Data Mining and Matrices

05 – Semi-Discrete Decomposition

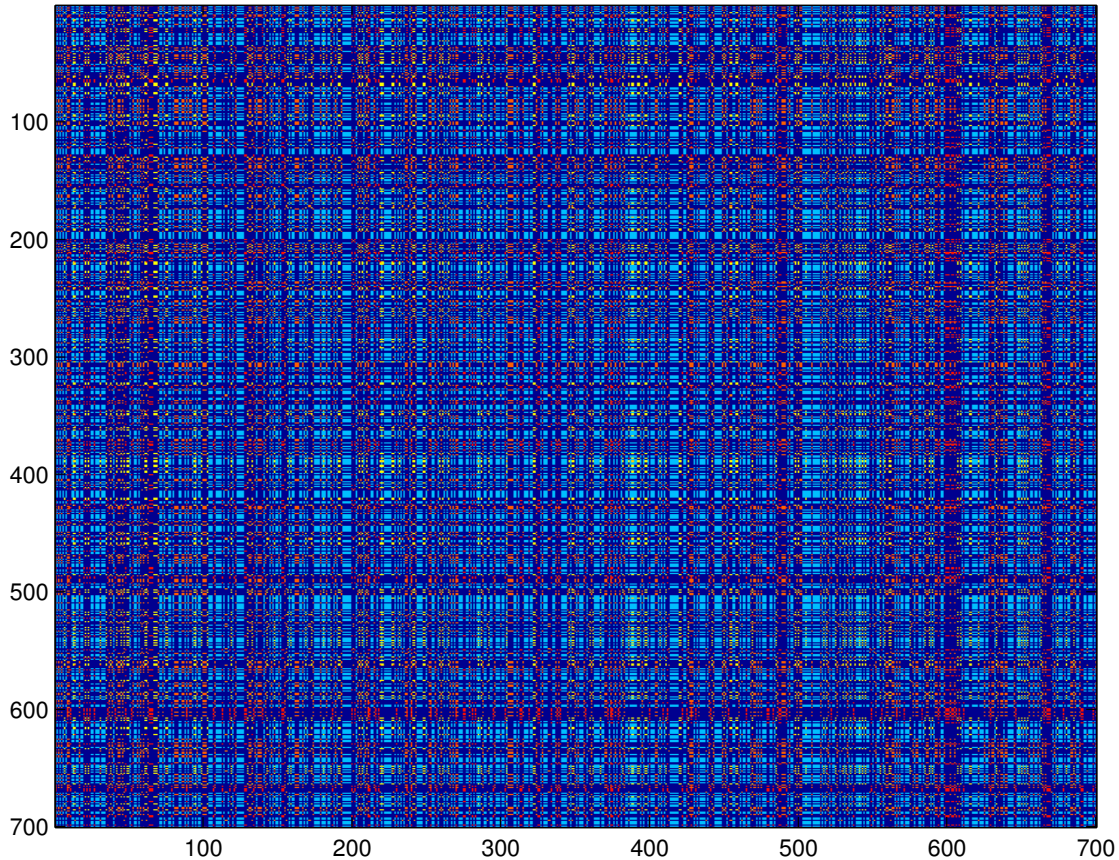
Rainer Gemulla, Pauli Miettinen

May 16, 2013

Outline

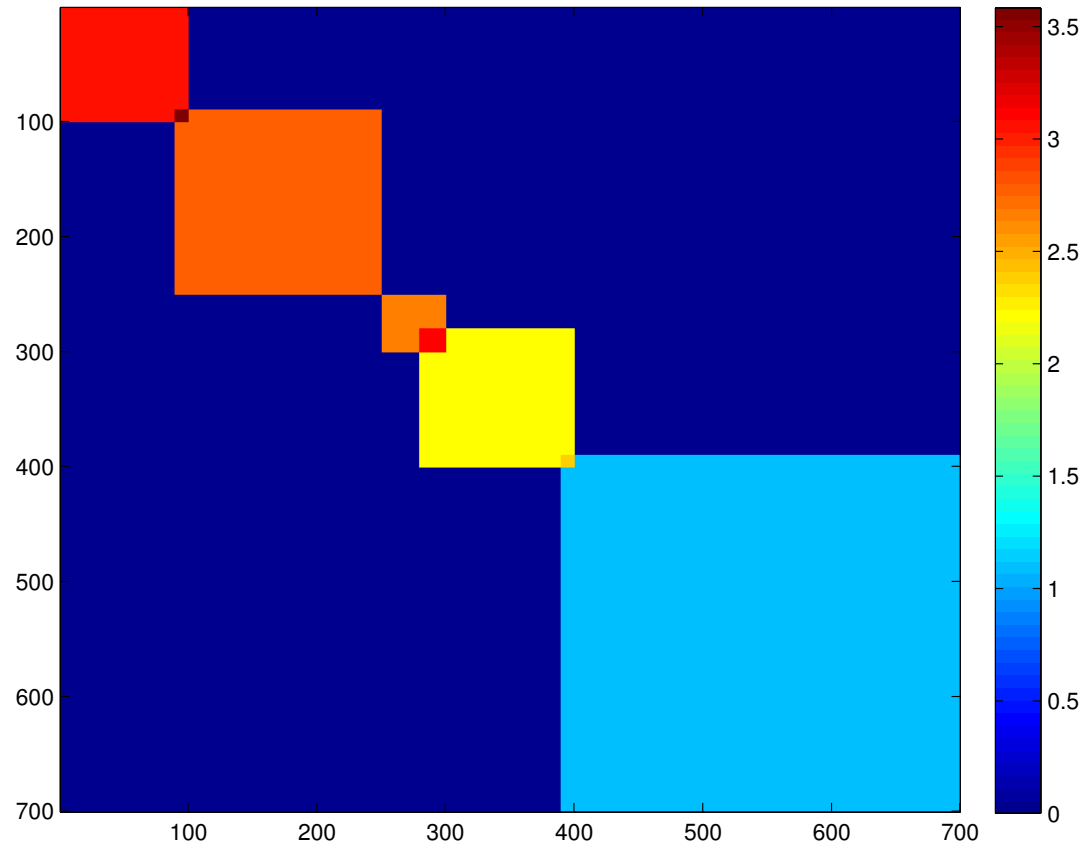
1. Hunting the Bump
2. Semi-Discrete Decomposition
3. The Algorithm
4. Applications
   - SDD alone
   - SVD + SDD
5. Wrap-Up

An example data



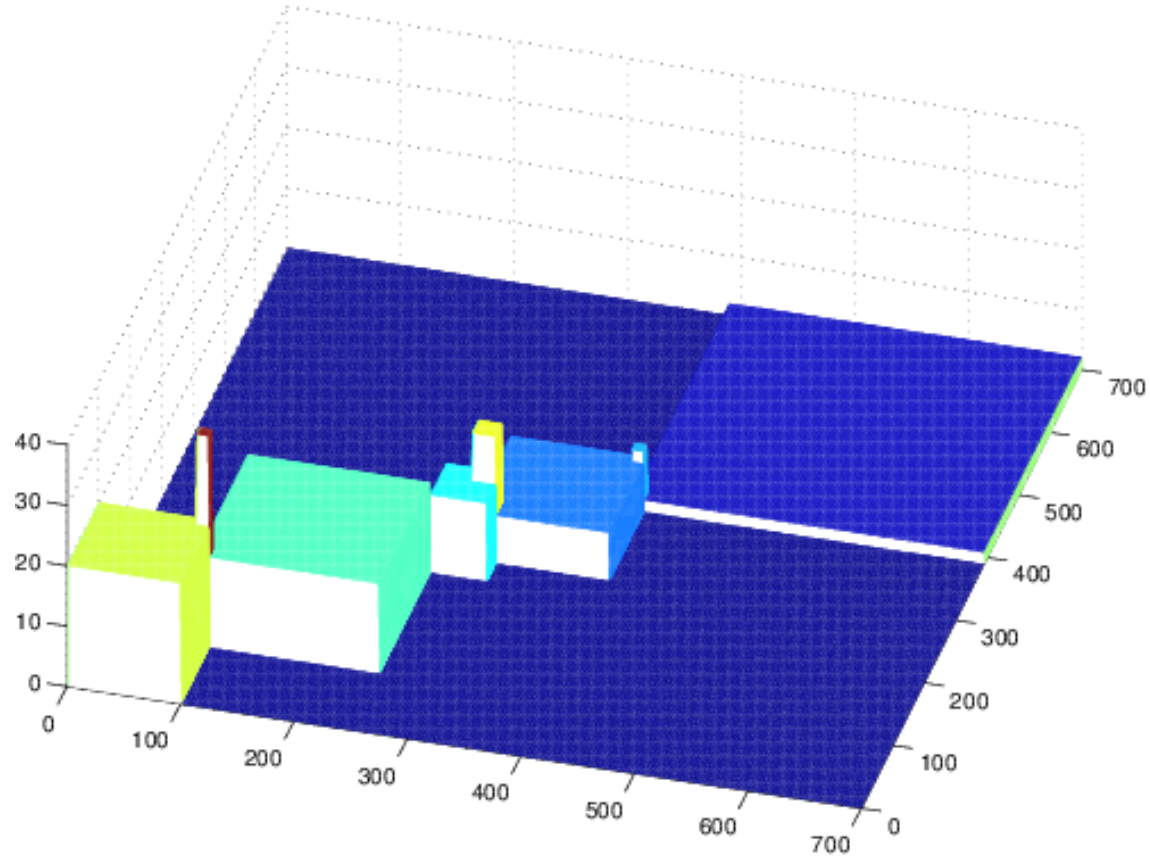
The data

An example data



The data after permuting rows and columns

An example data



The data in a 3D view

Can we find the bumps in the picture automatically (from unpermuted data)?

What is a bump?

$$\boldsymbol{A} = \begin{pmatrix} 3 & 1 & 3 \\ 2 & 3 & 1 \\ 3 & 2 & 3 \end{pmatrix}$$

$I = \{1, 3\}$
$J = \{1, 3\}$

$$\boldsymbol{x} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \boldsymbol{y} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$\boldsymbol{A} \circ \boldsymbol{x}\boldsymbol{y}^T = \begin{pmatrix} 3 & 0 & 3 \\ 0 & 0 & 0 \\ 3 & 0 & 3 \end{pmatrix}$$

- A **submatrix** of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains some rows of $\boldsymbol{A}$ and some columns of those rows
  - Let $I \subseteq \{1, 2, \ldots, m\}$ have the row indices and $J \subseteq \{1, 2, \ldots, n\}$ have the column indices of the submatrix
  - If $\boldsymbol{x} \in \{0, 1\}^m$ has $x_i = 1$ iff $i \in I$ and $\boldsymbol{y} \in \{0, 1\}^n$ has $y_j = 1$ iff $j \in J$, then $\boldsymbol{x}\boldsymbol{y}^T \in \{0, 1\}^{m \times n}$ has $(\boldsymbol{x}\boldsymbol{y}^T)_{ij} = 1$ iff a_{ij} is in the submatrix
  - $\boldsymbol{A} \circ \boldsymbol{x}\boldsymbol{y}^T$ has the values of the submatrix and zeros elsewhere
    - ⋆ $(\boldsymbol{A} \circ \boldsymbol{B})_{ij} = a_{ij} b_{ij}$ is the **Hadamard matrix product**
- The submatrix is uniform if all (or most) of its values are (approximately) the same
  - Exactly uniform submatrices with value δ can be written as $\delta \boldsymbol{x}\boldsymbol{y}^T$ — a bump

The next bump and negative values

- Assume we know how to find the largest bump of a matrix
- To find another bump, we can subtract the found bump from the matrix and find the largest bump of the residual matrix
  - But after subtraction we might have negative values in the matrix
- We can generalize the uniform submatrices to require uniformity only in magnitude
  - Allow characteristic vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ to take values from $\{-1, 0, 1\}$
  - If $\boldsymbol{x} = (-1, 0, -1)^T$ and $\boldsymbol{y} = (1, 0, -1)^T$, then

$$\delta \boldsymbol{x}\boldsymbol{y}^T = \begin{pmatrix} -\delta & 0 & \delta \\ 0 & 0 & 0 \\ \delta & 0 & -\delta \end{pmatrix}$$

- This allows us to define bumps in matrices with negative values

Outline

1. Hunting the Bump
2. Semi-Discrete Decomposition
3. The Algorithm
4. Applications
   - SDD alone
   - SVD + SDD
5. Wrap-Up

The definition

Semi-Discrete Decomposition

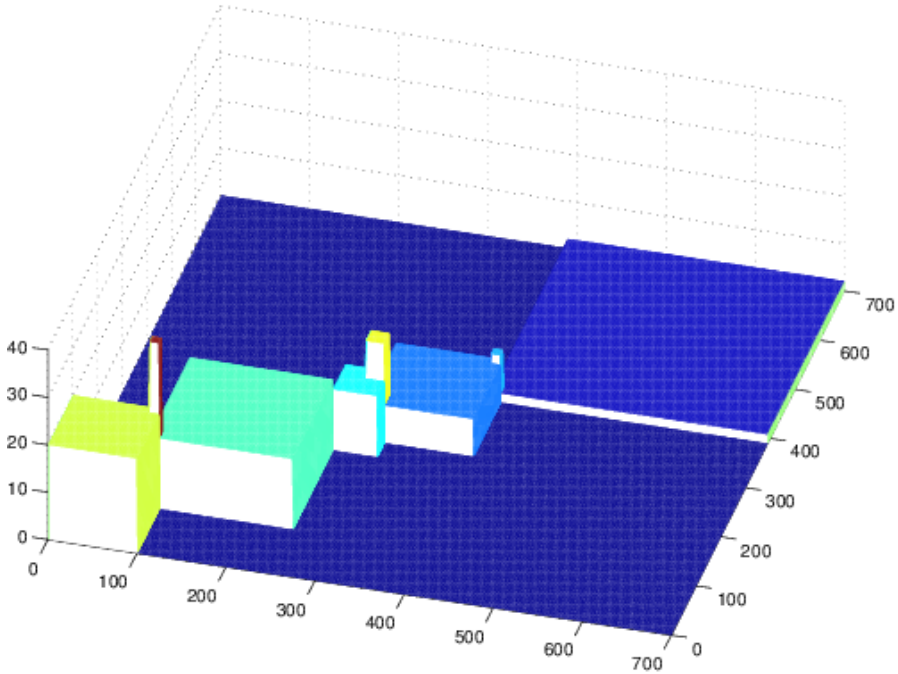
Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, the *semi-discrete decomposition* (SDD) of $\boldsymbol{A}$ of dimension k is

$$\boldsymbol{A} \approx \boldsymbol{X}_k \boldsymbol{D}_k \boldsymbol{Y}_k^T \ ,$$

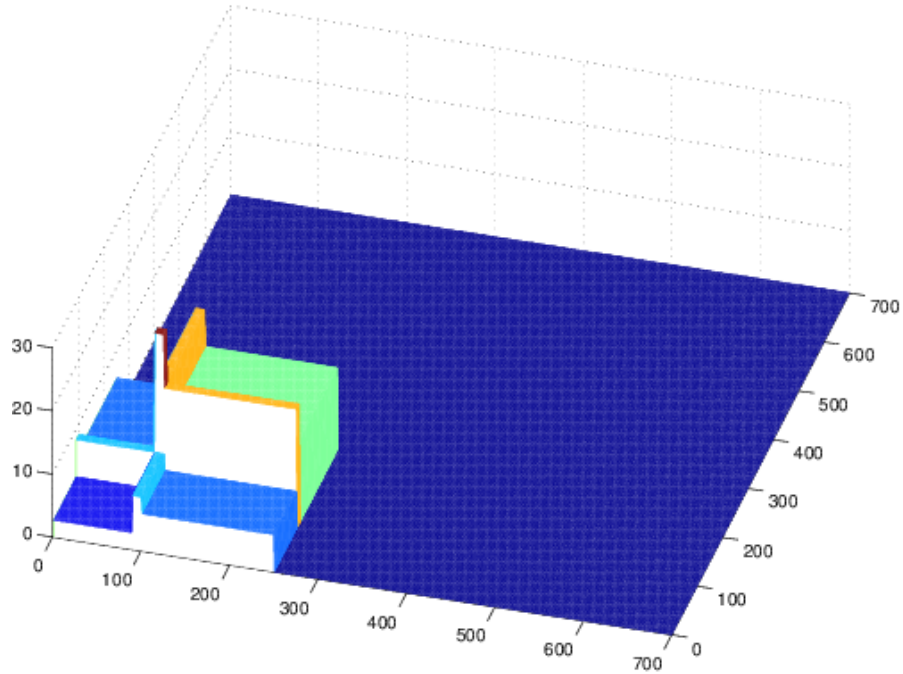
where

- $\boldsymbol{X}_k \in \{-1, 0, 1\}^{m \times k}$
- $\boldsymbol{Y}_k \in \{-1, 0, 1\}^{n \times k}$
- $\boldsymbol{D}_k \in \mathbb{R}_+^{k \times k}$ is a diagonal matrix

Example

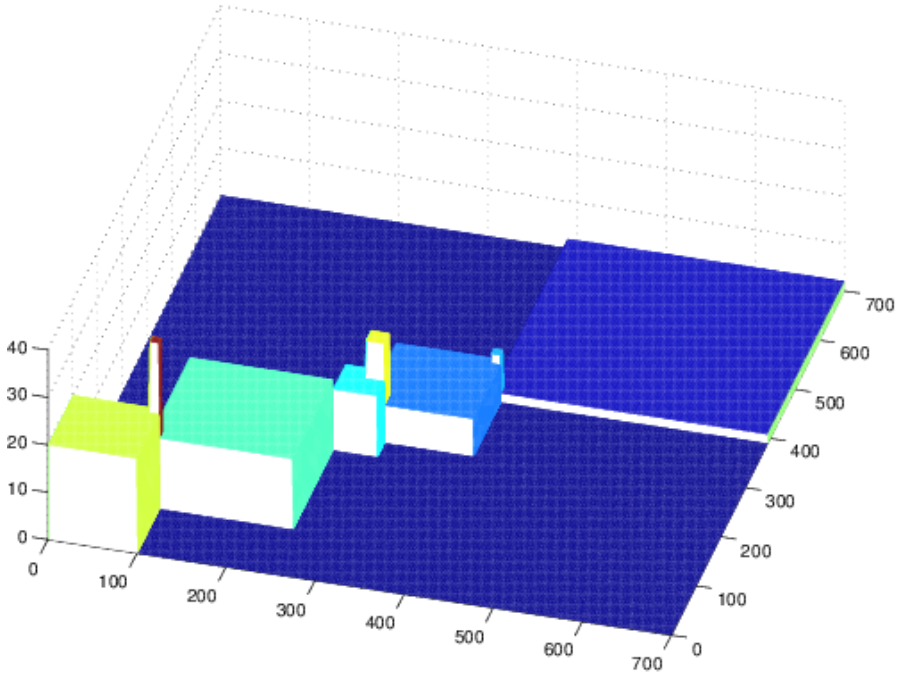


The data

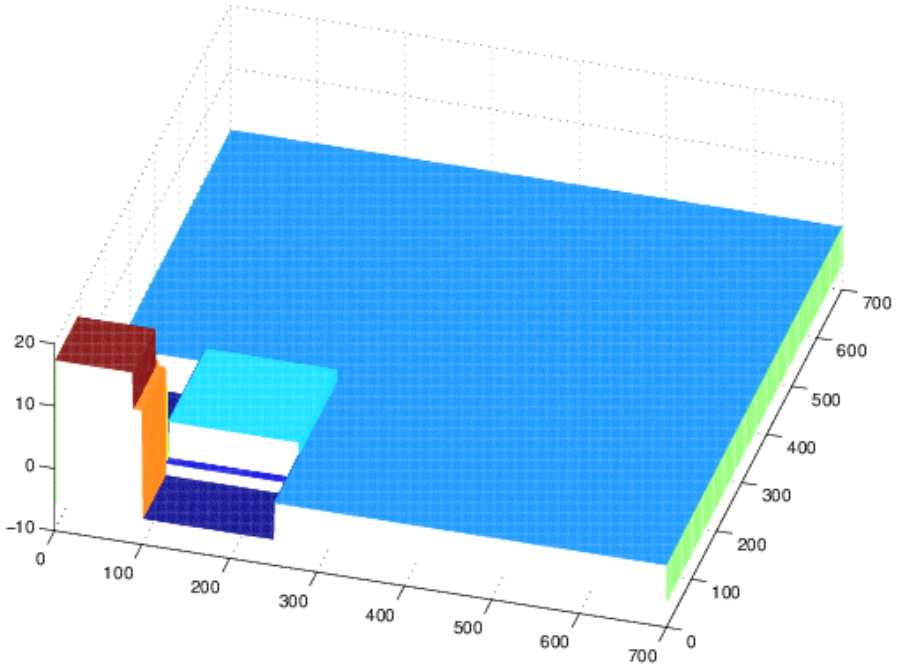


The first component $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$ using SVD

Example



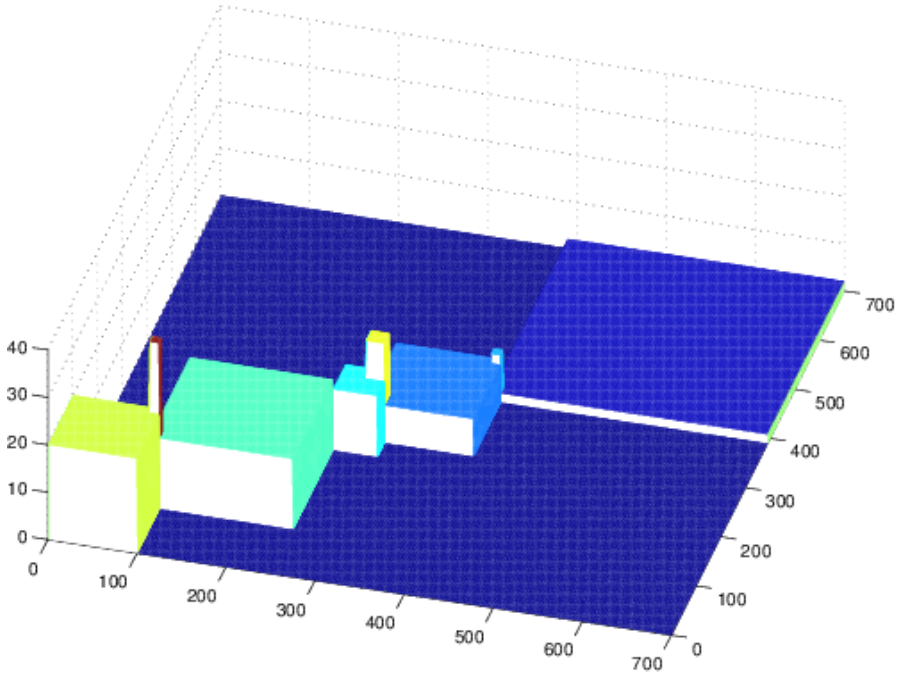
The data



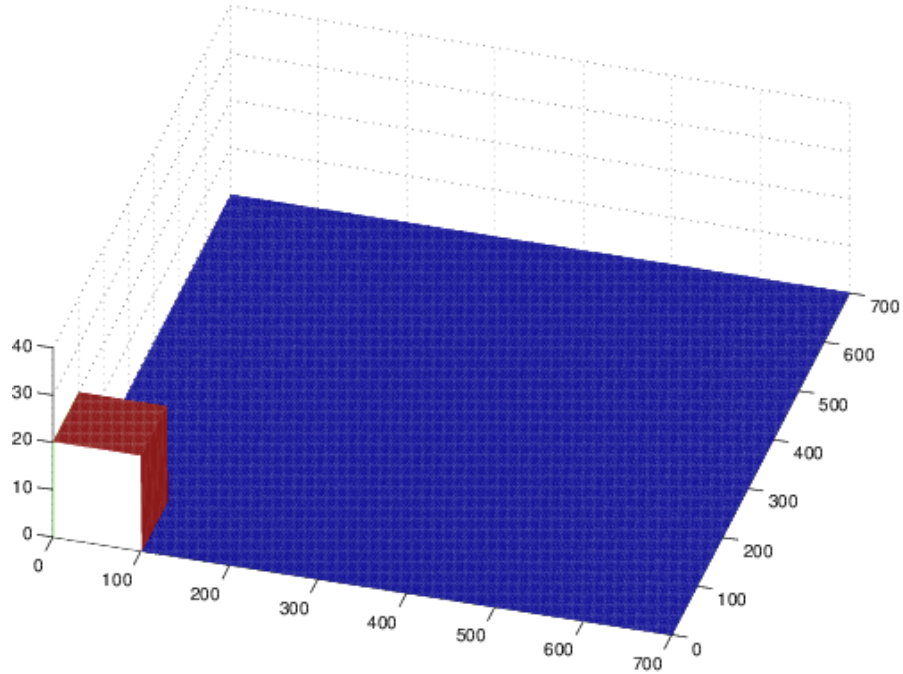
The second component $\sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T$ using SVD

The SVD cannot find the bumps

Example

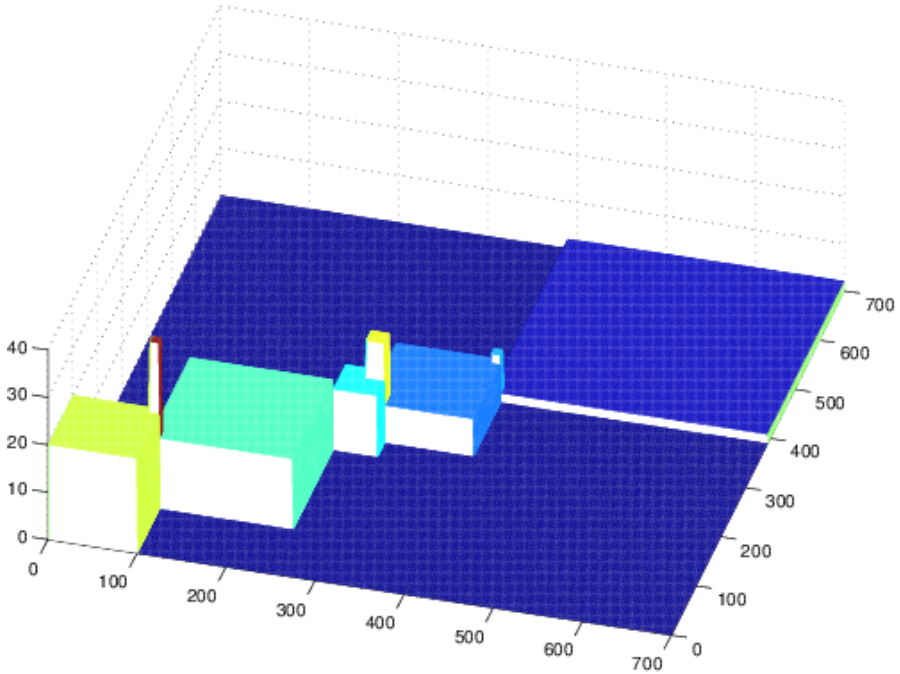


The data

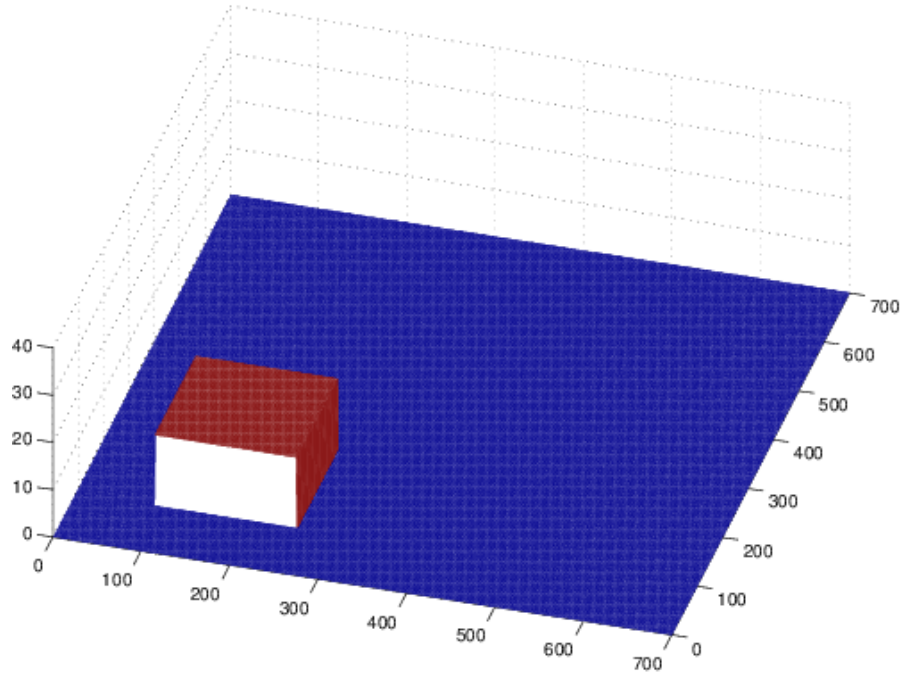


The first bump $d_1 \boldsymbol{x}_1 \boldsymbol{y}_1^T$ using SDD

Example

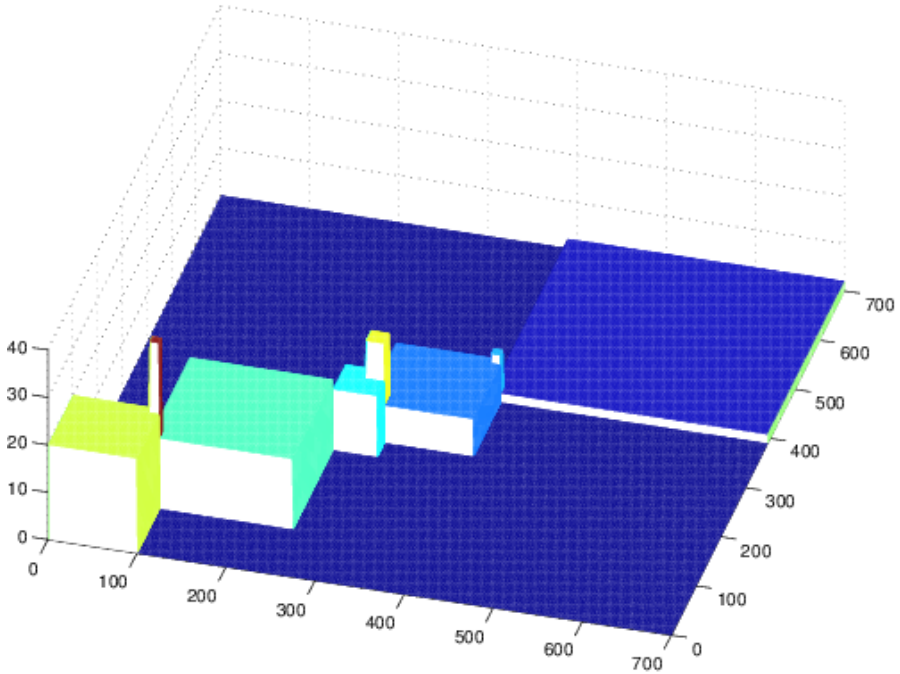


The data

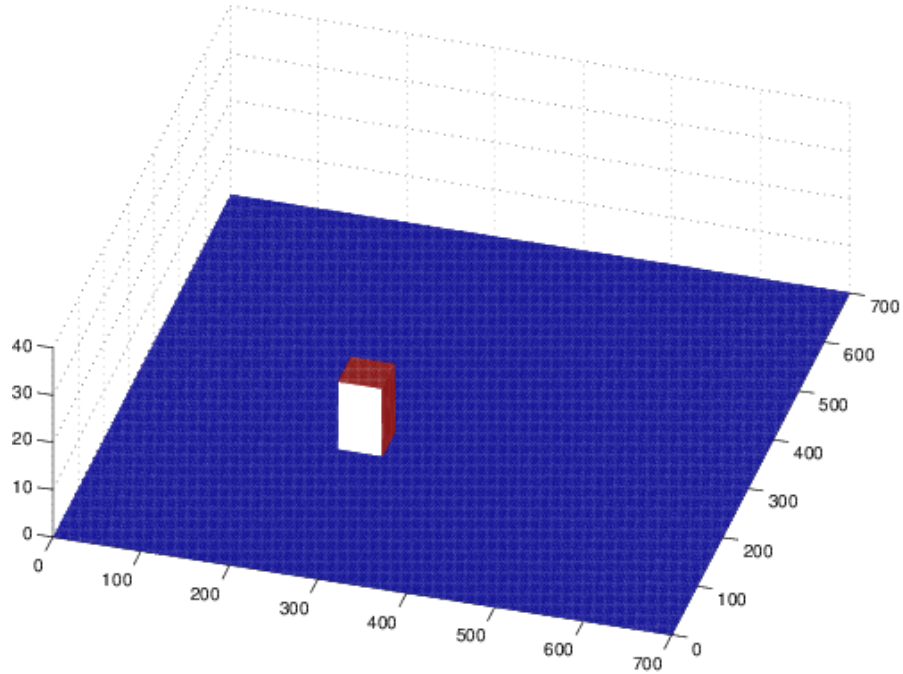


The second bump $d_2 \boldsymbol{x}_2 \boldsymbol{y}_2^T$ using SDD

Example

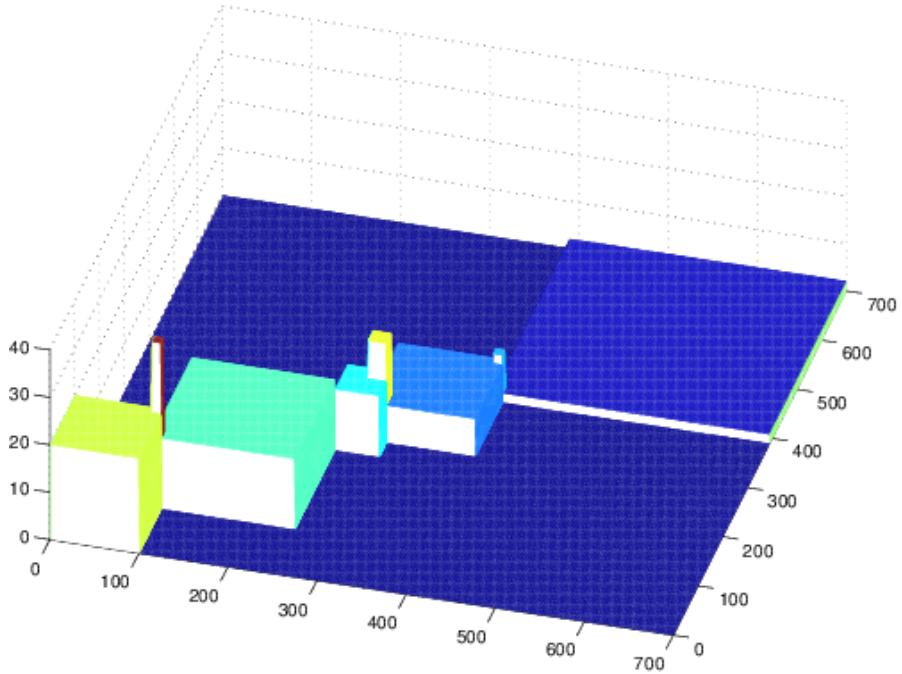


The data

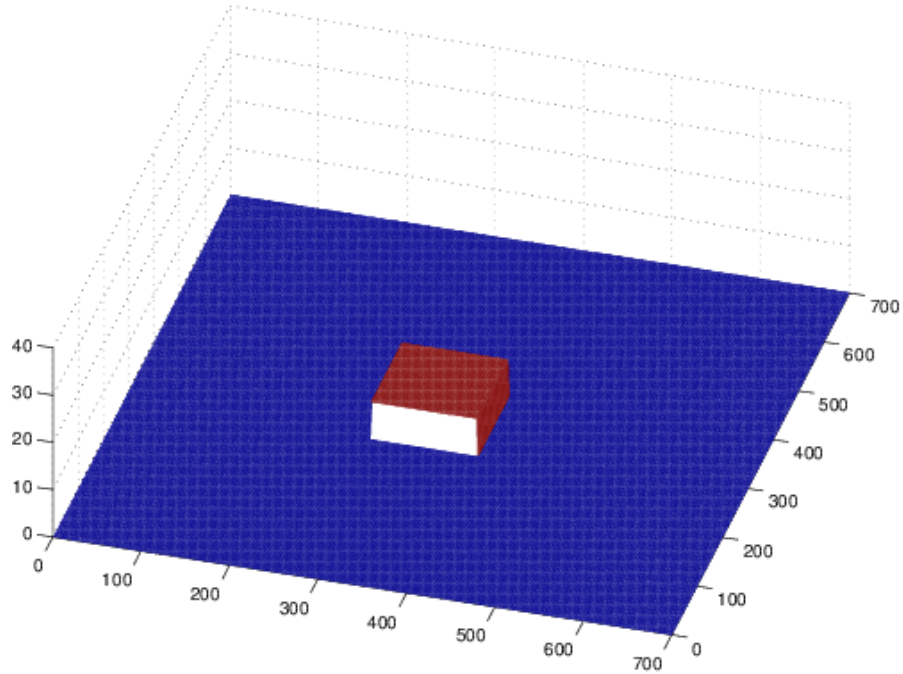


The third bump $d_3 \boldsymbol{x}_3 \boldsymbol{y}_3^T$ using SDD

Example

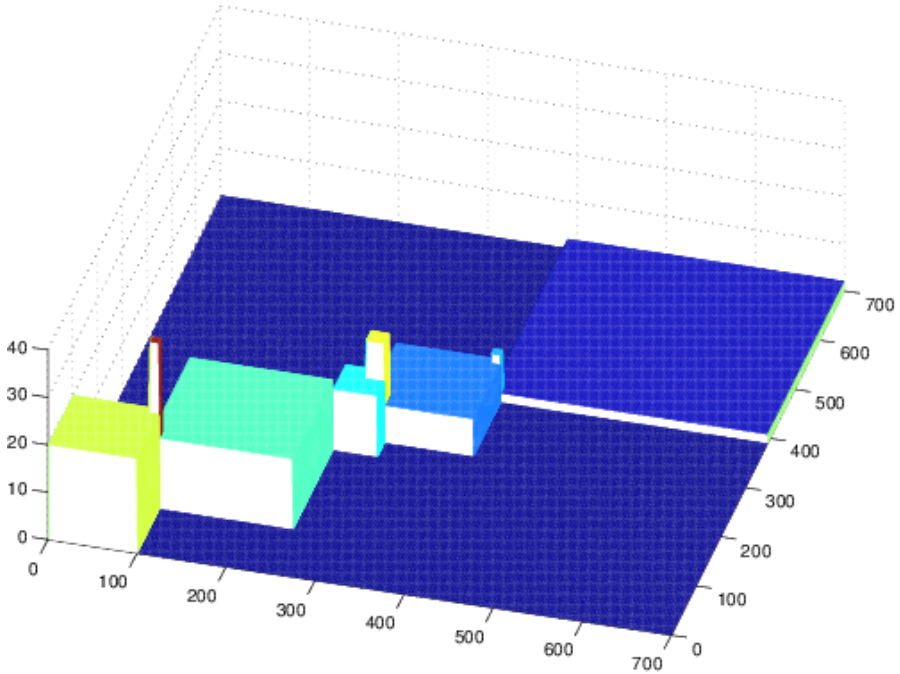


The data

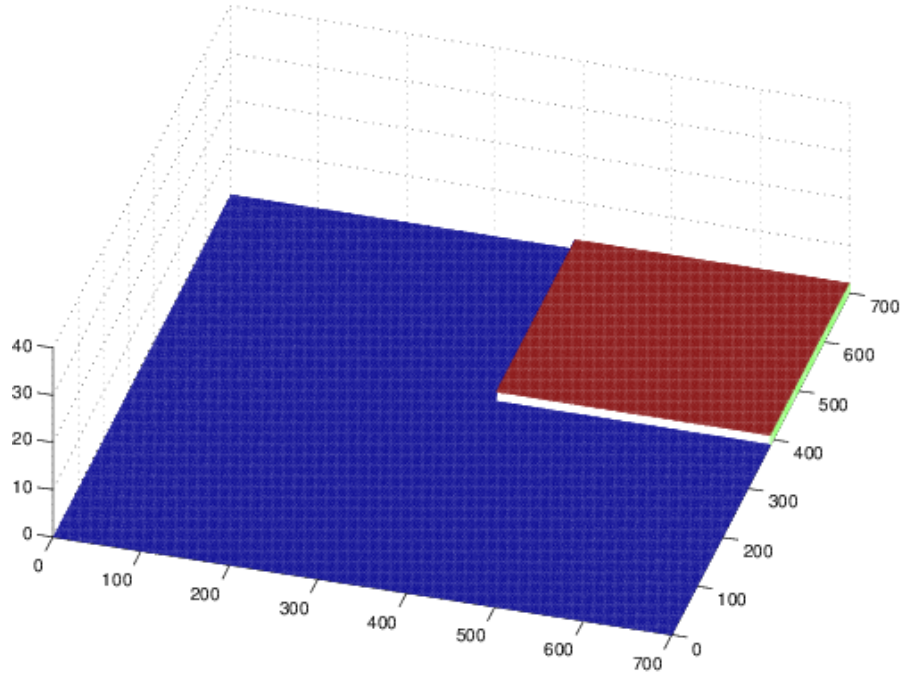


The fourth bump $d_4 \boldsymbol{x}_4 \boldsymbol{y}_4^T$ using SDD

Example

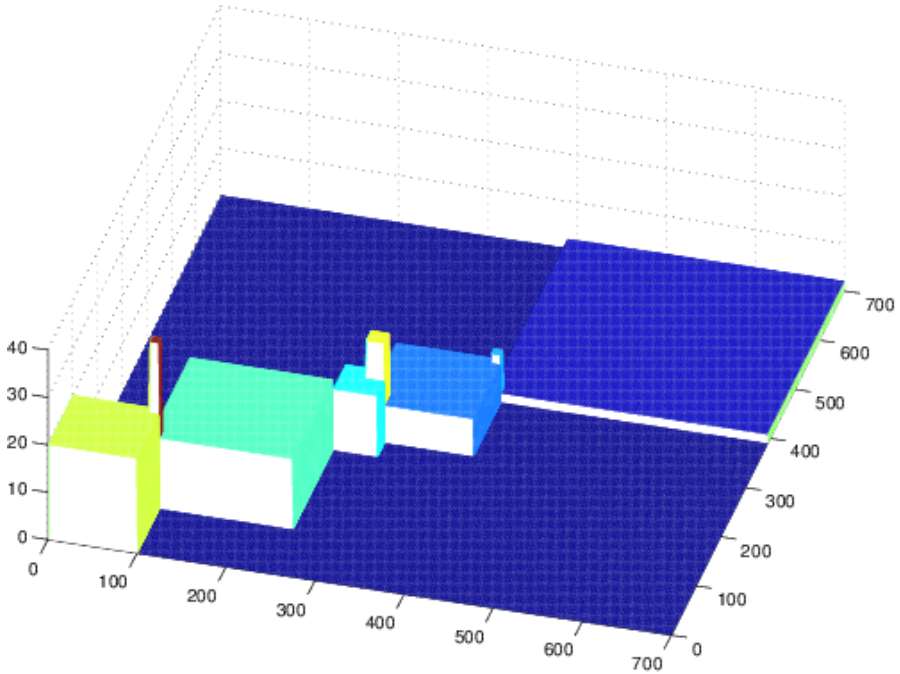


The data

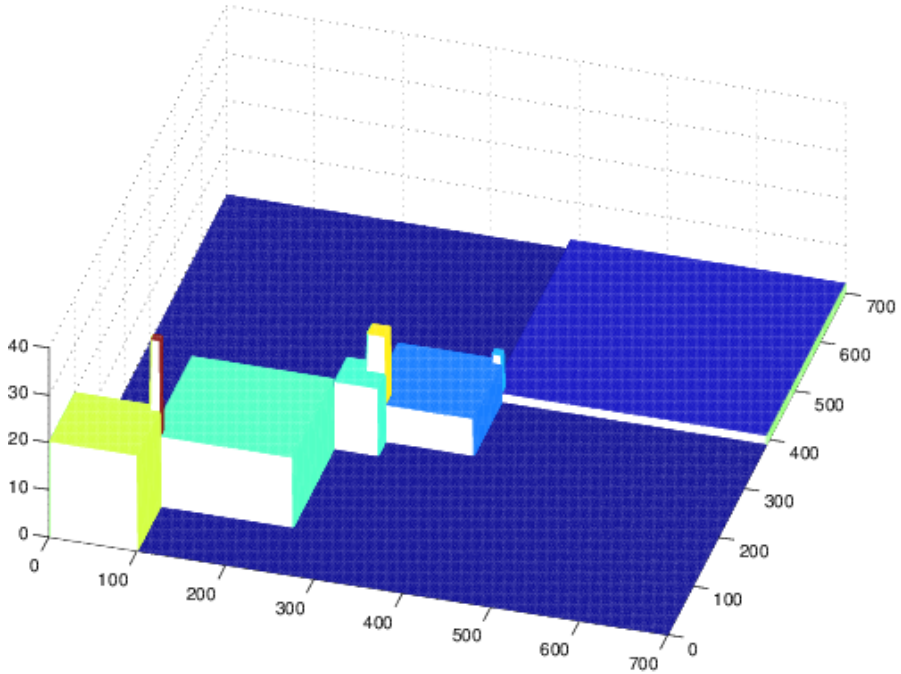


The fifth bump $d_5 \boldsymbol{x}_5 \boldsymbol{y}_5^T$ using SDD

Example



The data



The 5-dimensional SDD approximation $\boldsymbol{X}_5 \boldsymbol{D}_5 \boldsymbol{Y}_5^T$

Properties of SDD

- The columns of $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ do not need to be linearly independent
  - The same column can be even repeated multiple times
- The dimension k might need to be large for accurate approximation (compared to SVD)
  - $k = \min\{n, m\}$ is not necessarily enough for exact SDD
    - $k = nm$ is always enough
  - First factors don't necessarily explain much about the matrix
- SDD factors are local
  - Only affect a certain submatrix, typically not every element
  - SVD factors typically change every value
- Storing an k-dimensional SDD takes less space than storing rank-k truncated SVD
  - $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ are ternary and often sparse
- For every rank-1 layer of an SDD, all non-zero values in the layer have the same magnitude (d_{ii} for layer i)

Interpretation

- The factor interpretation is not very useful as the factors are not independent
  - A later factor can change just a subset of values already changed by an earlier factor
- The SDD can be interpret as a form of bi-clustering
  - Every layer (bump) defines a group of rows and columns with homogeneous values in the residual matrix
- The component interpretation is natural to SDD
  - The SDD is a sum of local bumps
  - SDD doesn't model global phenomena (e.g. noise) well

Outline

1. Hunting the Bump
2. Semi-Discrete Decomposition
3. The Algorithm
4. Applications
   - SDD alone
   - SVD + SDD
5. Wrap-Up

The outline of the algorithm

1. **Input:** Matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, non-negative integer k
2. **Output:** k-dimensional SDD of $\boldsymbol{A}$, i.e. matrices $X_k \in \{-1, 0, 1\}^{m \times k}$, $\boldsymbol{Y}_k \in \{-1, 0, 1\}^{n \times k}$, and diagonal $\boldsymbol{D}_k \in \mathbb{R}_+^{k \times k}$
3. $\boldsymbol{R}_1 \leftarrow \boldsymbol{A}$
4. **for** $i = 1, \ldots, k$
    1. Select $\boldsymbol{y}_i \in \{-1, 0, 1\}^n$
    2. **while** not converged
        1. Compute $\boldsymbol{x}_i \in \{-1, 0, 1\}^m$ given $\boldsymbol{y}_i$ and $\boldsymbol{R}_i$
        2. Compute $\boldsymbol{y}_i$ given $\boldsymbol{x}$ and $\boldsymbol{R}_i$
    3. **end while**
    4. Set d_i to the average of $\boldsymbol{R}_i \circ \boldsymbol{x}_i \boldsymbol{y}_i^T$ over the non-zero locations of $\boldsymbol{x}\boldsymbol{y}^T$
    5. Set $\boldsymbol{x}_i$ as the ith column of $\boldsymbol{X}_i$, $\boldsymbol{y}_i$ the ith column of $\boldsymbol{Y}_i$, and d_i the ith value of $\boldsymbol{D}_i$
    6. $\boldsymbol{R}_{i+1} \leftarrow \boldsymbol{R}_i - d_i \boldsymbol{x}_i \boldsymbol{y}_i^T$
5. **end for**
6. **return** $\boldsymbol{X}_k$, $\boldsymbol{Y}_k$, and $\boldsymbol{D}_k$

Finding the bump

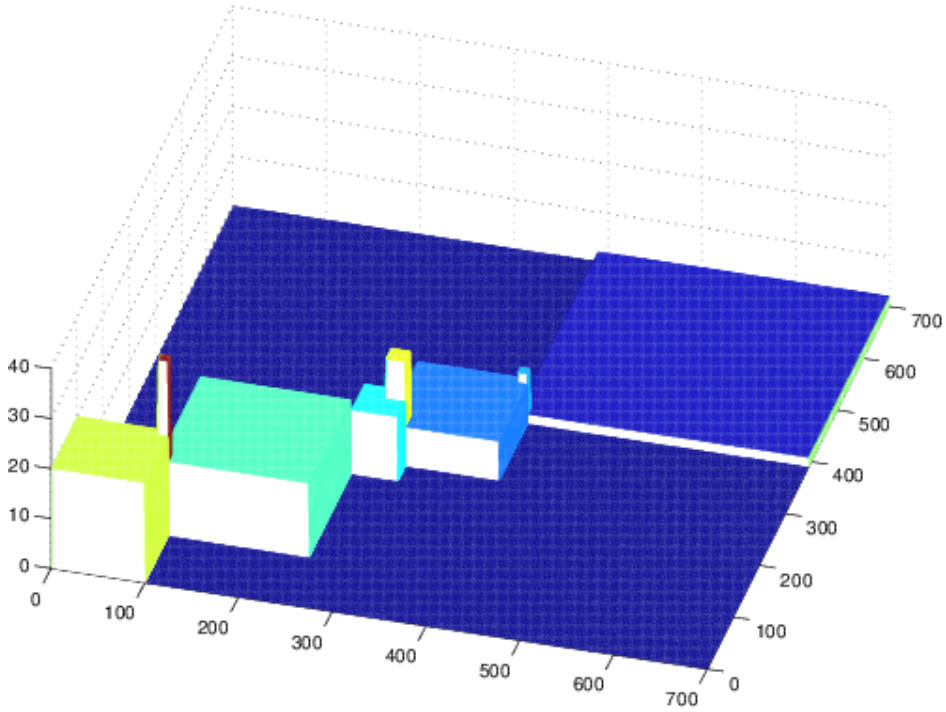
- **Problem:** Given $\boldsymbol{R} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{y} \in \{-1, 0, 1\}^n$, find $\boldsymbol{x} \in \{-1, 0, 1\}^m$ such that $\|\boldsymbol{R} - d\boldsymbol{x}\boldsymbol{y}^T\|_F^2$ is minimized
  - We set $d \leftarrow \boldsymbol{x}^T\boldsymbol{R}\boldsymbol{y}/\|\boldsymbol{x}\|_2^2\|\boldsymbol{y}\|_2^2$ (the average of $\boldsymbol{R} \circ \boldsymbol{x}\boldsymbol{y}^T$ over the non-zero locations of $\boldsymbol{x}\boldsymbol{y}^T$)
  - We want to minimize the residual norm
- Set $\boldsymbol{s} \leftarrow \boldsymbol{R}\boldsymbol{y}$
- **Task:** Find $\boldsymbol{x}$ that maximizes $F(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T\boldsymbol{s})^2/\|\boldsymbol{x}\|_2^2$
  - Maximizing F equals minimizing the residual norm after d is set as above
  - Can be solved optimally by trying 2^m different binary vectors and setting the sign appropriately
- **Solution:** Order values s_i so that $|s_{i_1}| \geq |s_{i_2}| \geq \cdots \geq |s_{i_m}|$ and set $x_{i_j} \leftarrow \text{sign}(s_{i_j})$ for the *first* J values s_i and 0 elsewhere
  - J is the number of nonzeros in $\boldsymbol{x}$
    - ⋆ Because we don't know J, we have to try every possibility and select the best
  - Values s_i contain the row sums of $\boldsymbol{R}$ from those columns that are selected by $\boldsymbol{y}$ and with sign set accordingly

Selecting the initial vector $\boldsymbol{y}$

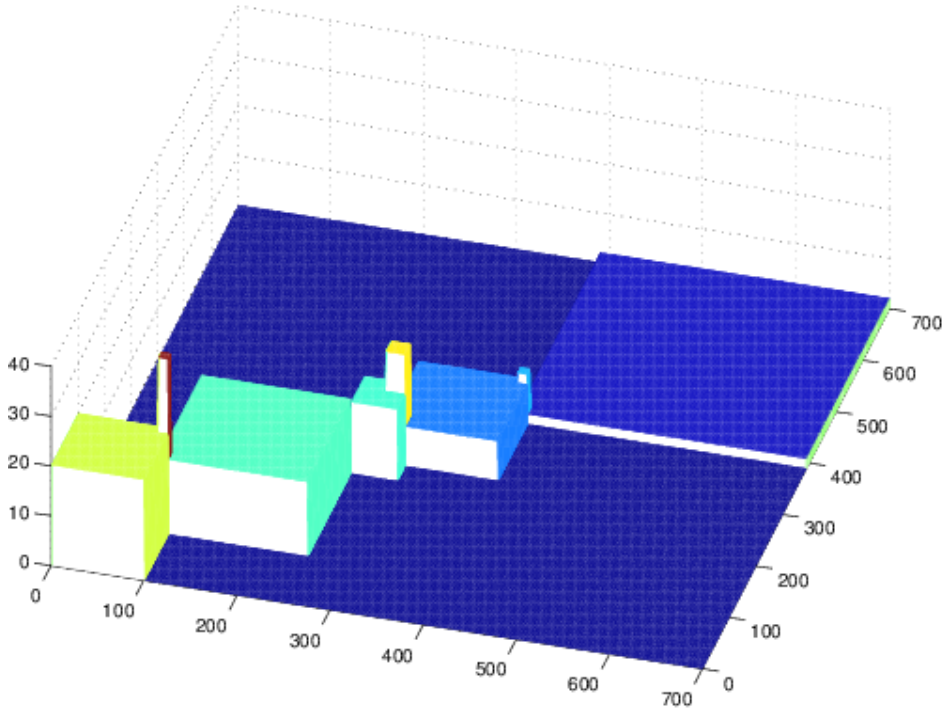
There are many ways to select the initial vector:

- MAX: set $y_j = 1$ for the column j that has the largest squared value of $\boldsymbol{R}$ and rest to zero
  - Intuition: the very largest squared value is probably in the best bump
- CYC: set $y_j = 1$ for $j = (k \mod n) + 1$
  - Cycle thru the columns
- THR: select a unit vector $\boldsymbol{y}$ that satisfies $\|\boldsymbol{Ry}\|_F^2 \geq \|\boldsymbol{R}\|_F^2/n$
  - The selected column must have a squared sum that's above the average squared sum
  - The selection can be random or columns can be tried one-by-one
    - ★ The CYC and THR can be mixed

Example result

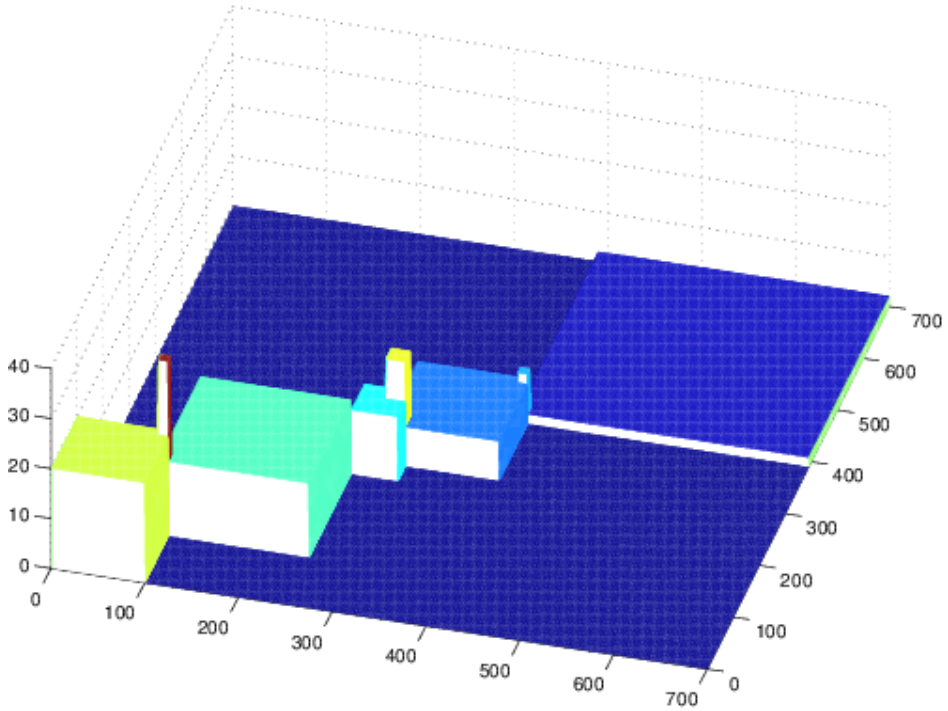


The data

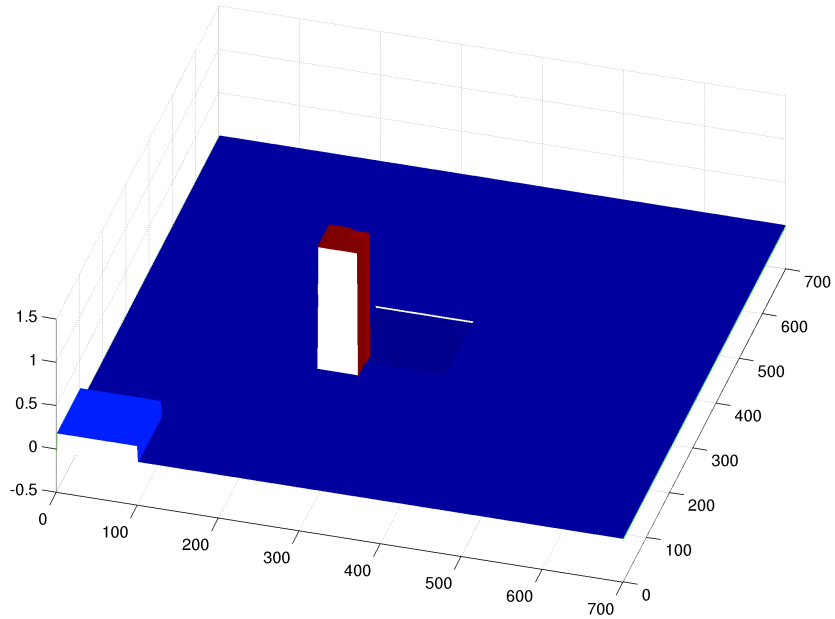


5-dimensional SDD

Example result



The data



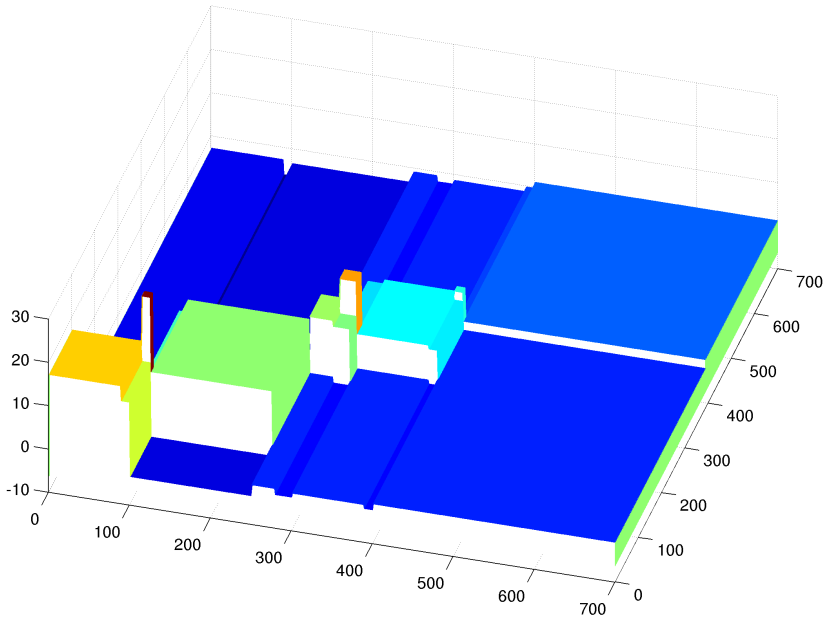
The matrix $\boldsymbol{X}_5 \boldsymbol{D}_5 \boldsymbol{Y}_5^T - \boldsymbol{A}$

Normalization

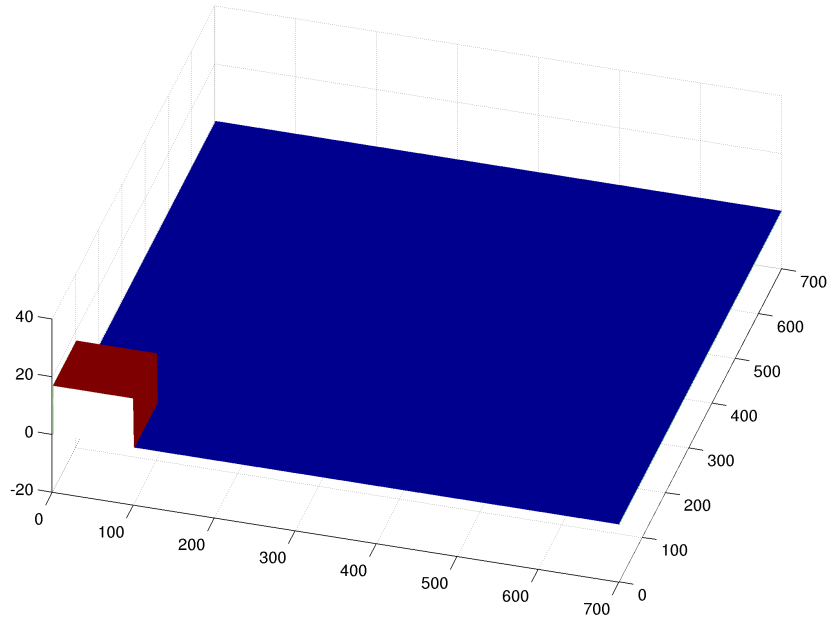
Normalization can have a profound effect on SDD

- Zero centering the columns will change the type of bumps found
  - The bumps in the original data have the largest-magnitude values
  - The bumps in the zero-centered data have the most extreme values
- Normalizing the variance will make the matrix to have more uniform values and thus changes the bumps
- Squaring the values will promote smaller bumps of exceptionally high values
- Square-rooting the values will promote larger bumps of smaller magnitude

Normalization example: zero-centered data

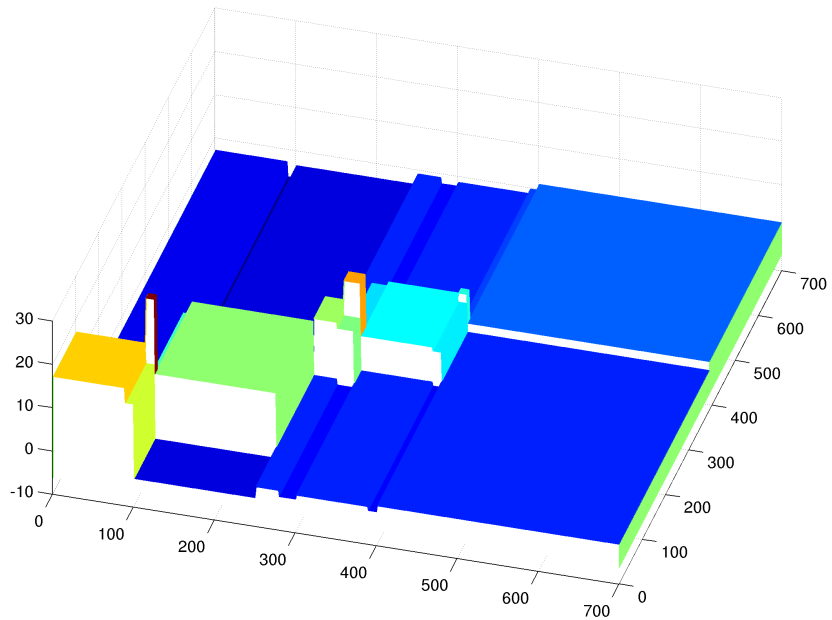


Zero-centered data

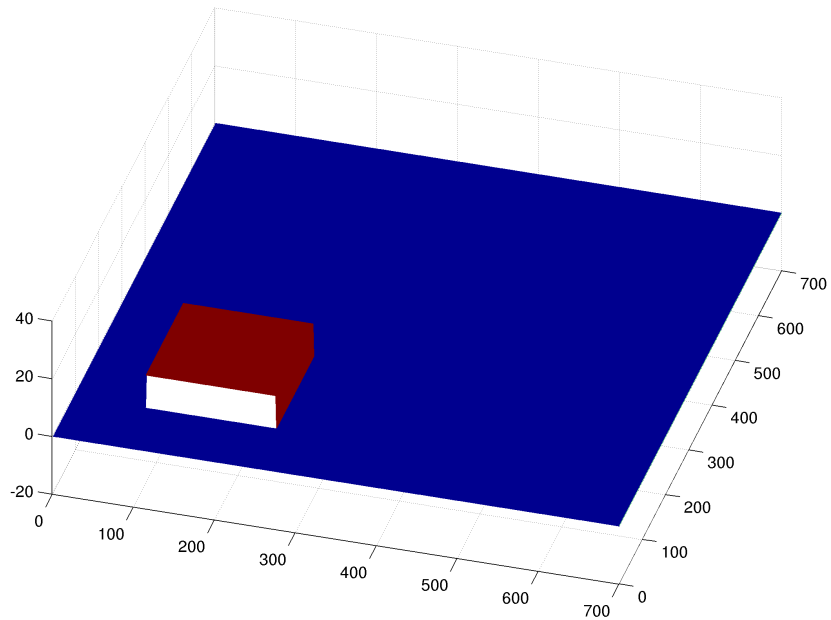


The first bump

Normalization example: zero-centered data

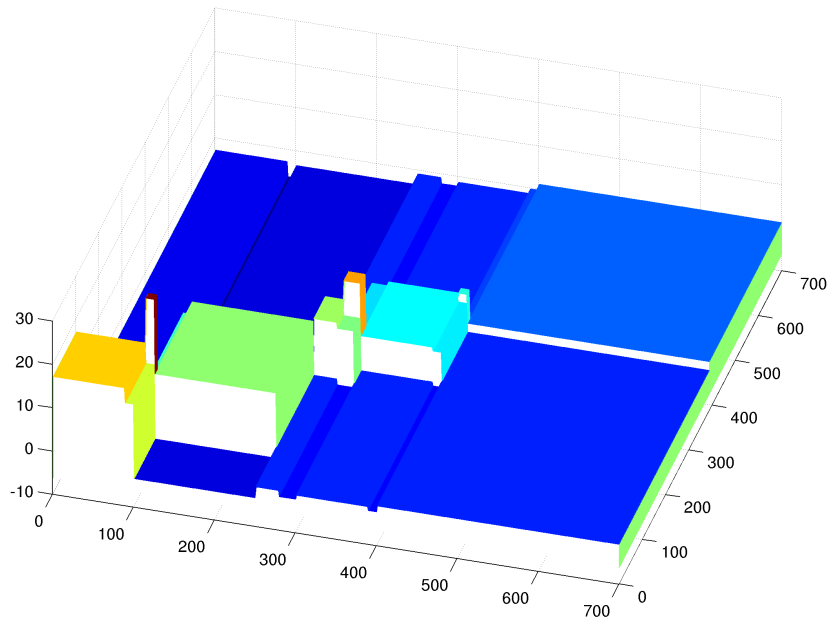


Zero-centered data

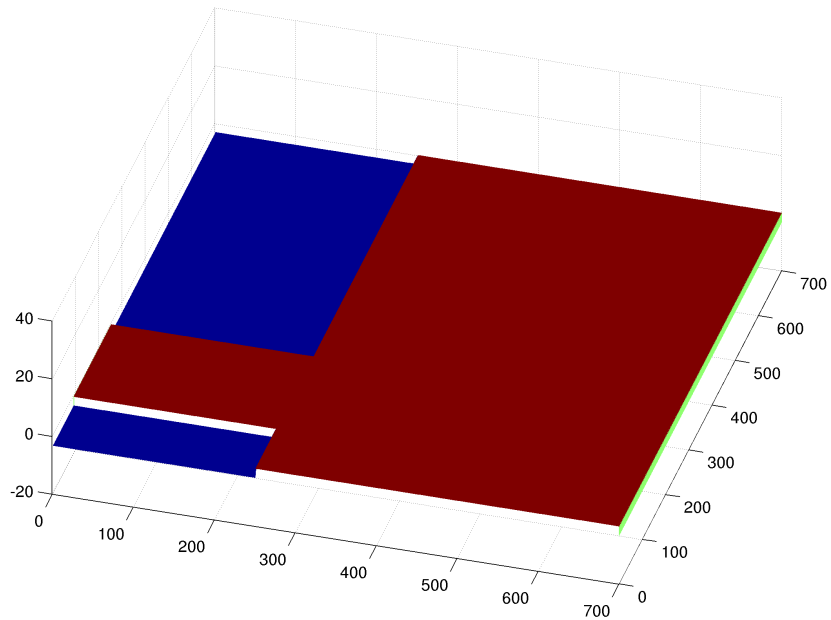


The second bump

Normalization example: zero-centered data



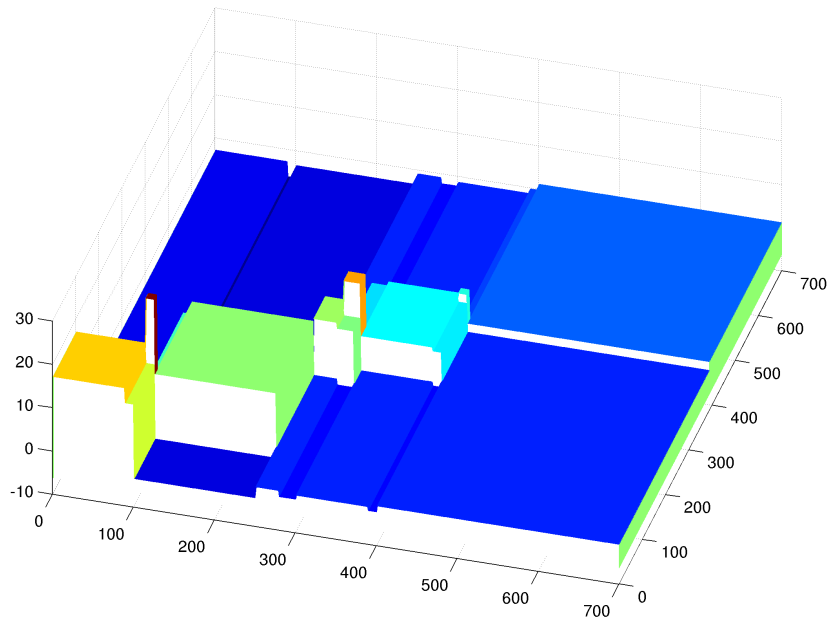
Zero-centered data



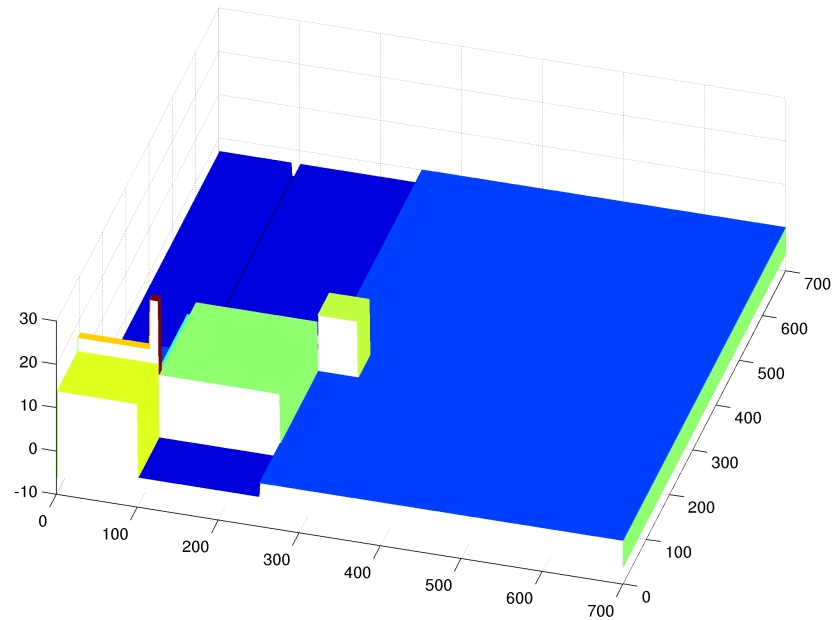
The third bump

Note that here red means 0

Normalization example: zero-centered data

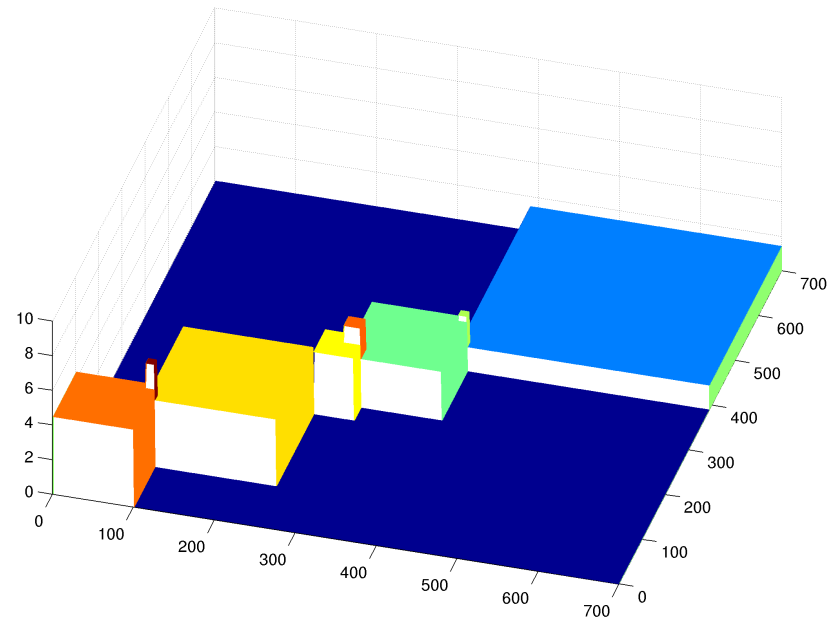


Zero-centered data

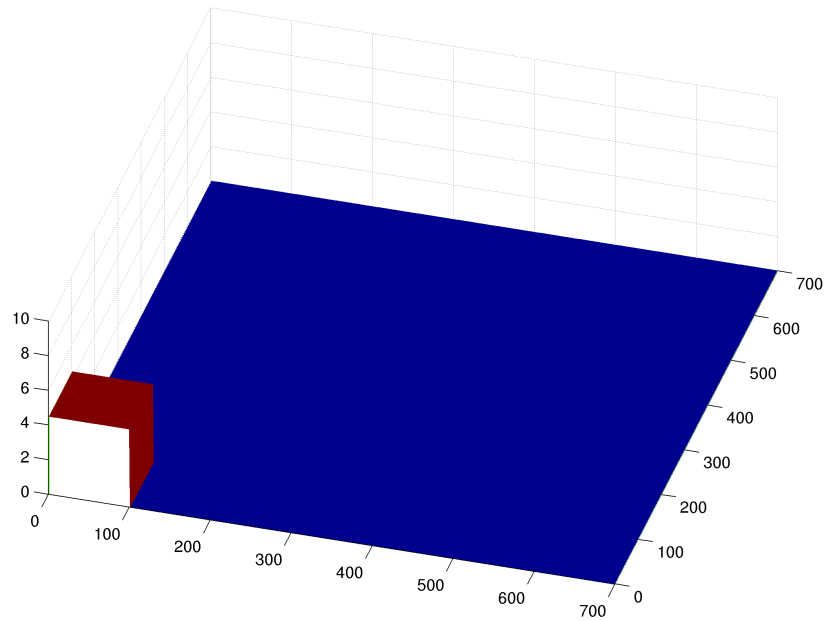


5-dimensional SDD

Normalization example: square-root of data

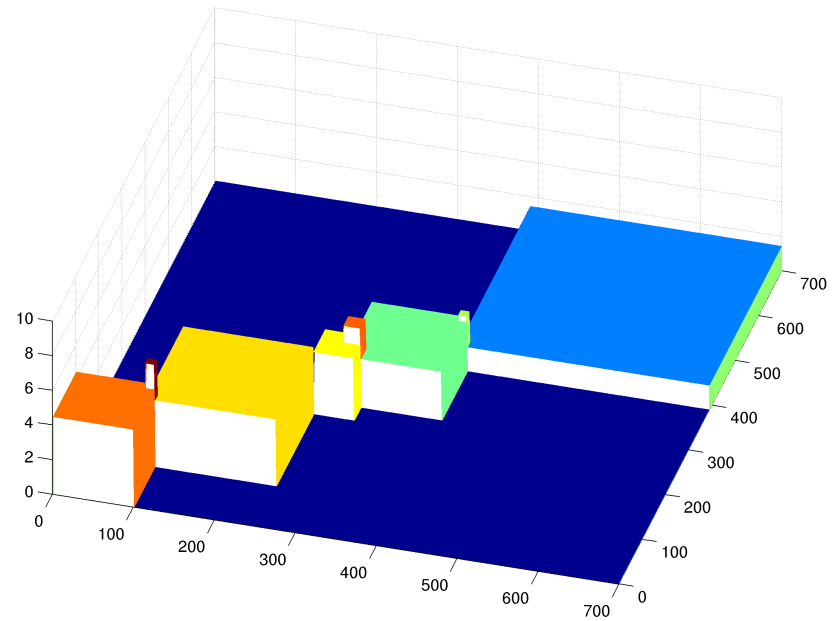


Data after taking element-wise square-root

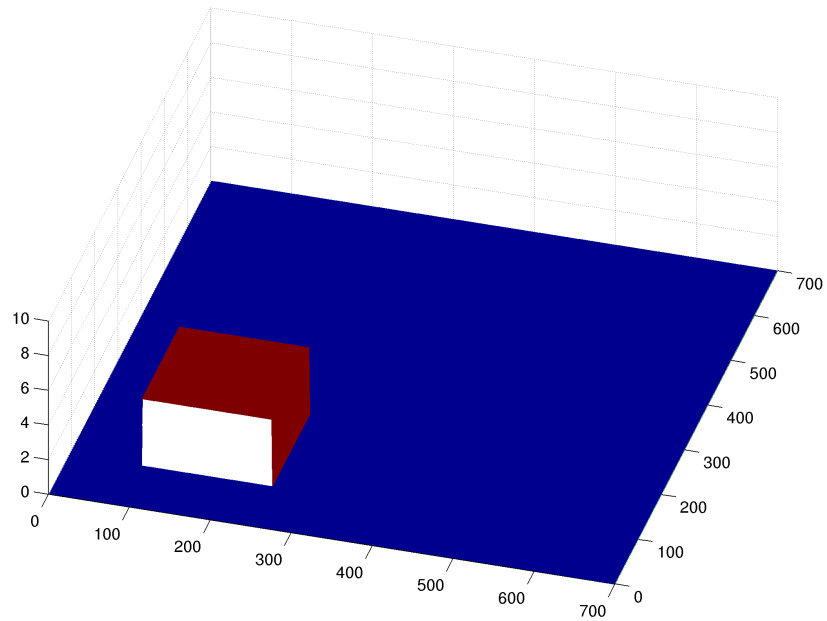


The first bump

Normalization example: square-root of data

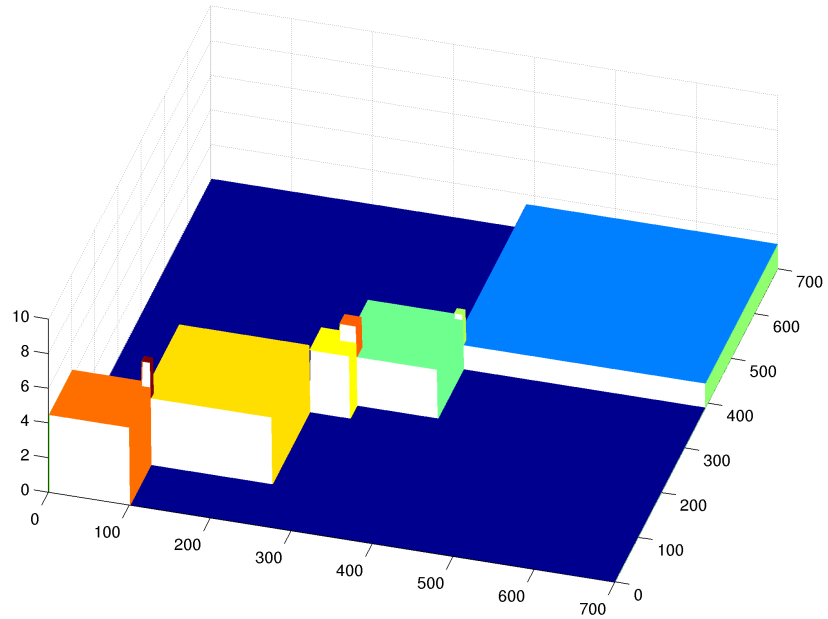


Data after taking element-wise square-root

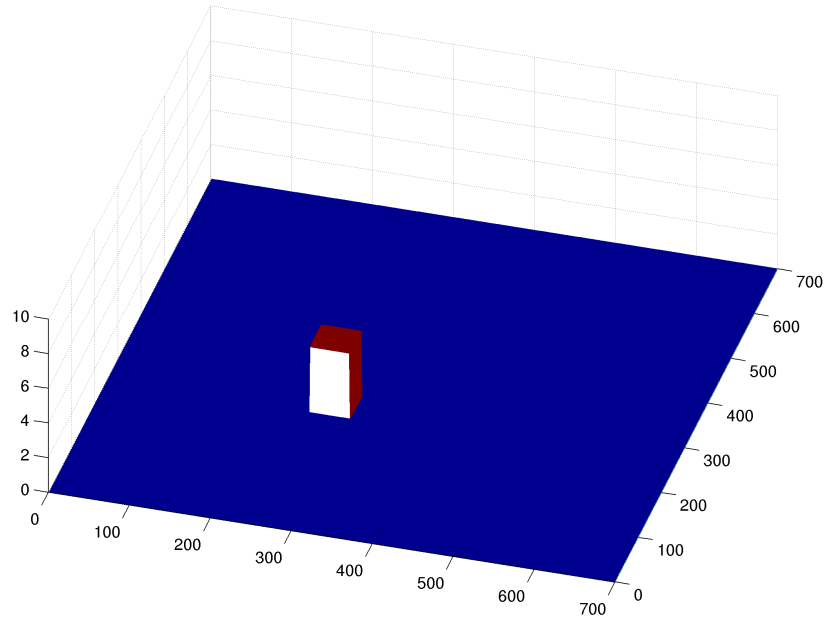


The second bump

Normalization example: square-root of data

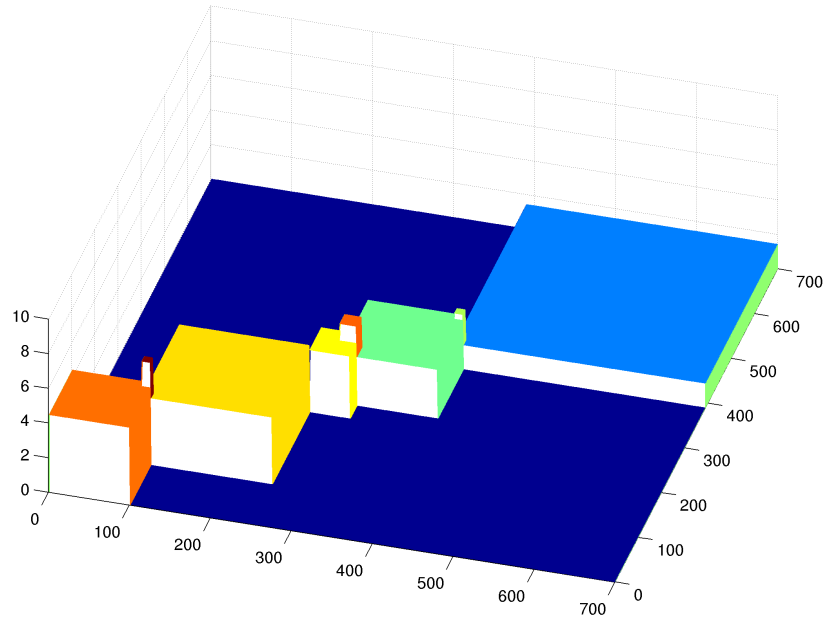


Data after taking element-wise square-root

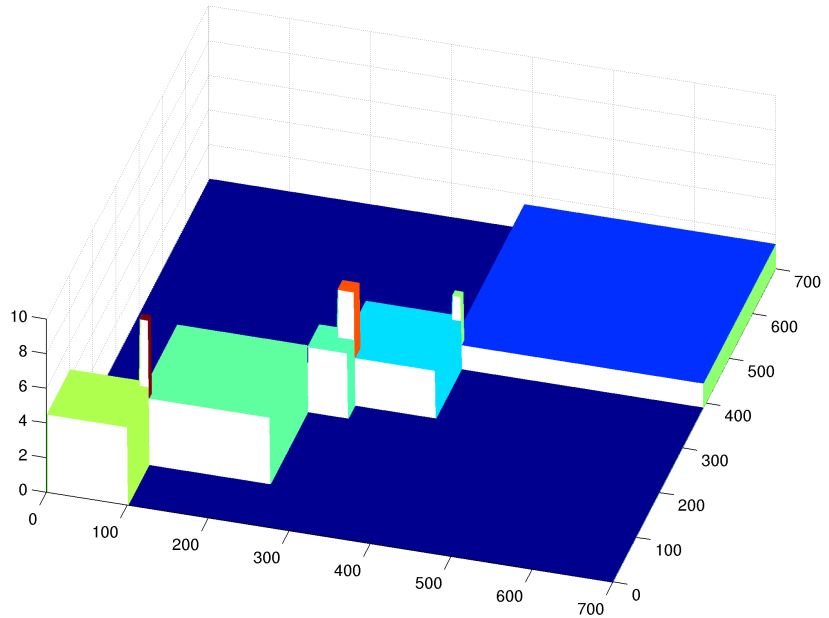


The third bump

Normalization example: square-root of data

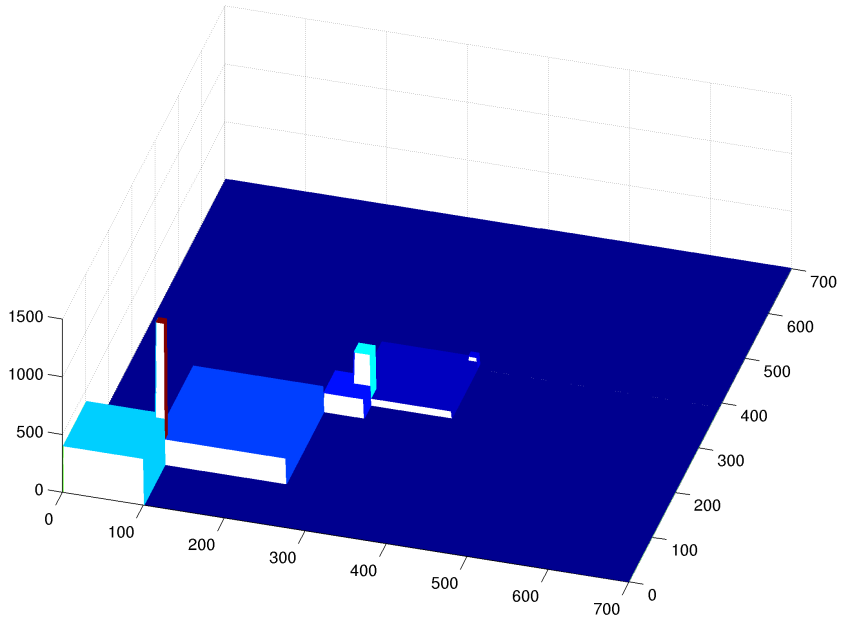


Data after taking element-wise square-root

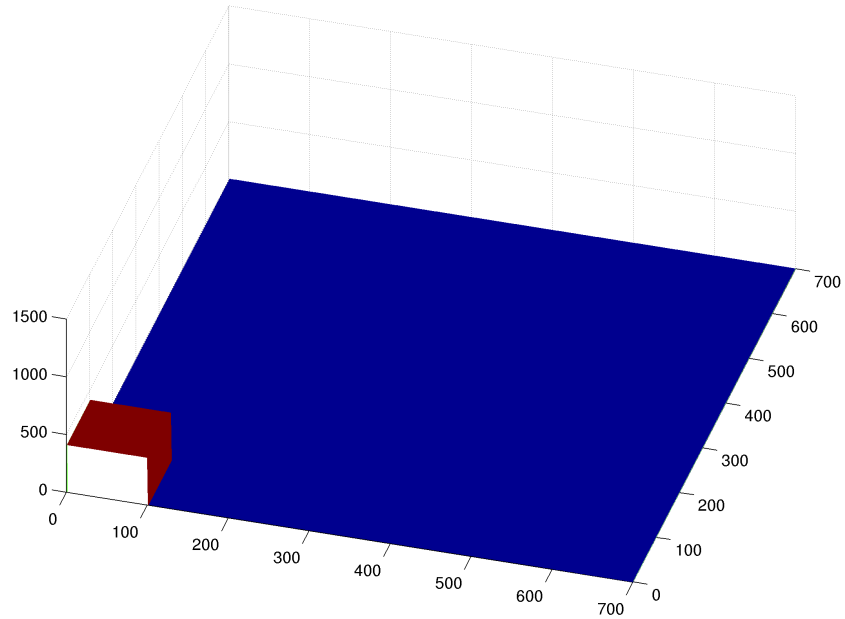


5-dimensional SDD

Normalization example: squared data

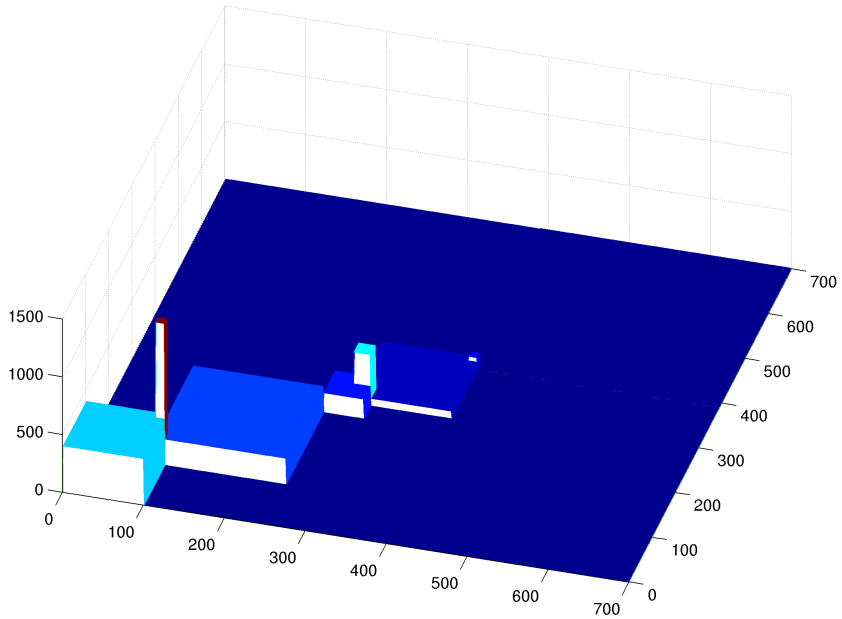


Squared data

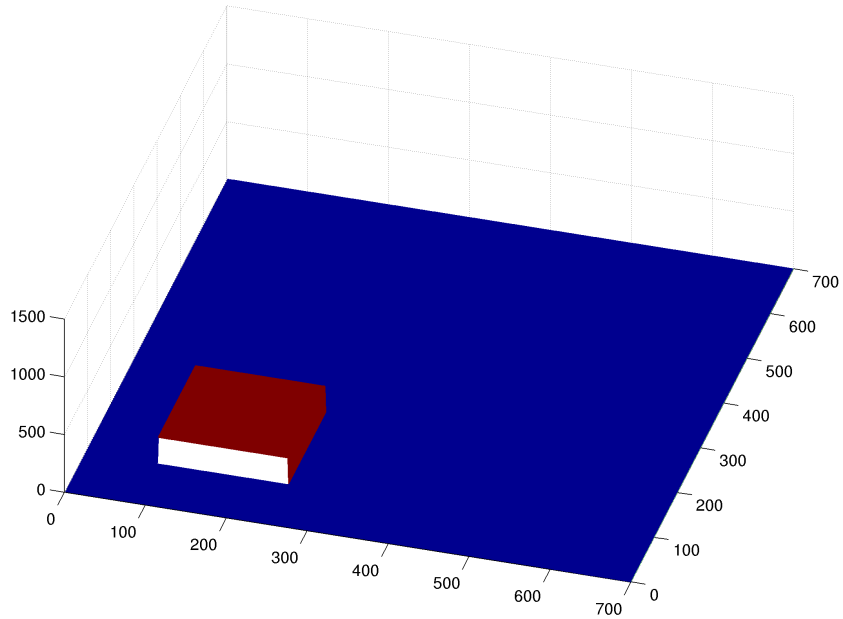


The first bump

Normalization example: squared data

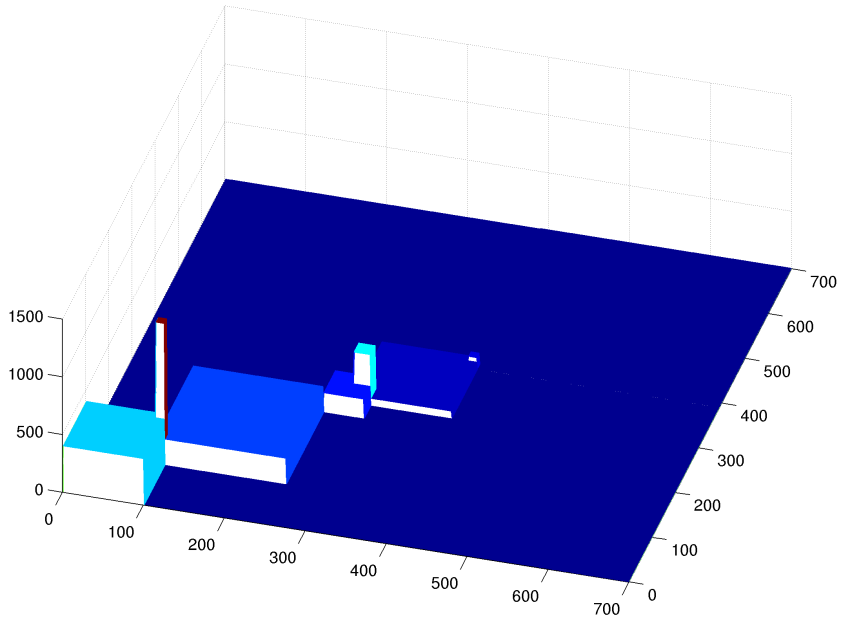


Squared data

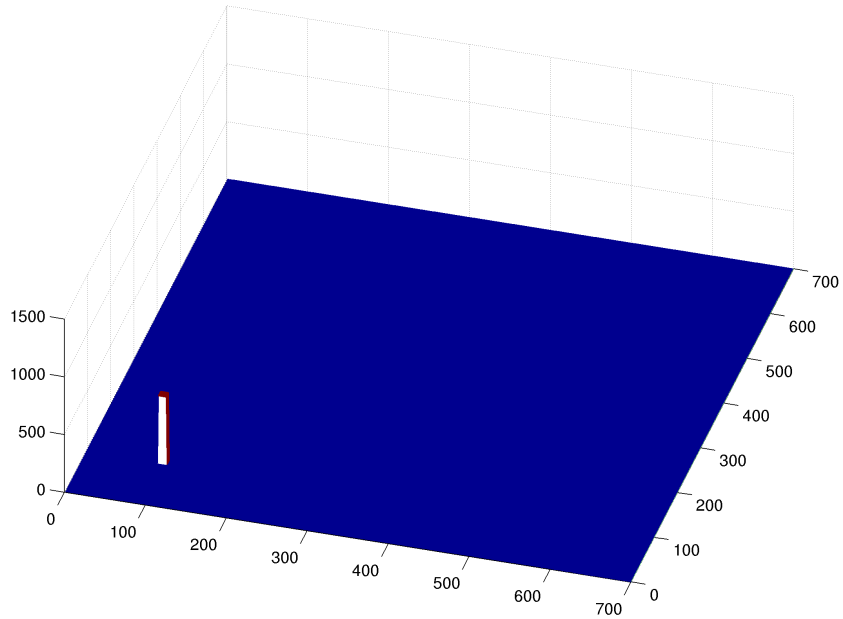


The second bump

Normalization example: squared data

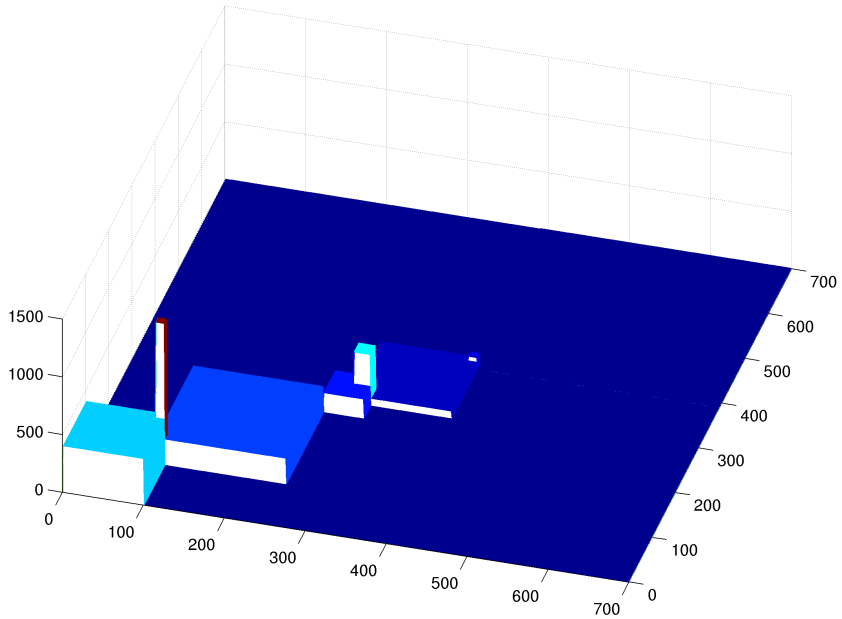


Squared data

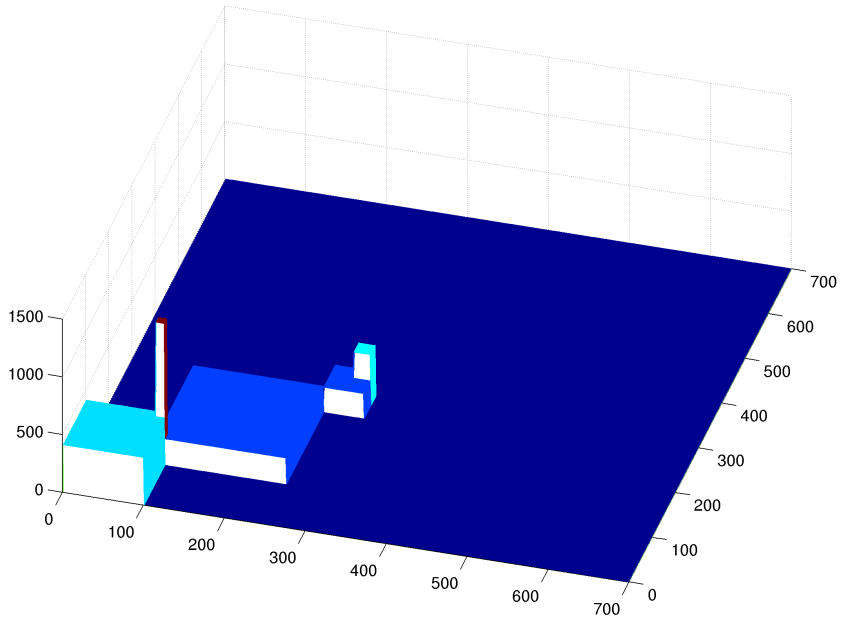


The third bump

Normalization example: squared data



Squared data



5-dimensional SDD

Outline

1. Hunting the Bump
2. Semi-Discrete Decomposition
3. The Algorithm
4. Applications
   - SDD alone
   - SVD + SDD
5. Wrap-Up

Outline

1. Hunting the Bump
2. Semi-Discrete Decomposition
3. The Algorithm
4. Applications
   - SDD alone
   - SVD + SDD
5. Wrap-Up

Clustering

- SDD performs a type of bi-clustering of the matrix
  - Every bump $d\boldsymbol{x}\boldsymbol{y}^T$ gives a cluster of rows $x_i \neq 0$ and a cluster of columns $y_j \neq 0$, together with 'centroid' d
  - This is **not** a partition clustering: the clusters can overlap and not every row or column has to belong to some cluster
- We can impose an ordering of the bumps based on the values of d_i
  - The algorithm usually returns the bumps in that order
- This ordering can be used to obtain a hierarchical clustering
  - First column of $\boldsymbol{X}$ clusters the rows of $\boldsymbol{A}$ into three sets $(-1, 0, 1)$, and same for the first column of $\boldsymbol{Y}$ and columns of $\boldsymbol{A}$
  - The second column of $\boldsymbol{X}$ splits the previous clusters again into three sets
    - ⋆ Some of these sets can be empty
  - And so on and so forth
- Distance between two objects in the hierarchical clustering is not the usual dendrogram depth, but depends on whether we use the 0 or ± 1 branches

Other applications

- Image compression (O'Leary & Peleg, 1983)
  - A grayscale image is compressed using its SDD
  - The original application, modern image compression techniques are better
- Latent topic models (Kolda & O'Leary, 1998)
  - Used similarly as SVD is used to compute LSA
  - Compute the SDD of the term-document matrix
- SDD to correlation matrices
  - A bump in the correlation matrix $\boldsymbol{A}\boldsymbol{A}^T$ corresponds to rows of $\boldsymbol{A}$ with similar values
  - A bump in $\boldsymbol{A}^T\boldsymbol{A}$ corresponds to columns with similar values

Outline

1. Hunting the Bump
2. Semi-Discrete Decomposition
3. The Algorithm
4. Applications
   - SDD alone
   - SVD + SDD
5. Wrap-Up

General approaches

- Most common way to combine SVD and SDD is to first use SVD to denoise the data and then to compute the SDD on the clean data
  - clean data = truncated SVD ($\boldsymbol{A}_k$)
  - SVD is good at finding global structure, SDD at finding local structure
- Another option is to first compute the $\boldsymbol{A}_k$ with SVD and then apply SDD to $\boldsymbol{A}_k \boldsymbol{A}_k^T$ (or $\boldsymbol{A}_k^T \boldsymbol{A}_k$)
  - SDD finds the objects with similar values
- The results can be visualized using the first 2–3 columns of $\boldsymbol{U}$ from SVD and the first layers of the hierarchical clustering from SDD

Classifying galaxies

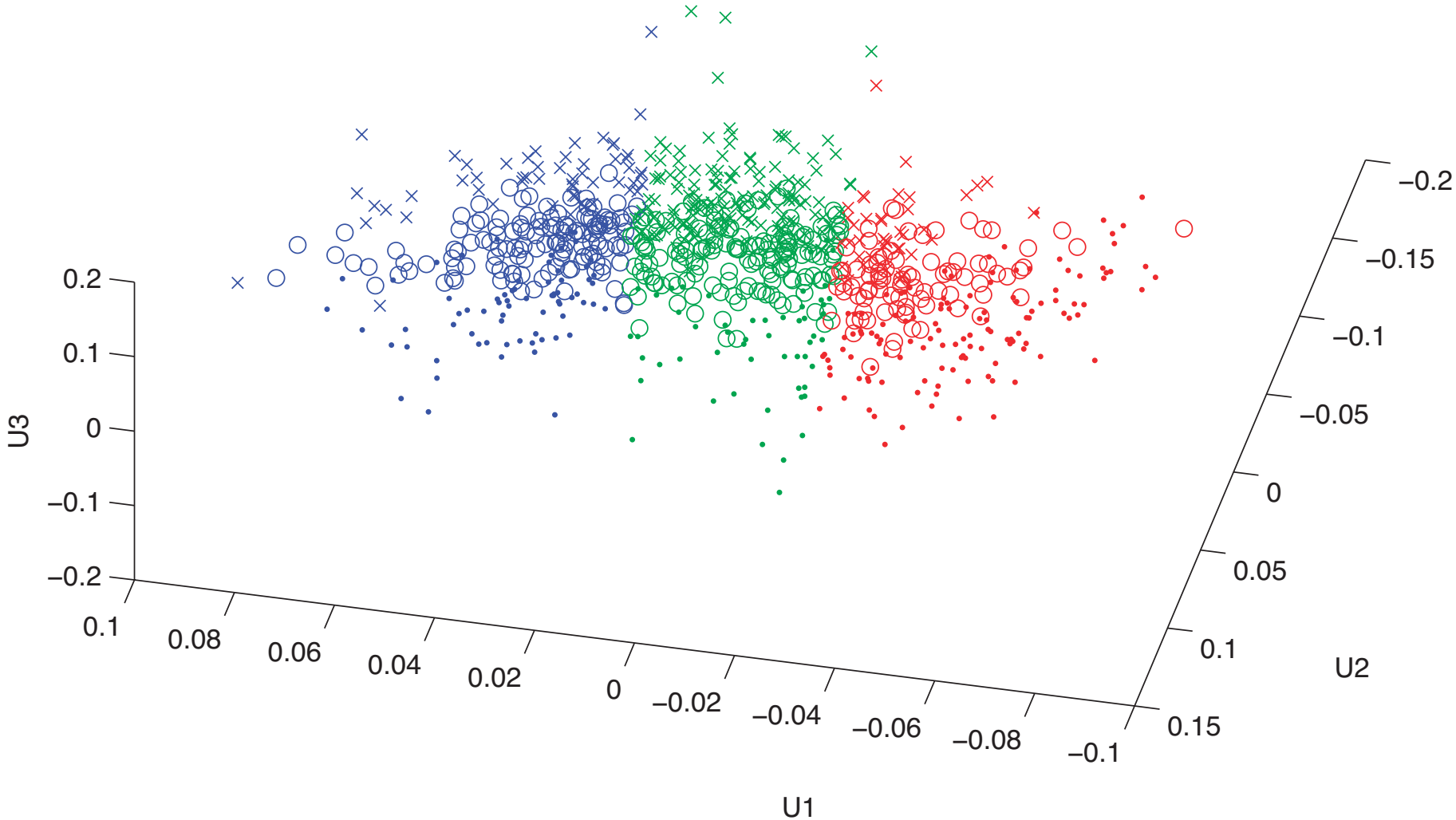


Figure 8. *Plot of the SVD of galaxy data, overlaid with the SDD classification.*

Finding minerals

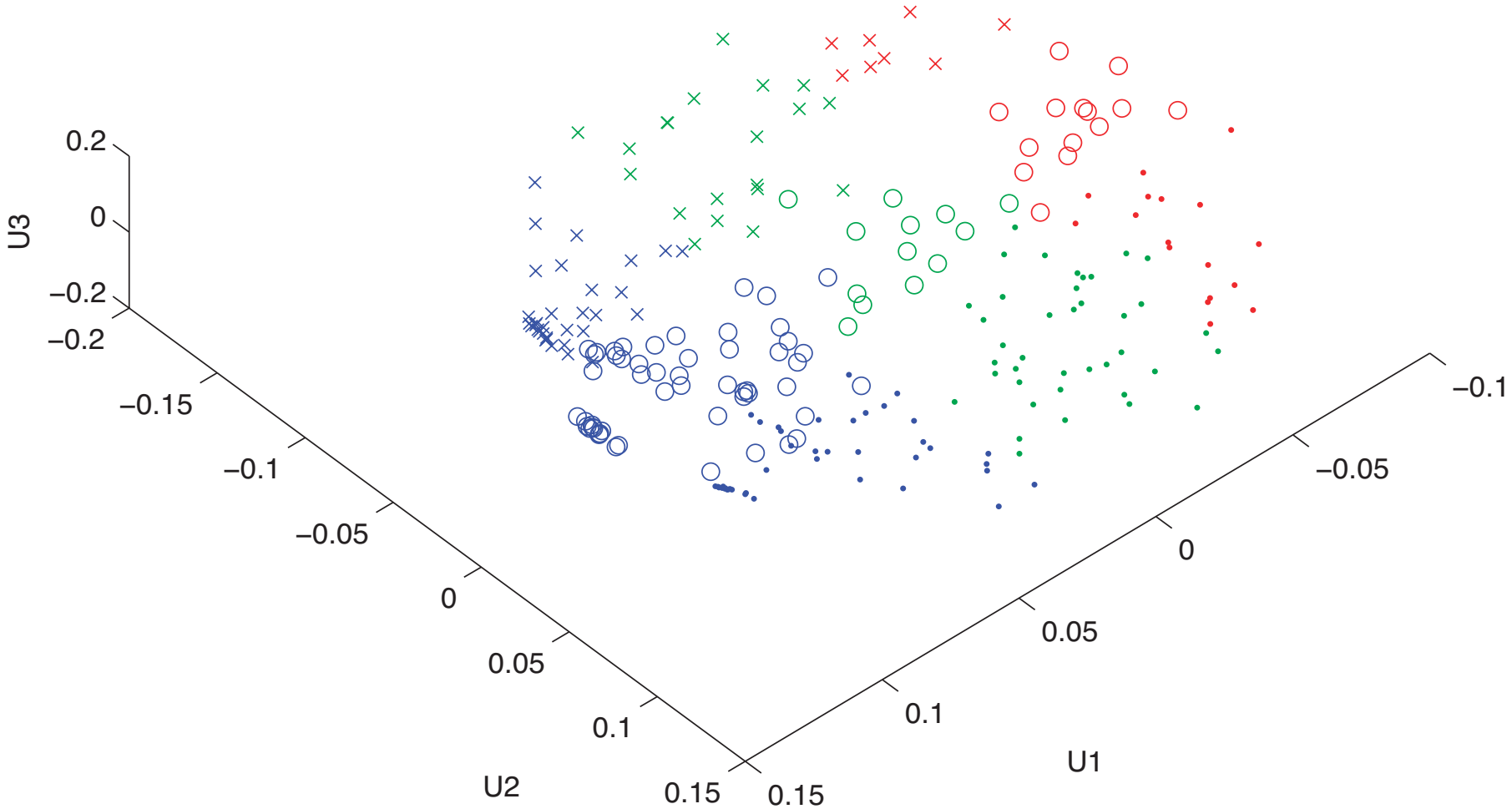


Figure 11. *Plot with position from the SVD, and color and shape labelling from the SDD.*

Clustering information from SDD added, first bump defines the colour, second the marker. Colour corresponds to the depth of the sample.

Outline

1. Hunting the Bump
2. Semi-Discrete Decomposition
3. The Algorithm
4. Applications
   - SDD alone
   - SVD + SDD
5. Wrap-Up

Lessons learned

- SDD finds the local areas of values with uniform magnitude
  $\rightarrow$ easier interpretation, 'orthogonal' view to SVD
- Finding SDD is hard and requires a heuristic
- Together SVD and SDD provide a strong analysis toolset

Suggested reading

- Skillicorn, Chapters 5 & 6
- Tamara G. Kolda & Diane P. O'Leary, 1998. A Semidiscrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval. *ACM Trans. Inf. Syst.* 16(4), pp. 322–346
  DOI: 10.1145/291128.291131
- Tamara G. Kolda & Diane P. O'Leary, 2000. Algorithm 805: Computation and Uses of the Semidiscrete Matrix Decomposition. *ACM Trans. Math. Software* 26(3), pp. 415–435
  DOI: 10.1145/358407.358424