# Organizational matters

- Remember to register for final exam in HISPOS
- Lecture on 27 November is **cancelled**
  - Schedule is pushed one week down
  - The DL for Topic IV's essay is still 12 February
    - Essay topics are given two weeks before

| Month | Day | Lecture topic | Essay |
|---|---|---|---|
| October | 16 | Intro | Warm-up essay |
| | 23 | T I intro: Pattern set mining | |
| | 30 | T I.1: Tiling | **Warm-up essay DL** |
| November | 6 | T I.2: MDL-based itemset mining | T I essay, w-u feedback |
| | 13 | T II intro: Graph mining | |
| | 20 | T II.1 | **T I essay DL** |
| | 27 | **No lecture** | |
| December | 4 | T II.2 | T II essay, T I feedback |
| | 11 | **No lecture** | |
| | 18 | T III intro: Assessing the significance | **T II essay DL** |
| | 25 | **No lecture, Christmas break** | |
| January | 1 | **No lecture, Christmas break** | |
| | 8 | T III.1 | T III essay, T II feedback |
| | 15 | T III.2 | |
| | 22 | T IV intro | **T III essay DL** |
| | 29 | T IV.1 | T IV essay, T III feedback |
| February | 5 | T IV.2 | |
| | 12 | | **T IV essay DL** |
| | 19 | Exam | |

# Topic I.1: Tiling Databases

Discrete Topics in Data Mining
Universität des Saarlandes, Saarbrücken
Winter Semester 2012/13

# T I.1 Tiling Databases

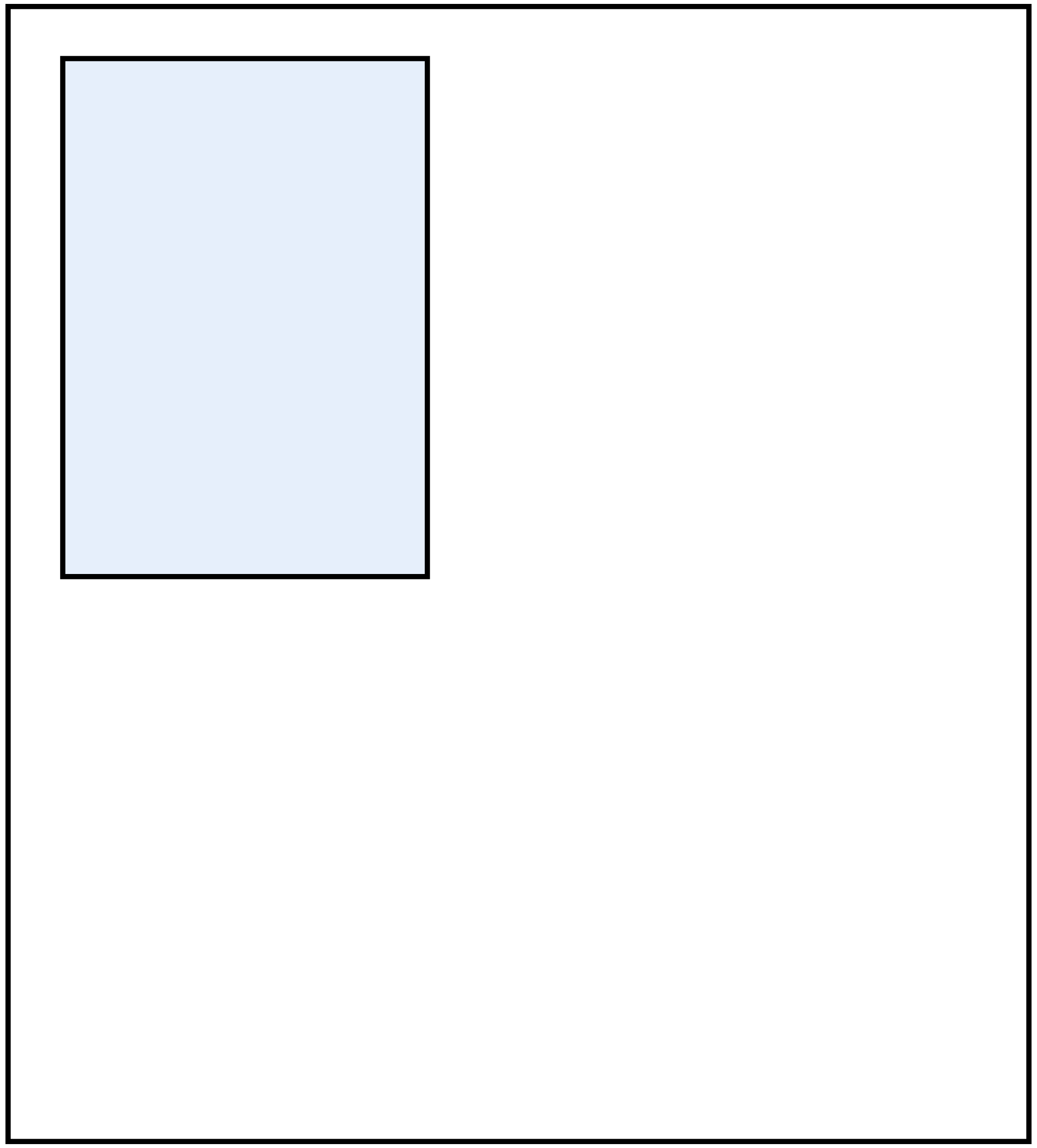

1. Background: Sets of Patterns
2. 0/1 Combinatorial Tiles
   2.1. What & Why
   2.2. The Set Cover Problem
   2.3. Finding the Tilings
3. Tiles as Density Estimates
   3.1. Combinatorial and Geometric Tiles
   3.2. An Algorithm for Finding Geometric Tiles
   3.3. A Bit of Art History

# Background: Sets of Patterns

- There are too many frequent itemsets and they contain repeated information
  - Every subset of a frequent itemset is a frequent itemset
- Closed, maximal, and non-derivable itemsets try to remove the redundancy in information
  - They might still yield to many almost-same itemsets
- **Tiling** addresses this problem by evaluating the **set** of itemsets with respect to the **data** they were found

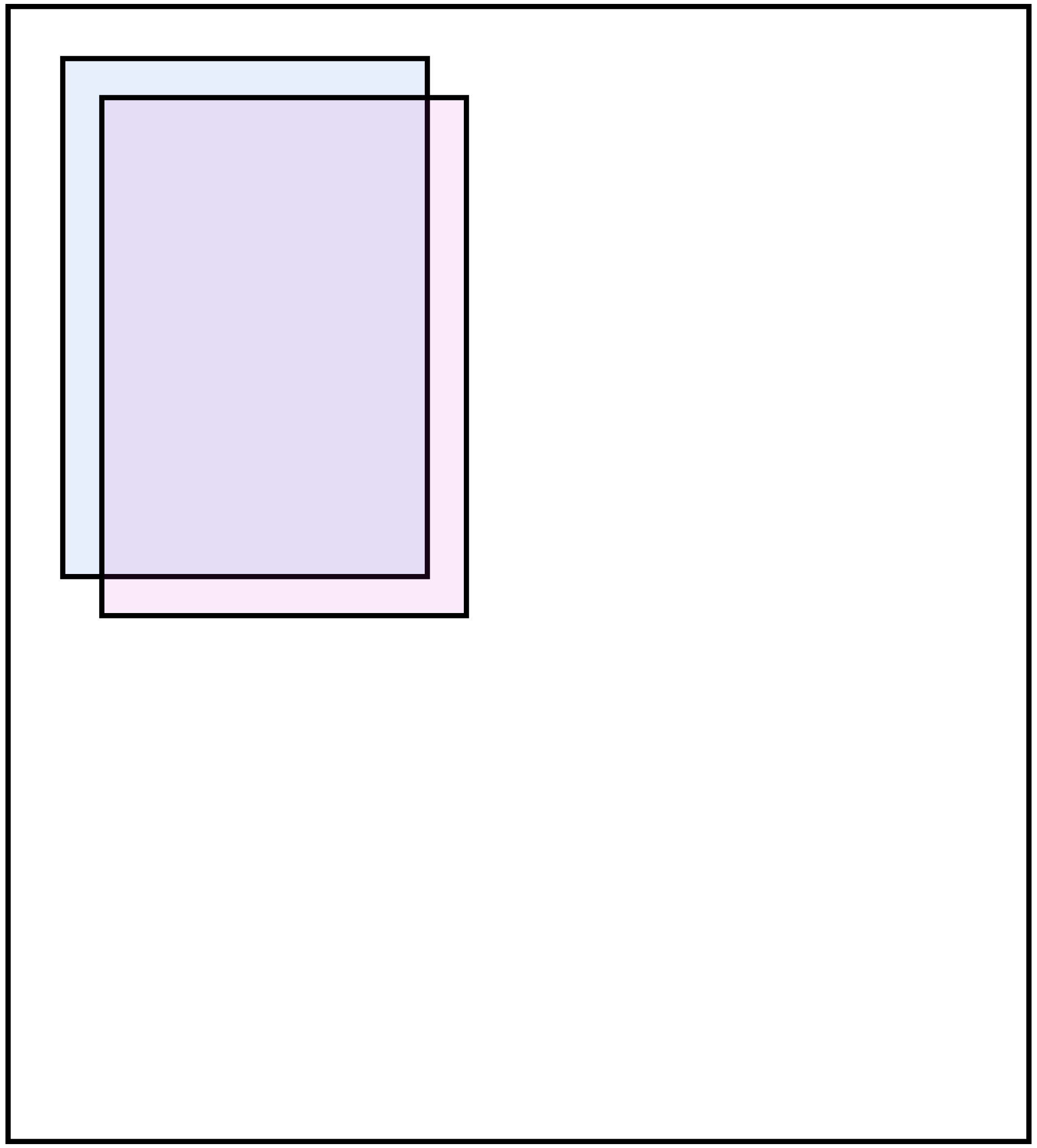
# Example

A frequent itemset



# Example

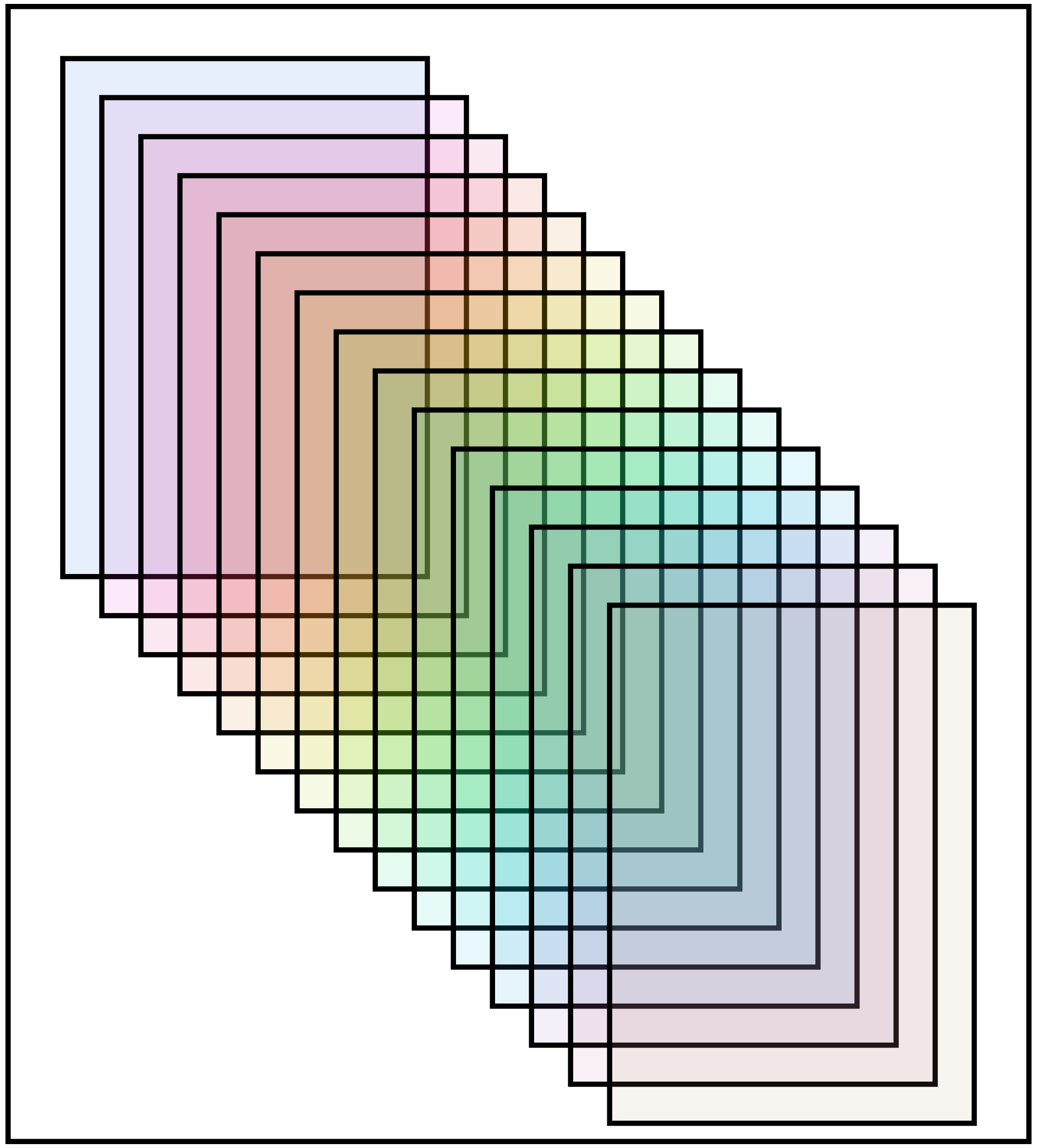
Both are closed (and possibly maximal)



# Example

All

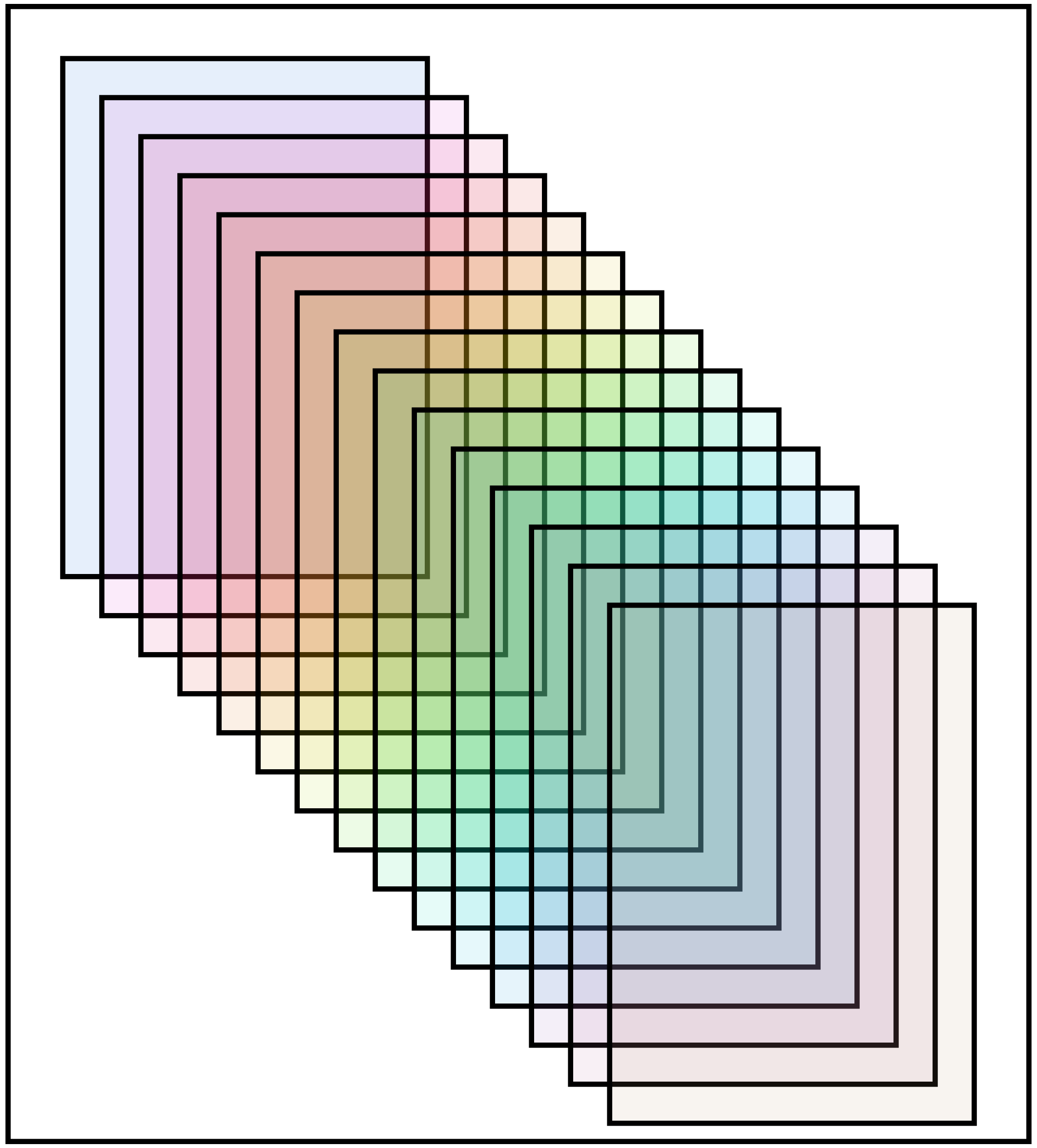
~~Both~~ are closed (and possibly maximal)



# Example

All

~~Both~~ are closed (and possibly maximal)

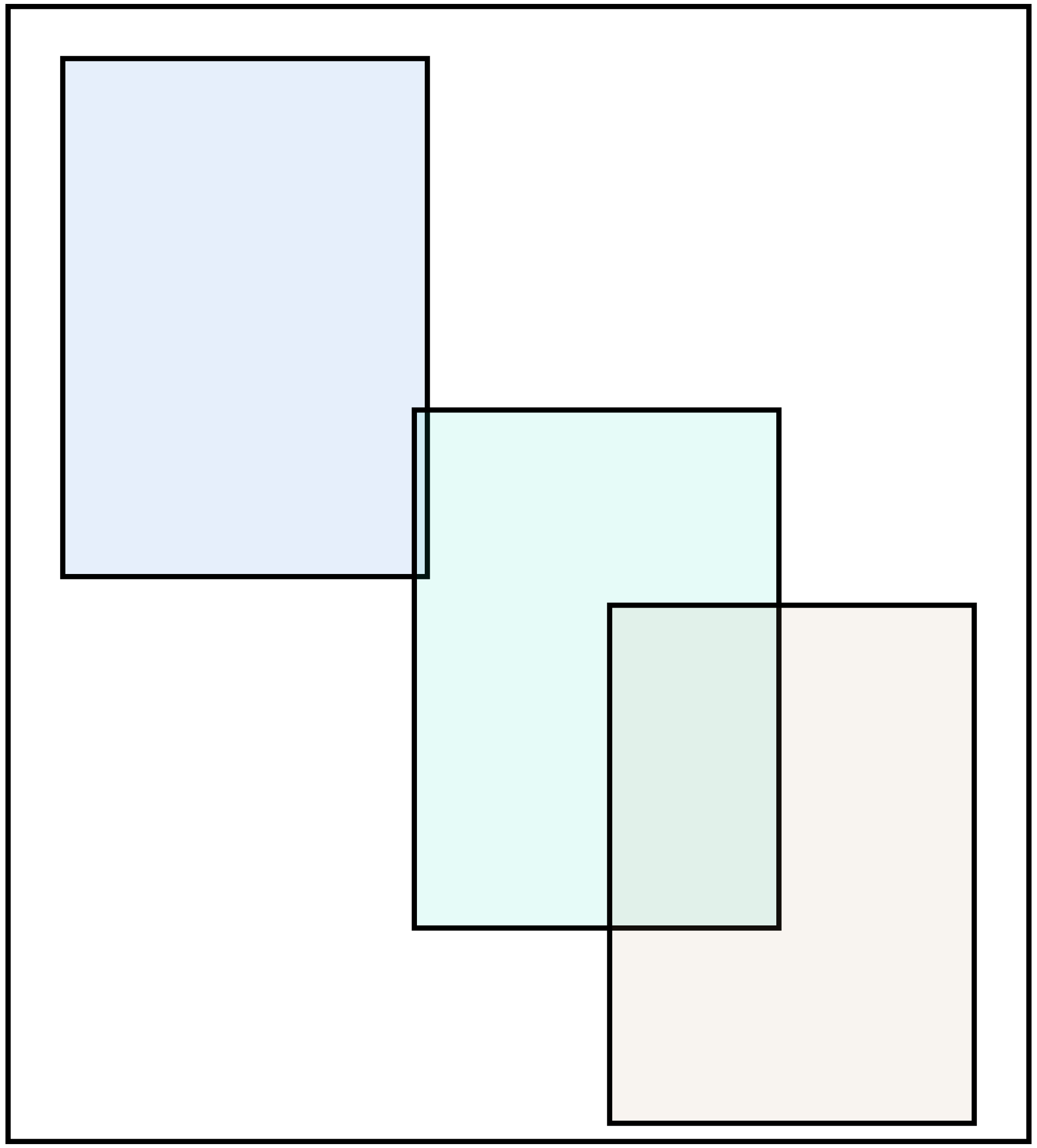


Perhaps we want to remove the redundancy

# Example

All

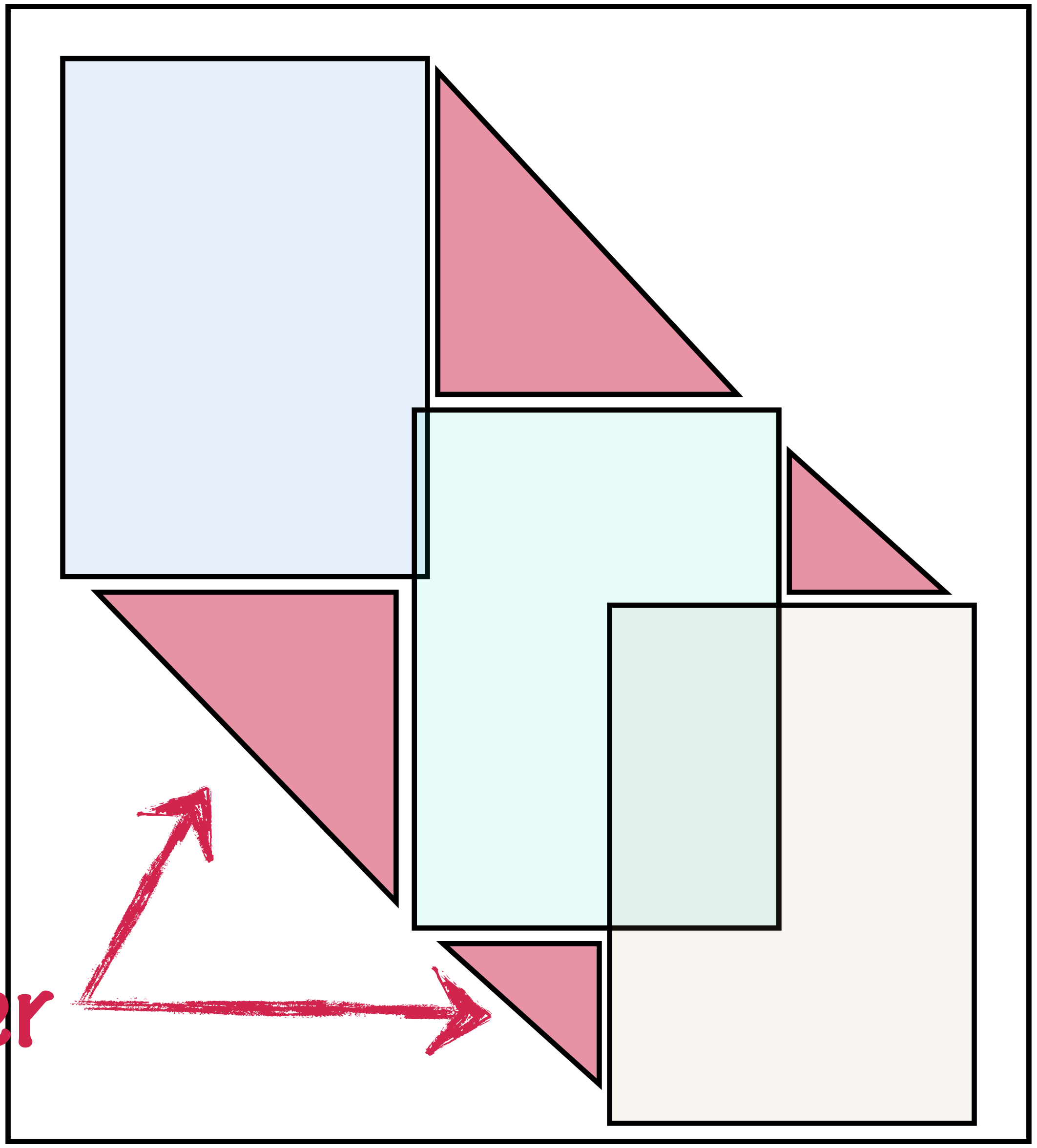
~~Both~~ are closed (and possibly maximal)



Perhaps we want to remove the redundancy

# Example

All
~~Both~~ are closed (and possibly maximal)



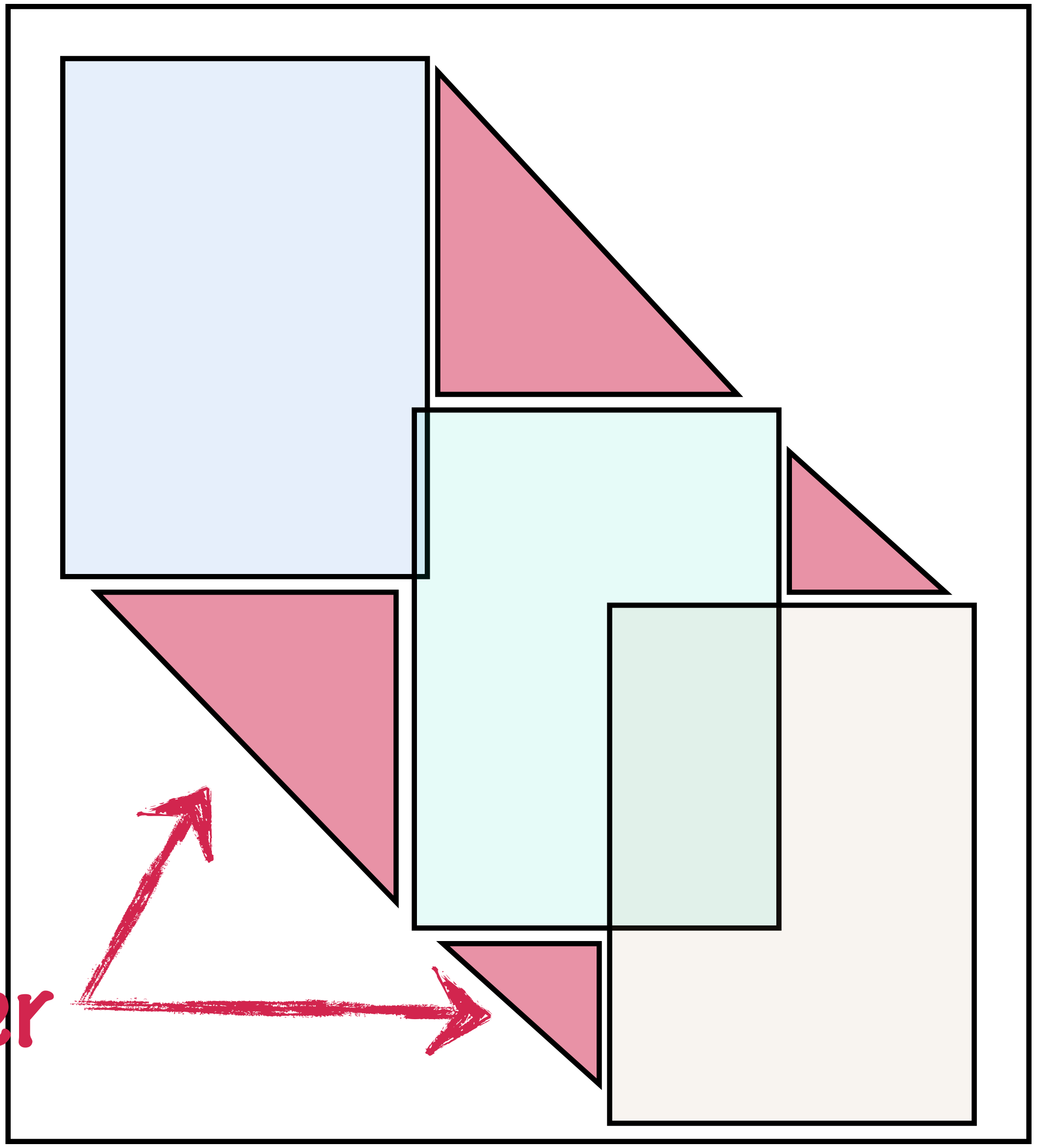
Perhaps we want to remove the redundancy

Area we don't cover

# Example

A rather good explanation of the full data

~~Both~~ All are closed (and possibly maximal)



Perhaps we want to remove the redundancy

Area we don't cover

# 0/1 Combinatorial Tiles

- Let $\boldsymbol{X}$ be an $n$-by-$m$ binary matrix (e.g. transaction data)
  - Let $\boldsymbol{r}$ be a $p$-dimensional vector of row indices ($1 \leq \boldsymbol{r}_i \leq n$)
  - Let $\boldsymbol{c}$ be a $q$-dimensional vector of column indices ($1 \leq \boldsymbol{c}_j \leq m$)
  - The $p$-by-$q$ *combinatorial submatrix* induced by $\boldsymbol{r}$ and $\boldsymbol{c}$ is

$$\mathbf{X}(\mathbf{r}, \mathbf{c}) = \begin{pmatrix} x_{r_1 c_1} & x_{r_1 c_2} & x_{r_1 c_3} & & x_{r_1 c_q} \\ x_{r_2 c_1} & x_{r_2 c_2} & x_{r_2 c_3} & \cdots & x_{r_2 c_q} \\ x_{r_3 c_1} & x_{r_3 c_2} & x_{r_3 c_3} & & x_{r_3 c_q} \\ & \vdots & & \ddots & \vdots \\ x_{r_p c_1} & x_{r_p c_2} & x_{r_p c_3} & \cdots & x_{r_p c_q} \end{pmatrix}$$

  - $\boldsymbol{X}(\boldsymbol{r},\boldsymbol{c})$ is *monochromatic* if all of its values have the same value (0 or 1 for binary matrices)
    - If $\boldsymbol{X}(\boldsymbol{r},\boldsymbol{c})$ is monochromatic 1, it (and $(\boldsymbol{r},\boldsymbol{c})$ pair) is called a **combinatorial tile**

Geerts, Goethals & Mielikäinen 2004

# Tiling problems

- **Minimum tiling.** Given $\boldsymbol{X}$, find the least number of tiles $(\boldsymbol{r},\boldsymbol{c})$ such that
  - For all $(i,j)$ s.t. $x_{ij} = 1$, there exists at least one pair $(\boldsymbol{r},\boldsymbol{c})$ such that $i \in \boldsymbol{r}$ and $j \in \boldsymbol{c}$ (i.e. $x_{ij} \in \boldsymbol{X}(\boldsymbol{r},\boldsymbol{c})$)
    - $i \in \boldsymbol{r}$ if exists $j$ s.t. $r_j = i$
- **Maximum $k$-tiling.** Given $\boldsymbol{X}$ and integer $k$, find $k$ tiles $(\boldsymbol{r}, \boldsymbol{c})$ such that
  - The number of elements $x_{ij} = 1$ that do belong in at least one $\boldsymbol{X}(\boldsymbol{r},\boldsymbol{c})$ is maximized

# Example

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |

# Example

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |

# Example

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |

# Example

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |

# Example

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |

# Example

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |

# Tiling and itemsets

- Each tile defines an itemset and a set of transactions where the itemset appears
  - Minimum tiling: each recorded transaction–item pair must appear in some tile
  - Maximum $k$-tiling: maximize the number of transaction–item pairs appearing on selected tiles
- Itemsets are local patterns, but tiling is global

# The Set Cover Problem

- A **set system** is a pair $(U, \mathcal{S})$, where $U$ (*universe*) is a (finite) set of elements and $\mathcal{S}$ a collection of subsets of $U$, $\mathcal{S} \subseteq 2^U$, such that $\bigcup_{S \in \mathcal{S}} S = U$
- **Set Cover.** Given a set system $(U, \mathcal{S})$, find the smallest subcollection $\mathcal{C} \subseteq \mathcal{S}$ such that $\bigcup_{C \in \mathcal{C}} C = U$
- **Max $k$-Cover.** Given $(U, \mathcal{S})$ and an integer $k$, find $k$ sets of $\mathcal{S}$ (in collection $\mathcal{C}$) such that $\left|\bigcup_{C \in \mathcal{C}} C\right|$ is maximized.

# Algorithm for Set Cover

**1. while** $U$ is not empty

**2.** Select the $S \in \mathcal{S}$ that has largest $|S \cap U|$

**3.** Add $S$ to $C$

**4.** Set $U \leftarrow U \setminus S$

**5. return** $C$

- This greedy algorithm achieves $\log(n)$ approximation for the Set Cover
  - This is best possible unless P = NP
- Stopping after $k$ sets gives e/(e – 1) approximation of Max $k$-Cover

# From Set Cover to Tiling

- We can use the set cover algorithm if we can reduce the tiling problem to a set covering problem
  - Let $\boldsymbol{X}$ be the 0/1 data matrix we want to tile
  - Let $U$ have one element for each 1 in $\boldsymbol{X}$, $U = \{u_{ij} : x_{ij} = 1\}$
  - Let $\mathcal{S}$ have one set for each *possible tile* in $\boldsymbol{X}$
    - For each $S \in \mathcal{S}$, we have row and column index vectors $\boldsymbol{r}$ and $\boldsymbol{c}$ such that $\boldsymbol{X}(\boldsymbol{r}, \boldsymbol{c})$ is monochromatic 1
    - Then $S = \{u_{ij} : i \in \boldsymbol{r} \text{ and } j \in \boldsymbol{c}\}$
- Now an optimum set covering gives us an optimum minimum tiling
  - Same for max $k$-covering and maximum $k$-tiling

# Job Done?

- The number of possible tiles is exponential with respect to the size of the data base
  - Generating the set system takes exponential time
  - Running the algorithm takes exponential time
  - And if I'm going to spend exponential time, I can as well just find the optimum solution
- How to solve this?
  - Reduce the number of tiles you consider
  - Find the tile to add without having to know all the tiles explicitly

# Reducing the Number of Tiles

- We don't need to consider *all* possible tiles
  - If $T_1$ and $T_2$ are tiles such that $T_1 \subset T_2$, we only need to consider $T_2$
  - We only need to consider *maximal* tiles (that are not subtiles of any other tile)
- Maximal tiles are those induced by closed itemsets
  - Adding new rows would require us to remove columns and vice versa
- But there still are (potentially) exponential number of closed itemset…

# Considering only Implicit Tiles

- Assume an oracle that, given a binary matrix and a tiling thereof, returns in polynomial time the tile that covers most of the 1s in the matrix *not yet covered by the given tiling*
  - If we have such oracle, we can execute the greedy algorithm in polynomial time
- If we don't have the oracle, but we can *approximate* the tile within some factor $R(n)$, we can approximate the set cover within $R(n)\log(n)$

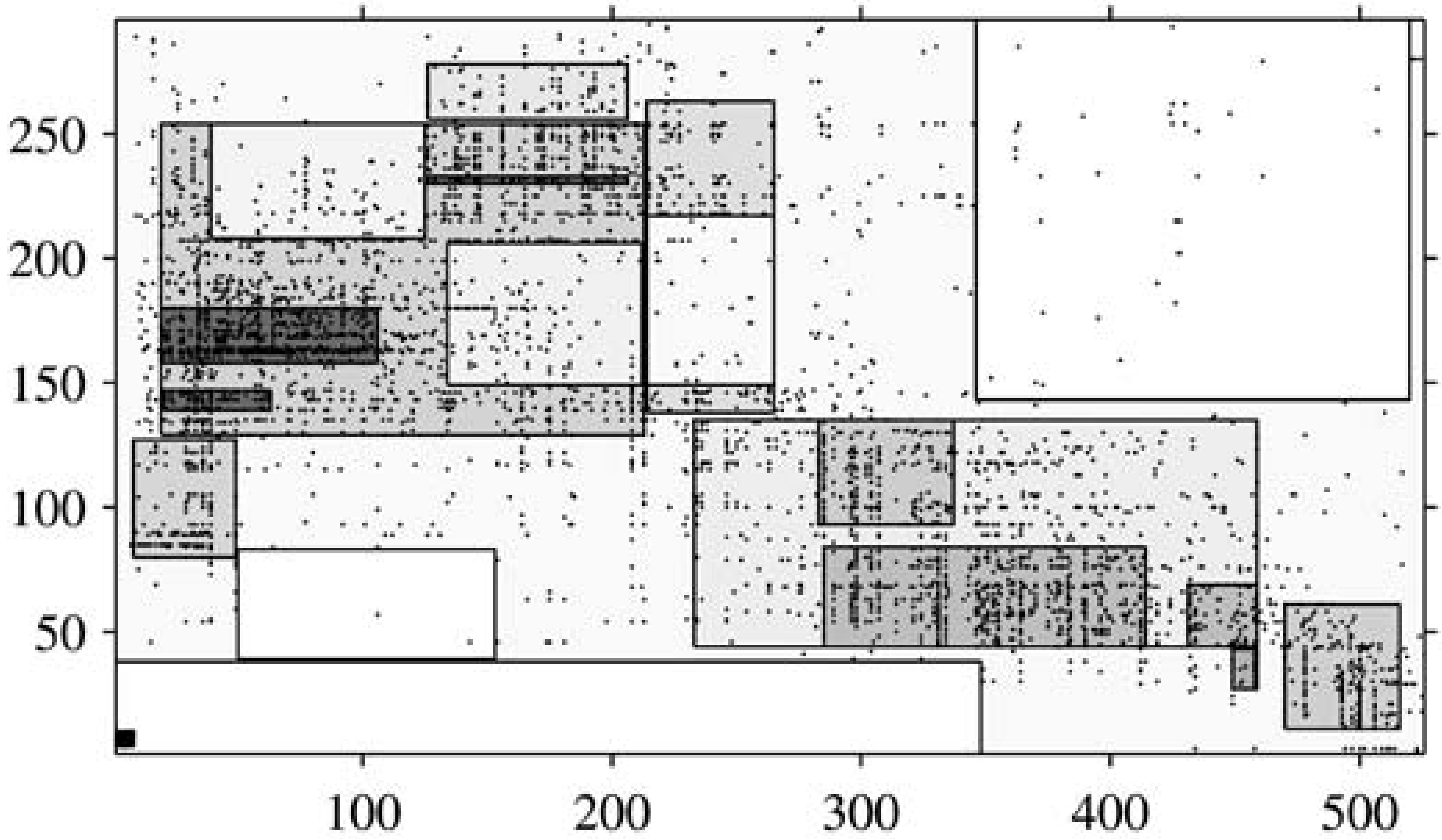
# A Practical Algorithm

- Replace the oracle with a large tile mining algorithm that takes into account the already-covered area
  - Finds only maximal tiles (closed itemsets)
  - Similar to ECLAT & CHARM
  - *Cannot* use downwards closedness property directly
    - Area of a tile is **not** downwards closed
  - Can still compute upper bounds on the maximum area of a super-tile of the given tile
  - Details left for reader
- Gives a practical algorithm for finding the minimum tiling and maximum $k$-tiling

# Tiles as Density Estimates

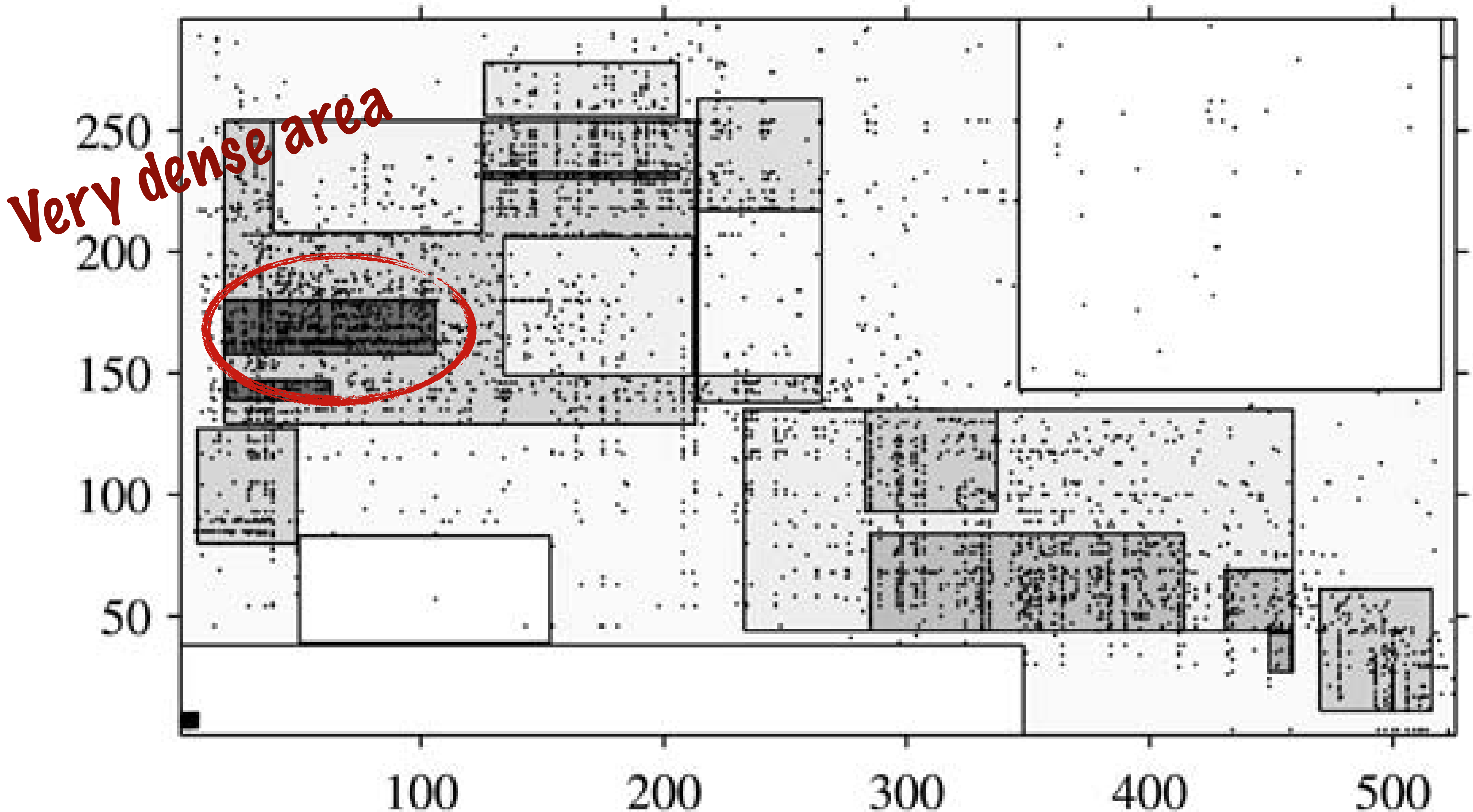
- A tile must be monochromatic 1
  - But real-world data often has noise
  - Noise breaks tiles
- Areas with lots of zeros can be interesting, as well
  - And areas of zeros within areas of ones
- We can consider tiles as areas of certain *density*
  - Density should be different in neighbouring areas
  - Within tiles, there can be sub-areas of different density
  - These are called **density tiles**
- Thus density tiles can be seen as density patterns in the data

# Example



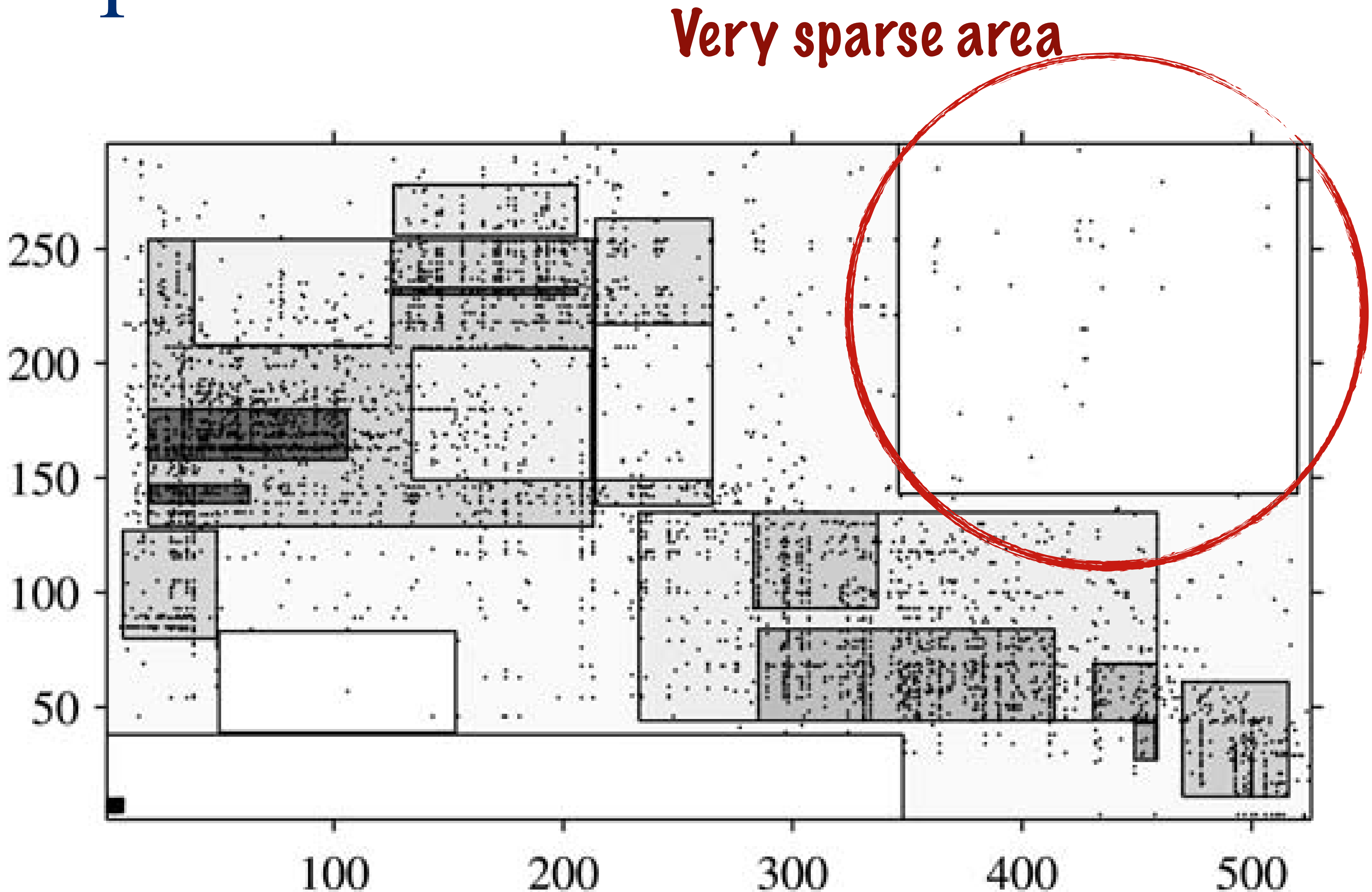

Gionis, Mannila & Seppänen 2004

# Example




Gionis, Mannila & Seppänen 2004

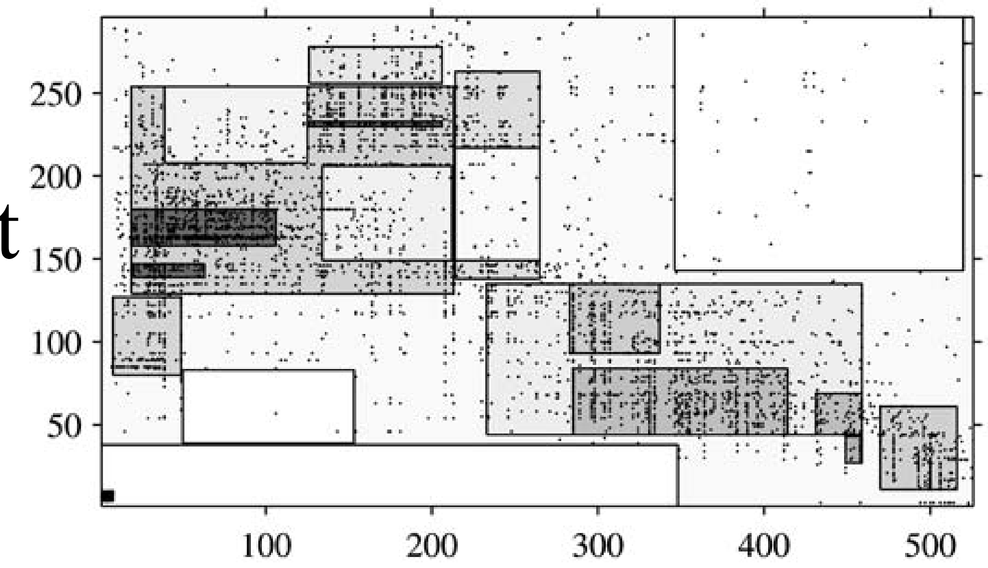
# Example




Gionis, Mannila & Seppänen 2004

# Geometric Tiles

- There are $2^n 2^m$ possible combinatorial submatrices in an $n$-by-$m$ matrix
  - If we look for density, we cannot look just monochromatic areas
- A **geometric (density) tile** is a tile with continuous row and column indices
  - It can be described given two corners
    - Or specific corner plus width and height
  - Only $n^2m^2$ possible
- We also allow a hierarchy of tiles
  - A sub-tile must be completely within its parent



# Mining the Geometric Density Tiles

- The goal for density tile mining is non-obvious
  - A single density tile can cover the whole data
  - What is the error induced by a tiling?
  - How many tiles? How many sub-tiles?
- General idea: use the tiling to give a *likelihood* of the data
  - Likelihood is the probability of the data given the density tiling
    - Zero on a dense tile is improbable, as is one on a sparse tile
- Bound the complexity using some model-order selection method

# The Likelihood of the Data

- Let $x_{ij}$ be an element of the data and $\tau$ a tile with density $p$
  - If $\tau$ has no sub-tile that covers $x_{ij}$, then the likelihood $q(\tau; i,j)$ of $x_{ij}$ is $p$
  - Otherwise, if $x_{ij} \in \tau' \subset \tau$, likelihood of $x_{ij}$ is computed with tile $\tau'$
    - Most specific tile defines the likelihood
- The likelihood of the whole data given $\tau$ is

$$L(\mathbf{X} \mid \tau) = \prod_{(i,j)\in\tau} q(\tau; i, j)^{x_{ij}} \left(1 - q(\tau; i, j)\right)^{1-x_{ij}}$$

  - The likelihood of the whole data is computed using a root tile

# How Many Tiles?

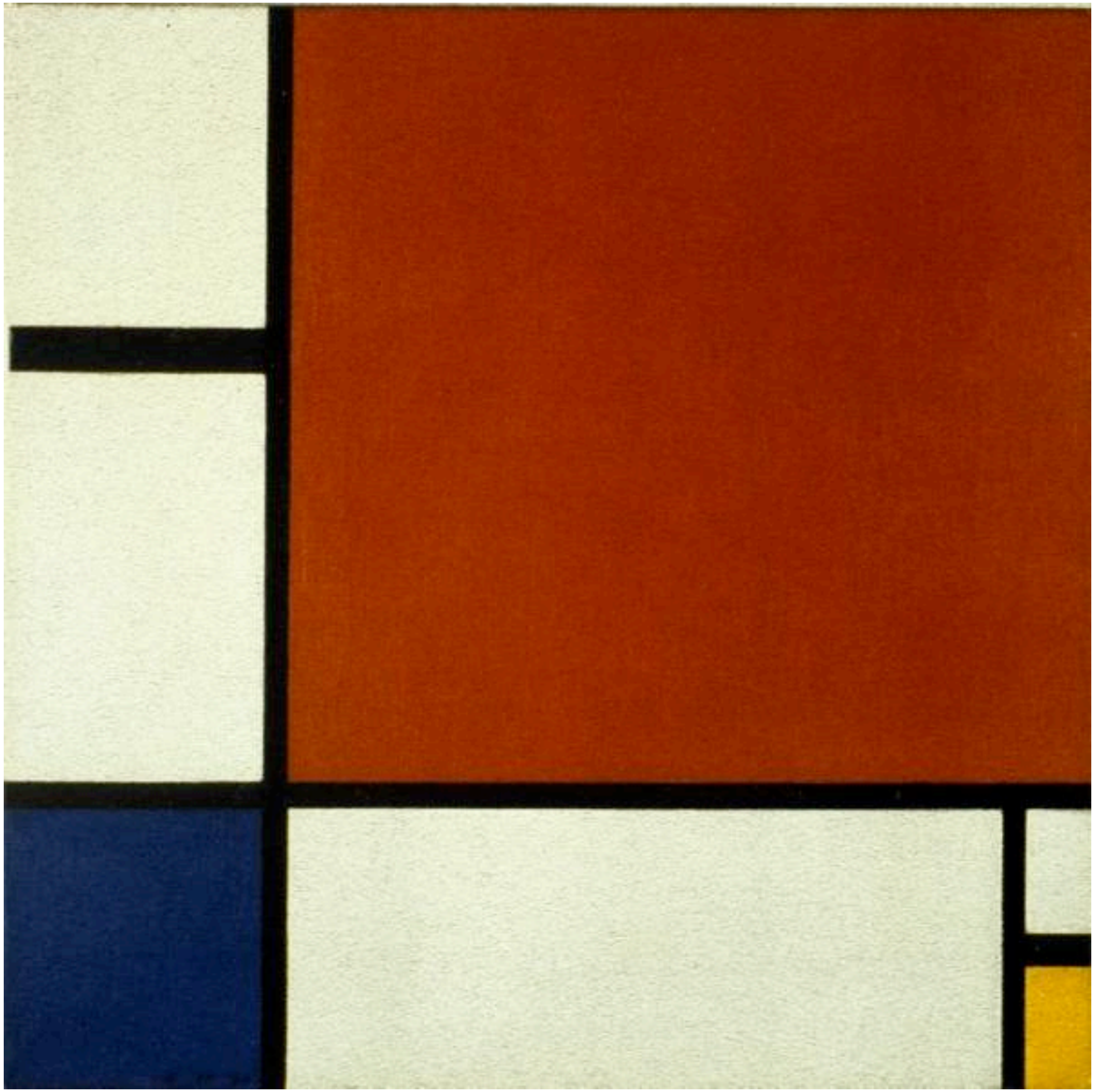
- We can get perfect likelihood
  - But the model would be too complex
- Balance between the complexity of the model and the likelihood
- For example, Bayesian Information Criterion (BIC)
  - Minimize $k \times \log(nm) - 2\log(L(\boldsymbol{X} \mid \tau))$
    - $k$ is the number of sub-tiles
  - The first part explains how complex tiling we have and the second part is twice the log likelihood

# How to Find Tilings

- Randomized greedy algorithm for one tile:
  - Draw a random rectangle $(a, b)\times(c, d) = \{(i, j) : a \leq i \leq b \text{ and } c \leq j \leq d\}$
  - Try to expand and shrink it to all directions
    - E.g. $(a, b)\times(c, d + 1)$, $(a, b)\times(c, d + 2)$, $(a, b)\times(c, d + 3)$, …
  - Out of all tried rectangles, select the one with highest likelihood
    - If this is better than the likelihood of the original rectangle, choose this as a new original rectangle, and start expanding and shrinking it
  - Stop when the likelihood cannot be improved using expansions or shrinks
- For tilings, find tiles one-by-one and stop when BIC stops decreasing

Gionis, Mannila & Seppänen 2004

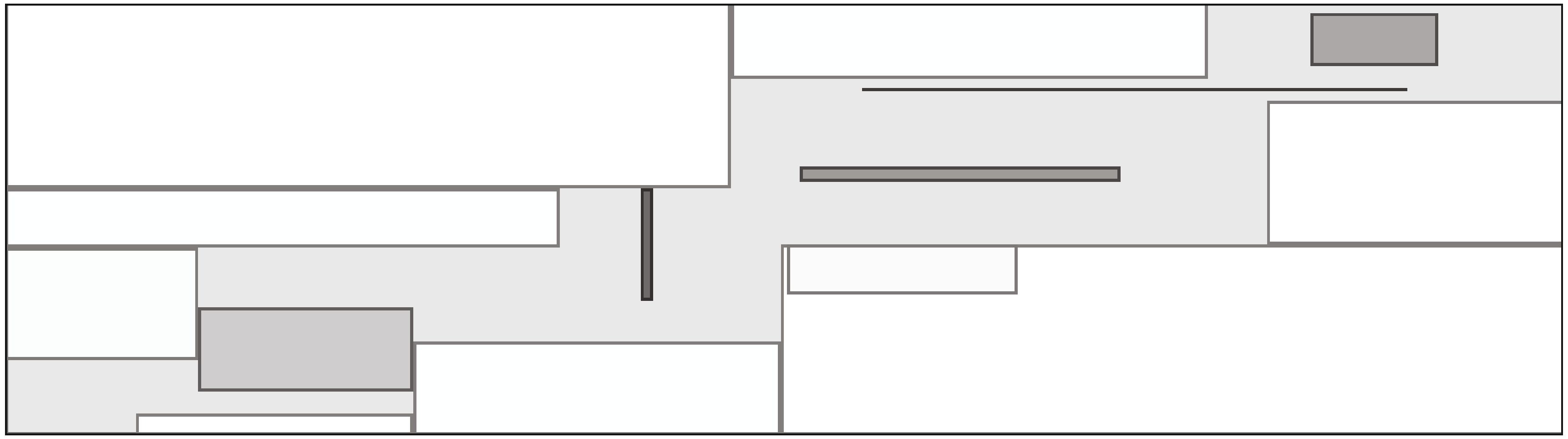
# Stijl – An Algorithm and a Movement



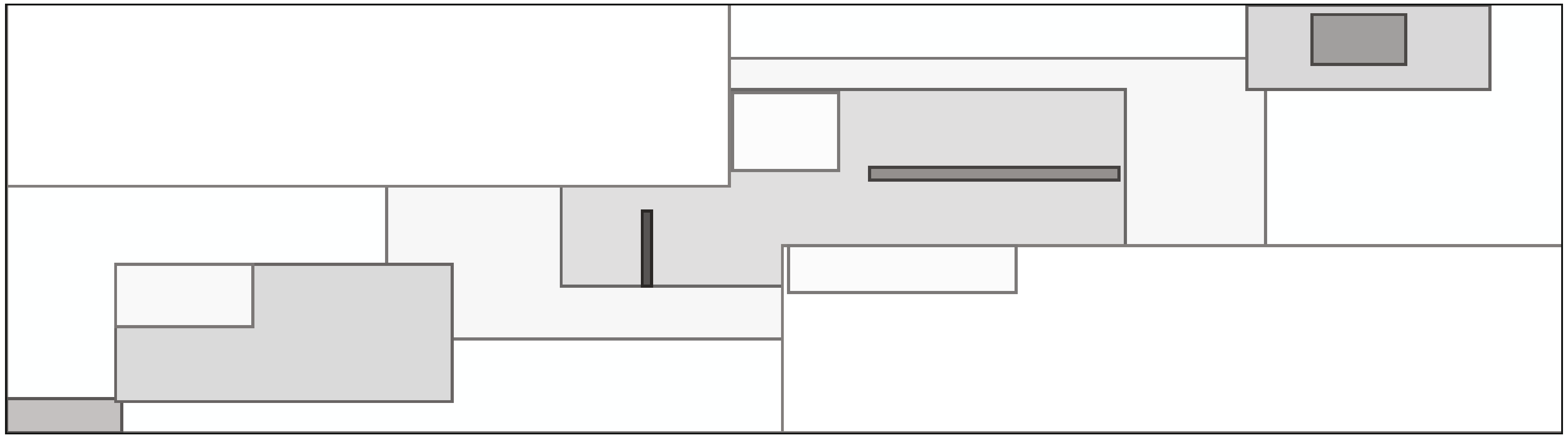
Piet Mondrian: *Composition II in Red, Blue, and Yellow*, 1930

# Tiles That Overlap Within Parents

No overlap

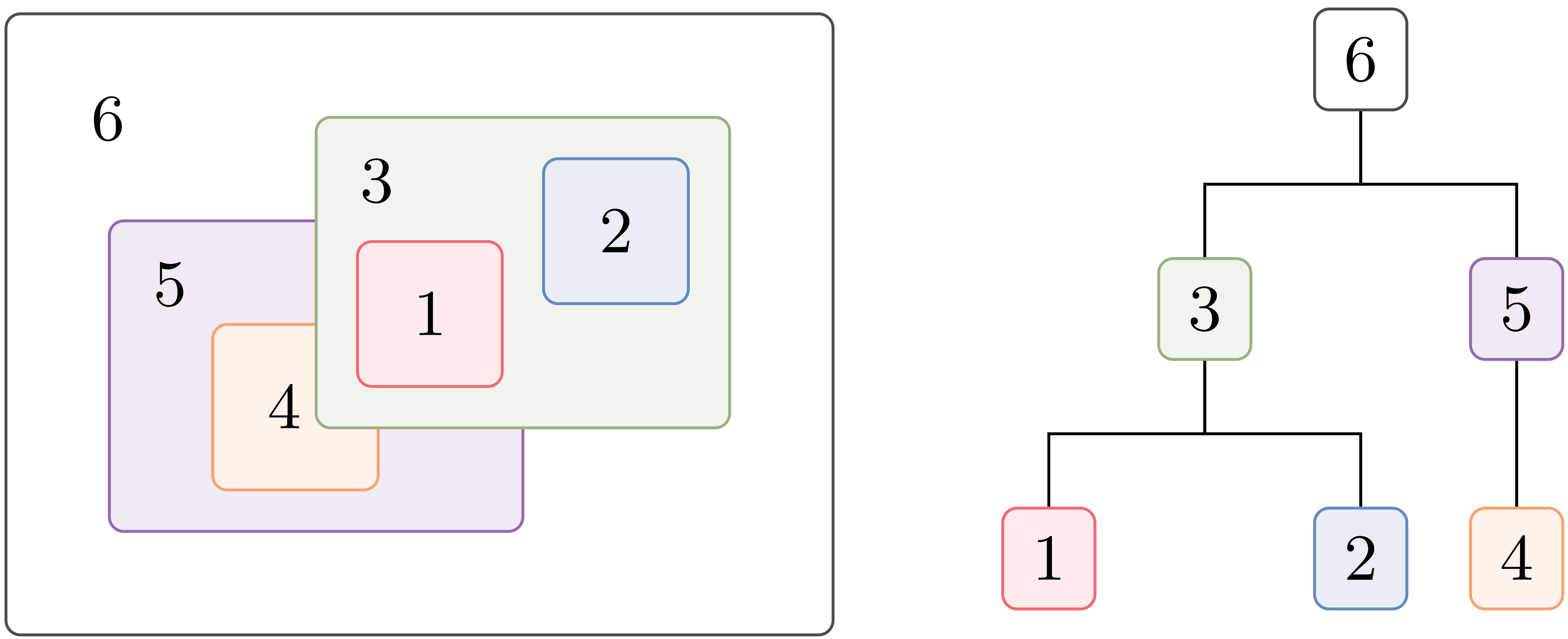


Overlap within parent



Tatti & Vreeken 2012

# Tile Trees




Tatti & Vreeken 2012

# The Minimum Description Length Principle (MDL)

- Another tool for model (order) selection
- *The model that compresses the data best is the best*
- Two-part MDL: To compress the data, we need to explain the model and the data given the model
  - $L(M) + L(D \mid M)$
  - Here: model is the tiling and we need to explain how to reconstruct the data given the tiling
    - The more homogeneous the tiles, the easier the latter part
- More on MDL next week…

# The Stijl Algorithm

- Goal: Find a tree of tiles (where tiles can overlap within their parent) that minimizes the description length of the data
- A greedy algorithm that adds tiles one-by-one
  - Can find a single, *optimal* tile to add in $O(nm \min(n,m))$
  - Uses MDL to decide the size of the tree
  - Based on a *linear*-time algorithm to decide the optimal tile *given* the columns of it

Tatti & Vreeken 2012

# From Geometric to Combinatorial

- We only know how to find geometric density tiles
  - What about combinatorial density tiles?
- Given a combinatorial tile, we can always re-order rows and columns to yield geometric tile
  - Not always possible for all tiles in a tiling simultaneously
- We can try to find an ordering *a priori*, and then find the geometric tiles in it

# Spectral Ordering

- Order the rows of $\boldsymbol{X}$ as follows:
  - Compute $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{X}^T$ (symmetric and positive semidefinite)
  - Let $\boldsymbol{D}$ be a diagonal matrix with the sums of $\boldsymbol{Y}$'s rows on its diagonal
  - Let $\boldsymbol{L}$ be the *Laplacian* of $\boldsymbol{Y}$: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{Y}$
  - Compute the second eigenvector of $\boldsymbol{L}$ (the Fiedler vector) $\boldsymbol{f}$
    - Intuitively, similar rows have similar values in $\boldsymbol{f}$
  - Order the rows based on their values in $\boldsymbol{f}$
- Columns are ordered analogously
- Here, similarity is measured using dot product
  - Other similarity measures are possible

Gionis, Mannila & Seppänen 2004

# References


- Geerts, F., Goethals, B. & Mielikäinen, T., 2004. *Tiling Databases*. In *Proceedings of the DS 2004*, pp. 77–122
- Gionis, A., Mannila, H., & Seppänen, J.K., 2004. Geometric and Combinatorial Tiles in 0–1 Data. In *Proceedings of the PKDD 2004*, pp. 173–184
- Tatti, N., & Vreeken, J., 2012. Discovering Descriptive Tile Trees by Mining Optimal Geometric Subtiles. In *Proceedings of the ECML PKDD 2012*, pp. 9–24