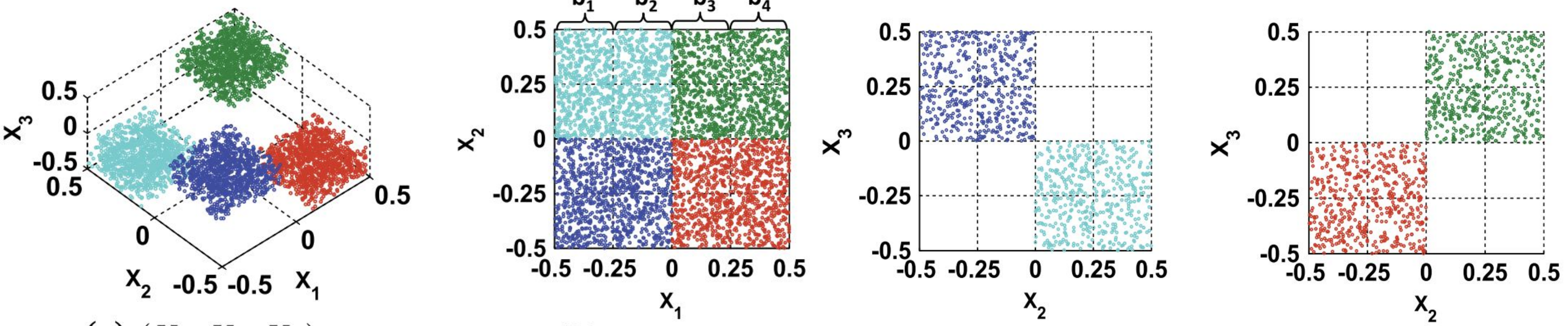
For hyper-rectangles the pattern search space is enormous

How to reduce the pattern search space?



Let’s **discretize** the search space! But how to do it **correctly**?

Attribute-wise discretization



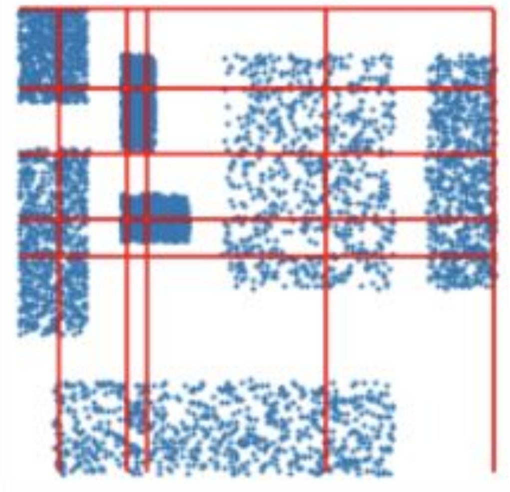
(a) (X_1, X_2, X_3) **(b)** (X_1, X_2) **(c)** $p(X_2, X_3 \mid b_1)$ and $p(X_2, X_3 \mid b_2)$ **(d)** $p(X_2, X_3 \mid b_3)$ and $p(X_2, X_3 \mid b_4)$

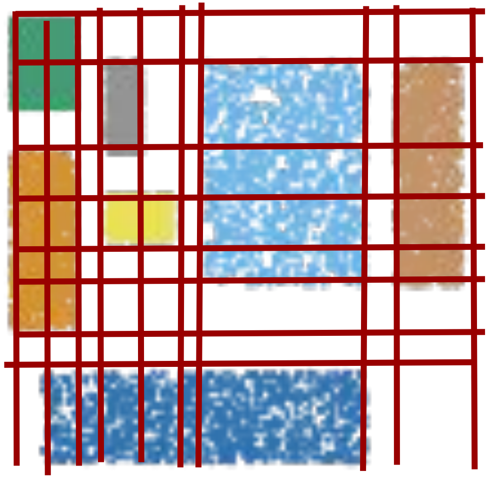
Attributes should be discretized w.r.t. other attributes to preserve interaction!

Nguyen, Hoang-Vu, et al. "Unsupervised interaction-preserving discretization of multivariate data." *Data Mining and Knowledge Discovery* 28.5 (2014): 1366-1397

Uniform vs pattern-specific discretization

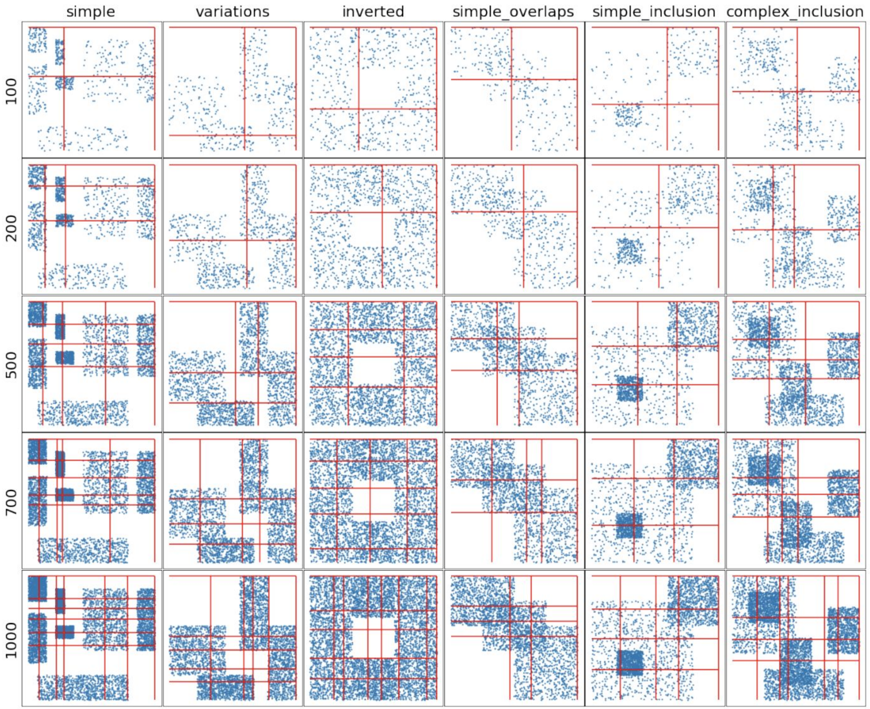
An “optimal” discretization is one that **preserves interactions** between features and put enough boundaries (not too much and not too little) in the feature space





Nguyen, Hoang-Vu, et al. "Unsupervised interaction-preserving discretization of multivariate data." *Data Mining and Knowledge Discovery* 28.5 (2014): 1366-1397.

Coarse vs pattern-specific discretization



Finding globally optimal interaction-preserving splitting does not allow to get patterns with precise boundaries

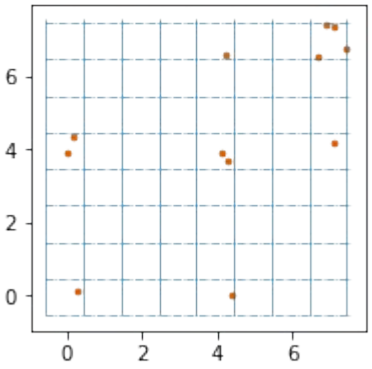
Makhalova, Tatiana, Sergei O. Kuznetsov, and Amedeo Napoli. "Mint: MDL-based approach for Mining INTeresting Numerical Pattern Sets." arXiv preprint arXiv:2011.14843 (2020).

Boundaries are computed by IPD, source code: https://people.mmci.uni-saarland.de/~jilles/prj/ipd/

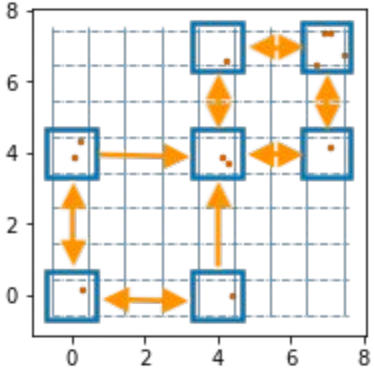
MDL for numerical pattern mining

- Use a fine-grained discretization
- Define an encoding scheme
- Gradually mine candidates and select those that ensure the best compression

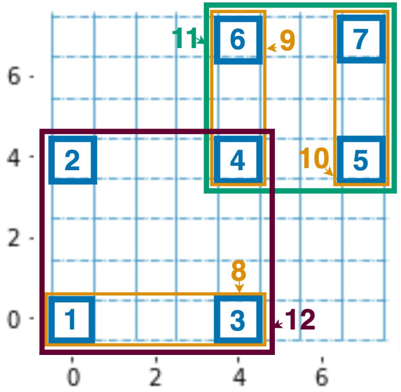
Example:



fine-grained discretization



a candidate is a smallest rectangle that contain two closest patterns



gradually select those candidates that ensure the best compression

Encoding of MINT

$$L(D, \mathcal{H}) = L(\mathcal{H}) + L(D|\mathcal{H})$$

$$L(\mathcal{H}) = \underbrace{L_{\mathbb{N}}(|M|) + \sum_{i=1}^{|M|} L_{\mathbb{N}}(|\mathcal{B}_i|)}_{length\ of\ the\ grid} + \underbrace{L_{\mathbb{N}}(|\mathcal{H}|) + |\mathcal{H}| \left(\sum_{i=1}^{|M|} \log\left(|\mathcal{B}_i|(|\mathcal{B}_i| + 1)/2\right) \right)}_{length\ of\ the\ pattern\ set}$$

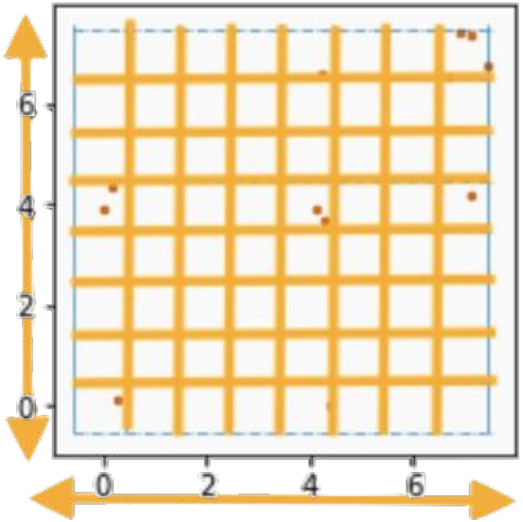
$$L(D|\mathcal{H}) = \underbrace{L_{\mathbb{N}}(|G|)}_{\substack{cost\ of\ encoding\ the\\ number\ of\ instances}} + \underbrace{L(D(\mathcal{H})|\mathcal{H})}_{\substack{cost\ of\ encoding\\ D\ with\ \mathcal{H}}} + \underbrace{L(D \ominus D(\mathcal{H})|\mathcal{H})}_{reconstruction\ cost}$$

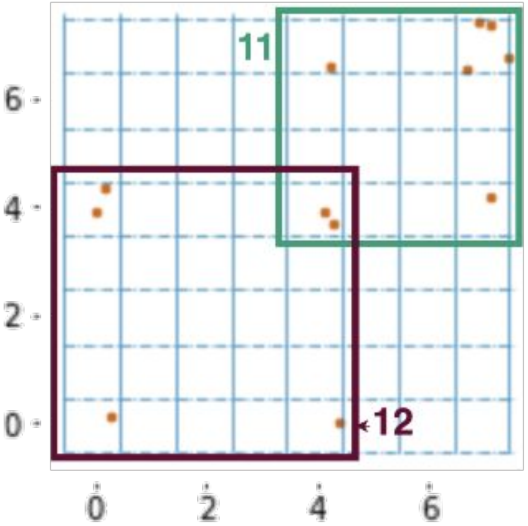
Makhalova, Tatiana, Sergei O. Kuznetsov, and Amedeo Napoli. "Mint: MDL-based approach for Mining INTeresting Numerical Pattern Sets." arXiv preprint arXiv:2011.14843 (2020).

Length of the model (pattern set)

$$L(\mathcal{H}) = L_{\mathbb{N}}(|M|) + \sum_{i=1}^{|M|} L_{\mathbb{N}}(|\mathcal{B}_i|) + \quad L_{\mathbb{N}}(|\mathcal{H}|) + |\mathcal{H}| \left(\sum_{i=1}^{|M|} \log\left(|\mathcal{B}_i|(|\mathcal{B}_i|+1)/2\right) \right)$$

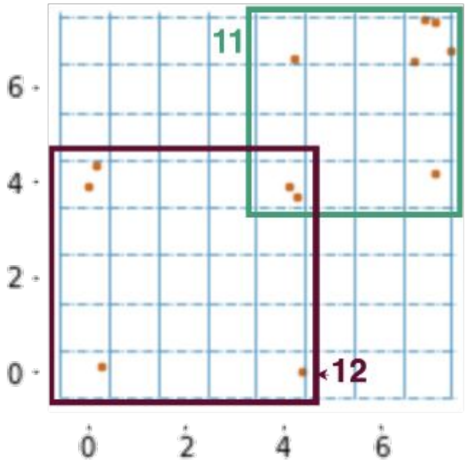
$$L(\mathcal{H}) = \underbrace{L_{\mathbb{N}}(2) + L_{\mathbb{N}}(8) + L_{\mathbb{N}}(8)}_{\textit{length of the grid}} + \quad \underbrace{L_{\mathbb{N}}(2) + 2\left(\log(36) + \log(36)\right)}_{\textit{length of the pattern set}}$$





Length of data

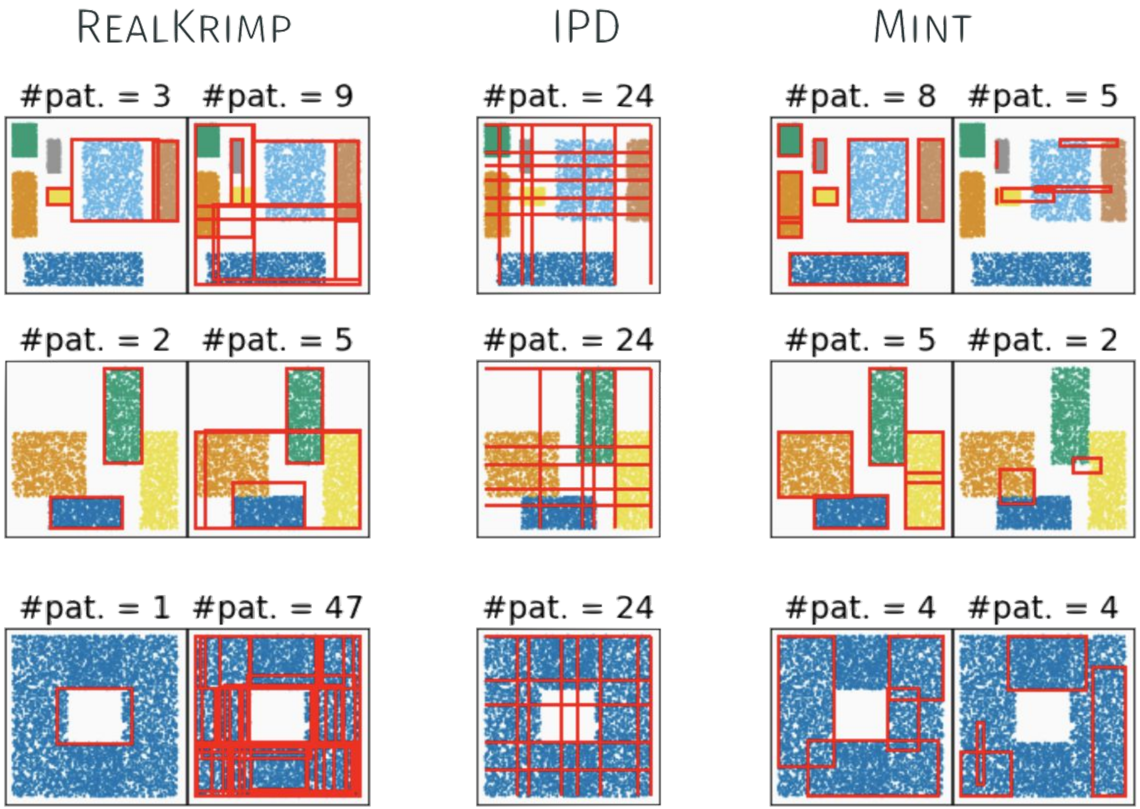
$$L(D|\mathcal{H}) = \underbrace{L_{\mathbb{N}}(12)}_{\text{cost of encoding the number of instances}} + \underbrace{L(D(\mathcal{H})|\mathcal{H})}_{\text{cost of encoding } D \text{ with } \mathcal{H}} + \underbrace{L(D \ominus D(\mathcal{H})|\mathcal{H})}_{\text{reconstruction cost}}$$



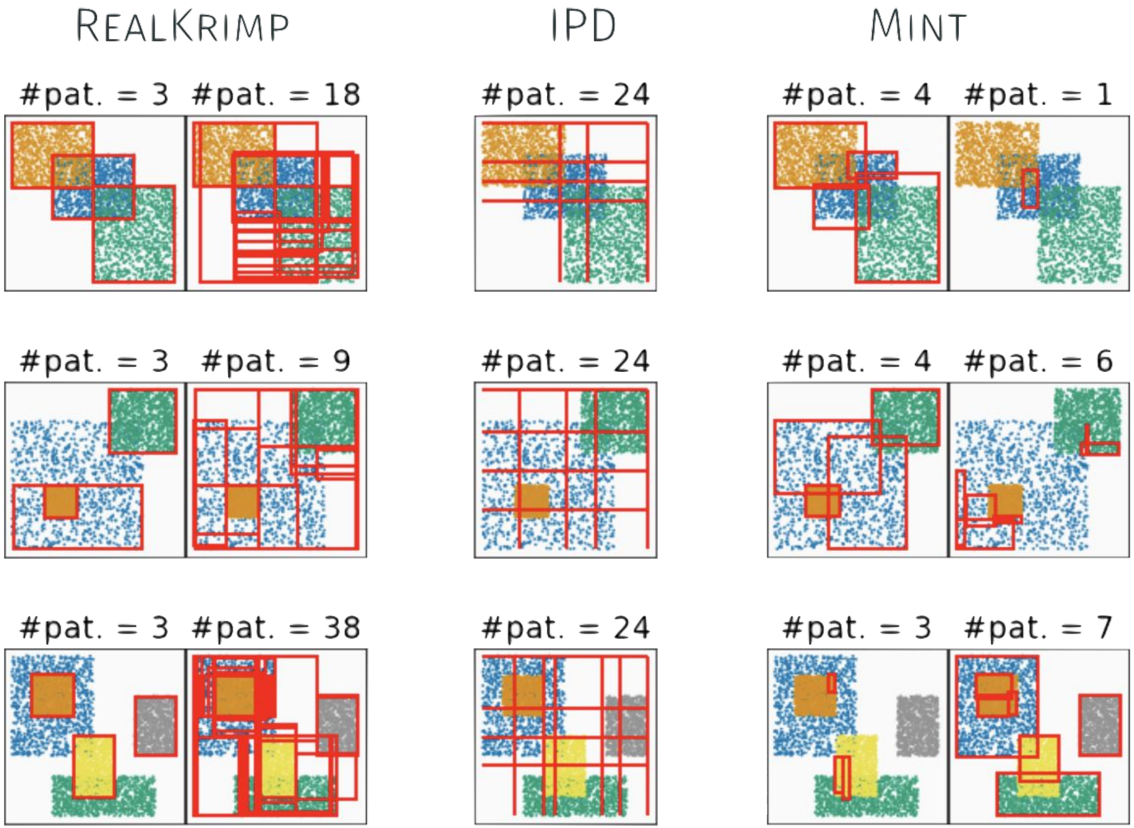
$usg(h_{11}) = 8$

$usg(h_{12}) = 4$

Comparison of pattern miners



Comparison of pattern miners



Mint: pros and cons

Pros:

- returns diverse and non-redundant patterns
- patterns have precise boundaries
- pattern descriptions are simple -> the results are interpretable

Cons:

- is based under the assumption on uniform distribution
- may be sensitive to noise
- requires a lot of memories
- does not contain “excluded” patterns in its language

Numerical pattern mining vs clustering

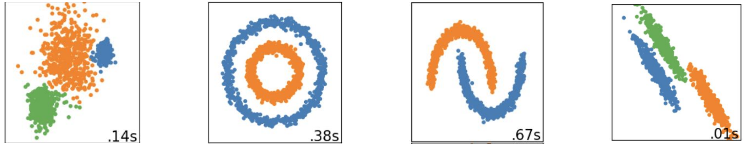
In pattern mining (PM) the **description** comes first, while in clustering the primacy is given to the **object similarity**

PM entails some requirements for the ease of **interpretation**, i.e., the resulting patterns should describe a region in the “attribute space” that is easy to interpret. In clustering, the focus is put on **groups** of objects or instances

The clusters can be constrained to have certain shapes, e.g., spheres in K-means or DBSCAN, but still the similarity of objects remains the most important characteristic of clusters

Numerical pattern mining vs clustering

Clustering is better when we need to identify groups of a complex shape but do not need to represent them in the attribute space



In the contrast to numerical pattern mining, clustering has the following **drawbacks**:

- selecting a number of clusters
- selecting a distance metric
- dealing with the curse of dimensionality
- uninterpretable description of patterns

https://scikit-learn.org/stable/modules/clustering.html#clustering

Wrapping up

- The state-of-the-art pattern mining approaches
  - discover pattern search space iteratively
  - evaluate pattern set instead of each pattern individually, e.g., using MDL
- Association rules (antecedent ➔ consequent)
  - do not say anything about causal relations
  - can be considered as classification rules (if consequent is a class label)
- Pre-discretization is not a good idea (should be avoided if possible)