Geometric Pattern Mining using the MDL Principle

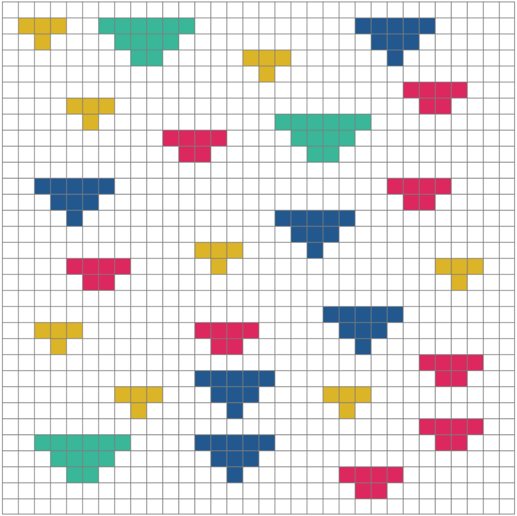
Micky Faas and Matthijs van Leeuwen

LIACS, Leiden University, Leiden, the Netherlands

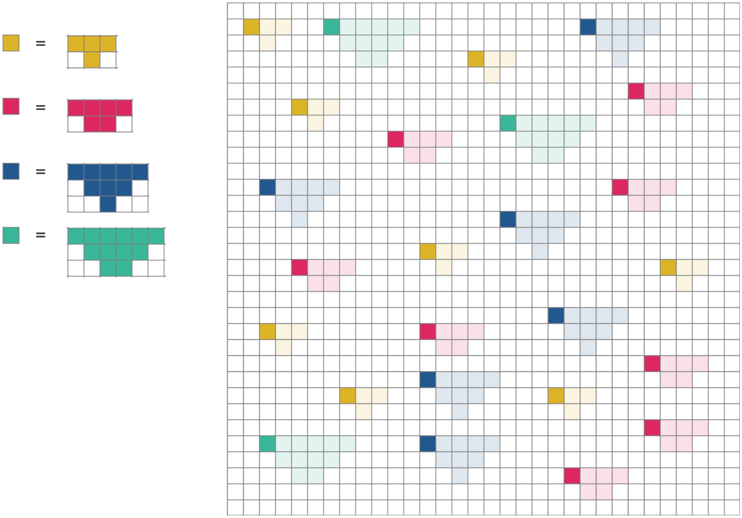
April 20, 2020

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Example Matrix



‘Compressed’ Matrix



Two-part MDL

Idea: the best solution is a balance of model and instantiation complexity, given by:

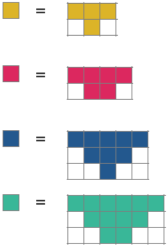
$$\underbrace{L(H)}_{\text{Model}} + \underbrace{L(A|H)}_{\text{Data given model}}$$

In this case:

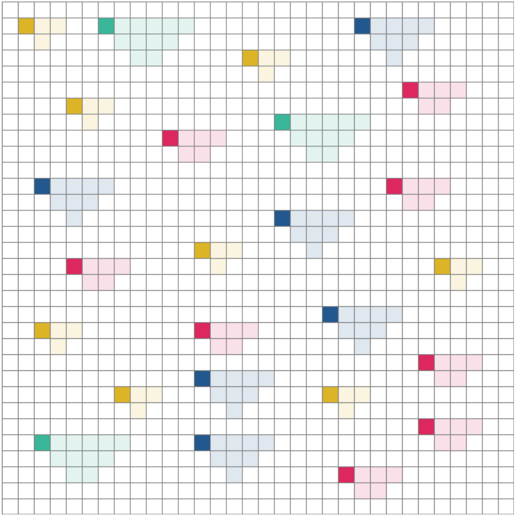
$$L\left(\underbrace{X = \overbrace{\begin{bmatrix} 1 & \cdot \\ \cdot & 1 \end{bmatrix}}^{\text{Pattern}}, Y = \overbrace{\begin{bmatrix} 1 \end{bmatrix}}^{\text{Pattern}}}_{\text{Model}}\right) + L\left(\underbrace{\overbrace{\begin{matrix} X & \cdot & \cdot & \cdot & \cdot & Y \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ X & \cdot & \cdot & \cdot & X & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ X & \cdot & X & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{matrix}}^{\text{Instantiation Matrix}}}_{\text{Data given model}}\right)$$

Model and Instantiation Matrix

Model H



Instantiation Matrix I

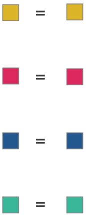


L(H) = 887
L(I) = 30
Compression: 89.5%

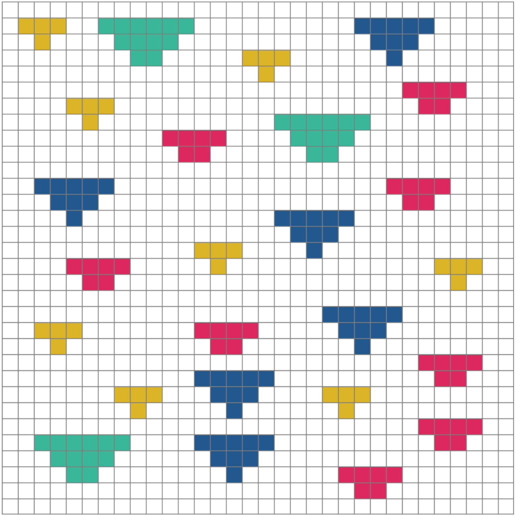
VOUW

Underfit Solution

Model H



Instantiation Matrix I



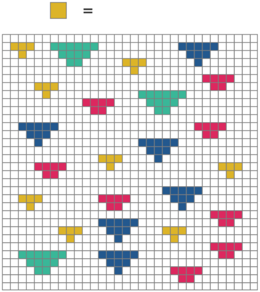
L(H) = 4

L(I) = 1024

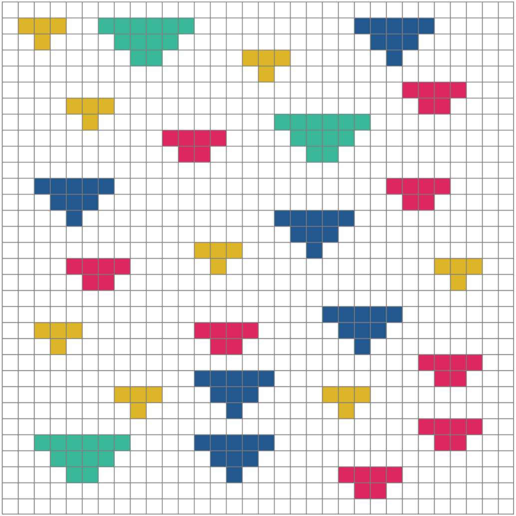
Ratio: 100.4%

Overfit Solution

Model H



Instantiation Matrix I

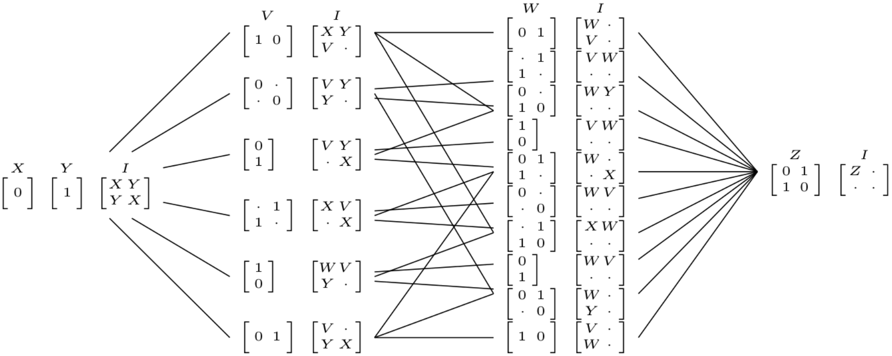


L(H) = 1024

L(I) = 1

Ratio: 100.1%

Search Space Lattice



Model space lattice for a 2 $\times$ 2 Boolean matrix. The V, W, and Z columns show which pattern is added in each step, while I depicts the current instantiation.

Candidates

Candidates are generated by enumerating all combinations of two adjacent patterns that occur in the instantiation matrix.

$$X = \begin{bmatrix}0\end{bmatrix}, Y = \begin{bmatrix}1\end{bmatrix},$$

$$I = \begin{bmatrix}X & Y \\ Y & X\end{bmatrix}$$

$$Z = \begin{bmatrix}1 & 0\end{bmatrix} \text{ gain } = ?$$

$$Z = \begin{bmatrix}0 & \cdot \\ \cdot & 0\end{bmatrix} \text{ gain } = ?$$

$$Z = \begin{bmatrix}0 \\ 1\end{bmatrix} \text{ gain } = ?$$

$$Z = \begin{bmatrix}\cdot & 1 \\ 1 & \cdot\end{bmatrix} \text{ gain } = ?$$

$$Z = \begin{bmatrix}1 \\ 0\end{bmatrix} \text{ gain } = ?$$

$$Z = \begin{bmatrix}0 & 1\end{bmatrix} \text{ gain } = ?$$

Length and Gain

$$\underbrace{\Delta L(A', c)}_{\text{Gain}} = \underbrace{\Big(L_1(H') + L_2(I')\Big)}_{\text{New Lengths}} - \underbrace{\Big(L_1(H) + L_2(I)\Big)}_{\text{Old Lengths}}$$

L_1 and L_2 are independent length functions that compute the length of the model and the instantiation, respectively.

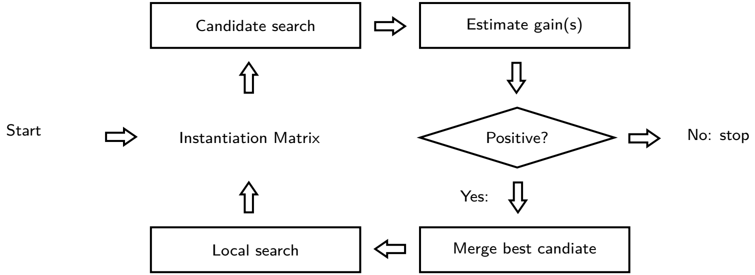
Please see the paper for more information on the encoding scheme.

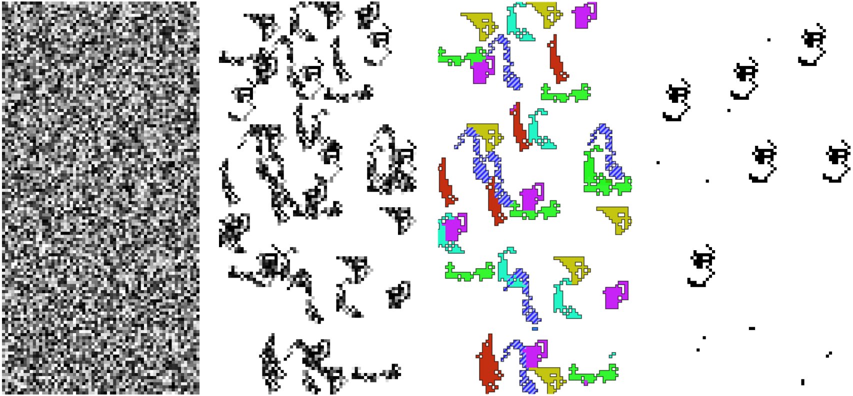
Pattern Union

We can construct complex patterns by repeatedly combining simpler ones. We use instances for this as they encode the position of one pattern relative to another.

$$\underbrace{\begin{matrix} X = \begin{bmatrix}0\end{bmatrix} \\[1em] Y = \begin{bmatrix}1\end{bmatrix} \end{matrix}}_{\text{Candidate patterns}} \longrightarrow \underbrace{\begin{matrix} \bar{X} = \begin{bmatrix} \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \\ \bar{Y} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot \end{bmatrix} \end{matrix}}_{\text{Instances}} \longrightarrow \bar{X} + \bar{Y} = \underbrace{\begin{bmatrix} \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot \\ \cdot & 1 & \cdot \end{bmatrix}}_{\text{Matrix sum}} \longrightarrow \underbrace{\begin{bmatrix} 0 & \cdot \\ \cdot & 1 \end{bmatrix}}_{\text{New pattern}}$$

Simplified Algorithm

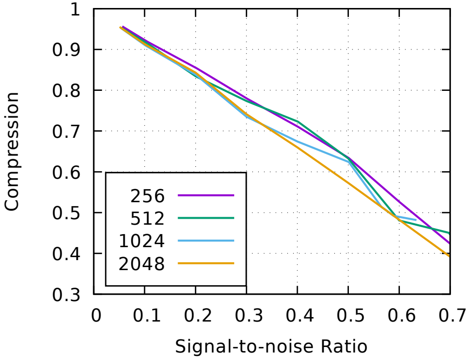




(a) Generated matrix (b) Ground truth (c) Found patterns (d) Difference

Figure: Synthetic patterns are added to a matrix filled with noise. The difference between the ground truth and the matrix reconstructed by the algorithm is used to compute precision and recall.

Compression and SNR



Results for different square matrices (256×256 to 1024×1024).
Signal-to-noise ratio is computed as $\frac{\text{signal}}{\text{signal+noise}}$.

The article was published at the Symposium on Intelligent Data Analysis (IDA) 2020

Archive link: http://arxiv.org/abs/1911.09587

Code repository: https://github.com/mickymuis/libvouw

Contact the authors:

Micky Faas <micky@edukitty.org>

Matthijs van Leeuwen <m.van.leeuwen@liacs.leidenuniv.nl>

Thank you for watching!