# Using Local Neighborhoods to Find Subspace Clusters

*Emin Aksehirli*

with

Bart Goethals, Emmanuel Müller and Jilles Vreeken

Universiteit Antwerpen

# High Dimensional Data

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |

?

# High Dimensional Data

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ |
|---|---|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 | 1 | 0 |
| $o_2$ | 6 | 8 | 170 | 150 | 0 | 0 |
| $o_3$ | 7 | 6 | 140 | 160 | 1 | 0 |
| $o_4$ | 3000 | 4000 | 160 | 110 | 1 | 1 |
| $o_5$ | 5000 | 4200 | 150 | 5 | 0 | 0 |
| $o_6$ | 4000 | 3200 | 140 | 5000 | 1 | 0 |

# High Dimensional Data

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ |
|---|---|---|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 | 1 | 0 | Amsterdam |
| $o_2$ | 6 | 8 | 170 | 150 | 0 | 0 | New York |
| $o_3$ | 7 | 6 | 140 | 160 | 1 | 0 | Brussels |
| $o_4$ | 3000 | 4000 | 160 | 110 | 1 | 1 | Mumbai |
| $o_5$ | 5000 | 4200 | 150 | 5 | 0 | 0 | Istanbul |
| $o_6$ | 4000 | 3200 | 140 | 5000 | 1 | 0 | Shanghai |

# High Dimensional Data

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ |
|---|---|---|---|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 | 1 | 0 | Amsterdam | (1,23) |
| $o_2$ | 6 | 8 | 170 | 150 | 0 | 0 | New York | (9,1,2) |
| $o_3$ | 7 | 6 | 140 | 160 | 1 | 0 | Brussels | (23,45) |
| $o_4$ | 3000 | 4000 | 160 | 110 | 1 | 1 | Mumbai | (1275) |
| $o_5$ | 5000 | 4200 | 150 | 5 | 0 | 0 | Istanbul | (65,78) |
| $o_6$ | 4000 | 3200 | 140 | 5000 | 1 | 0 | Shanghai | (4,8,3) |

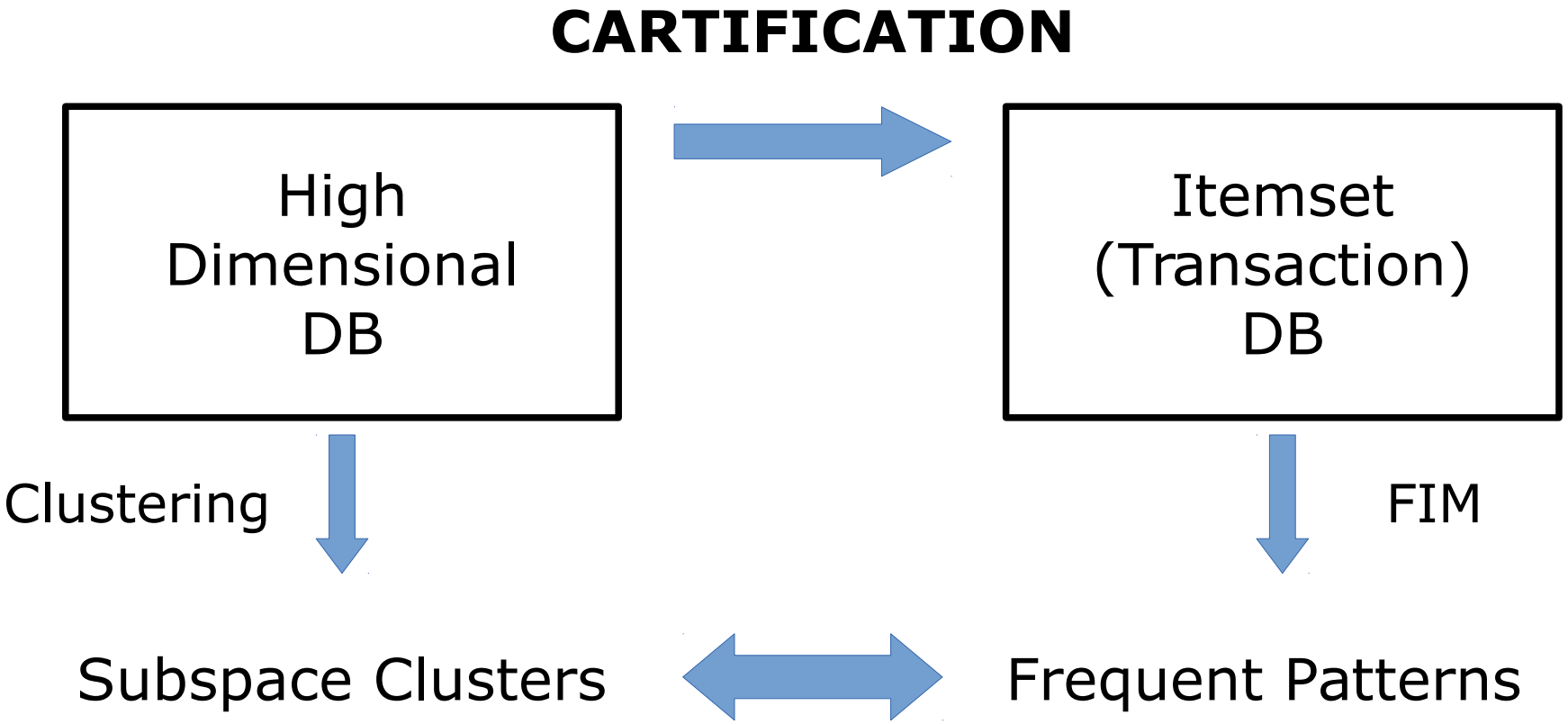
# High Dimensional Data

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ | $A_9$ |
|---|---|---|---|---|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 | 1 | 0 | Amsterdam | (1,23) |  |
| $o_2$ | 6 | 8 | 170 | 150 | 0 | 0 | New York | (9,1,2) |  |
| $o_3$ | 7 | 6 | 140 | 160 | 1 | 0 | Brussels | (23,45) |  |
| $o_4$ | 3000 | 4000 | 160 | 110 | 1 | 1 | Mumbai | (1275) |  |
| $o_5$ | 5000 | 4200 | 150 | 5 | 0 | 0 | Istanbul | (65,78) |  |
| $o_6$ | 4000 | 3200 | 140 | 5000 | 1 | 0 | Shanghai | (4,8,3) |  |

# Problem Setting

- Preserve local neighborhoods
- Combine different views on the data
- Produce explainable results

# Transformation

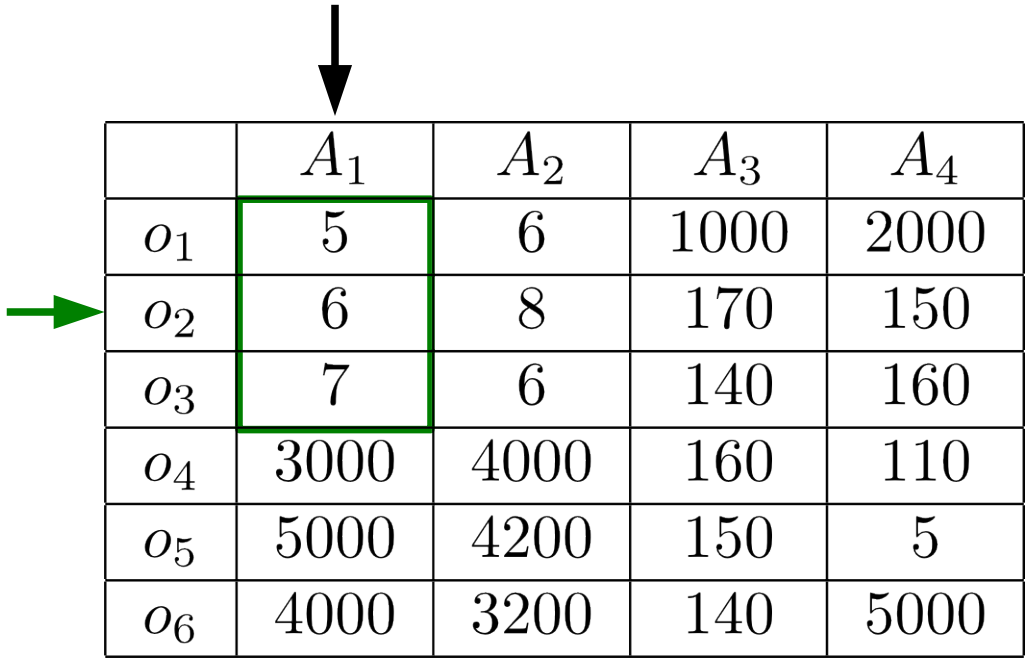


# Cartification

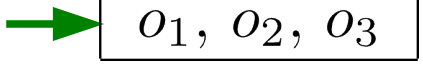
| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |

| $\mathcal{C}_{A_1}$ |
|---|
| $o_1, o_2, o_3$ |

# Cartification

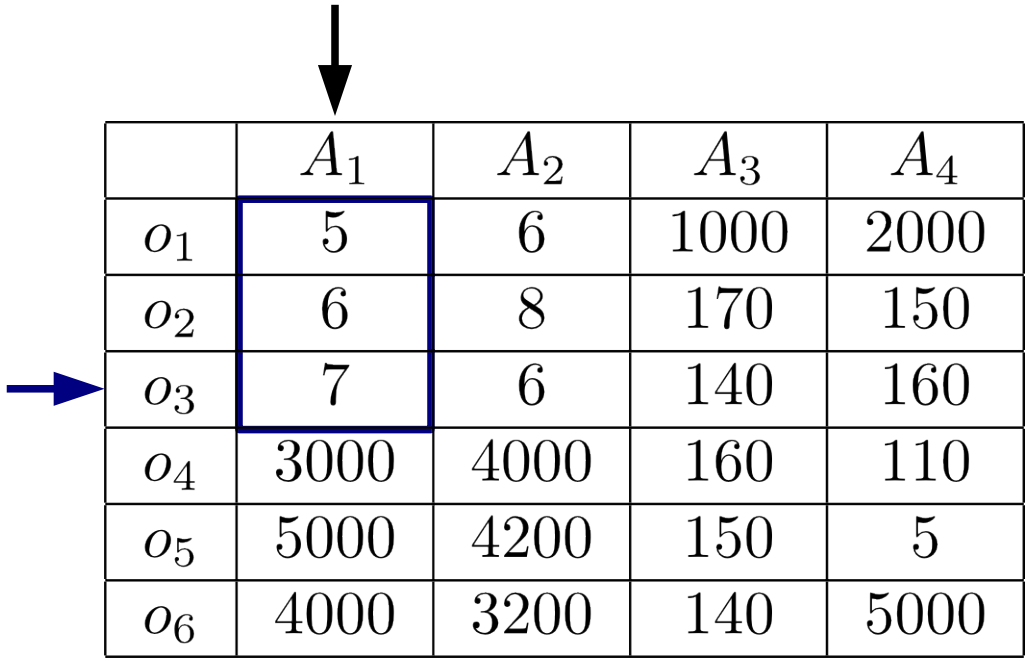


| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |

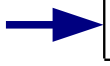


| $\mathcal{C}_{A_1}$ |
|---|
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |

# Cartification

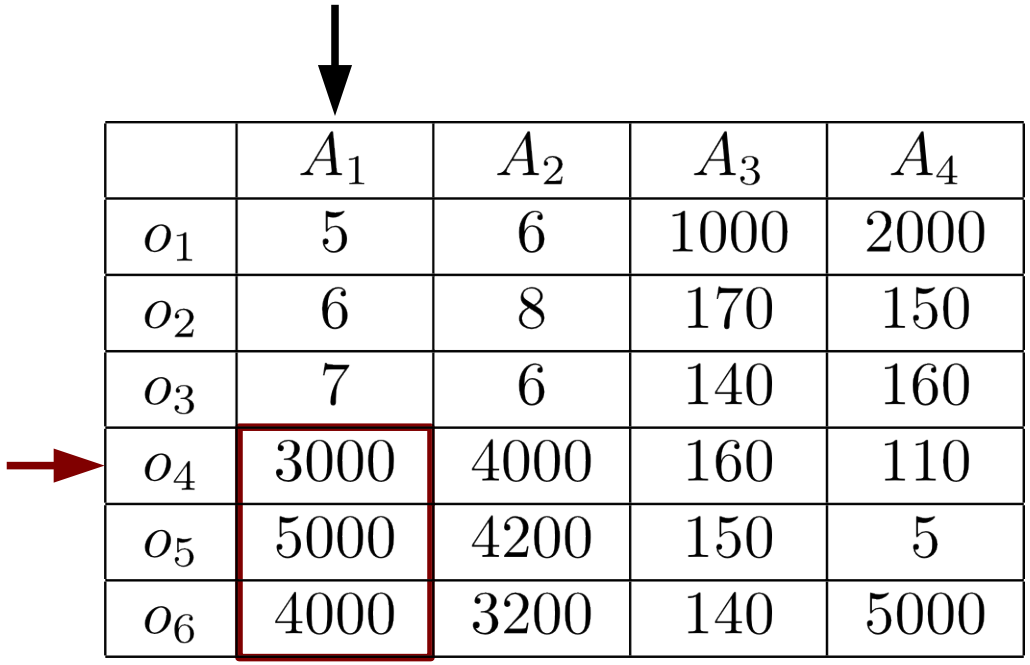


| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |

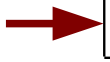


| $\mathcal{C}_{A_1}$ |
|---|
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |

# Cartification



| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |



| $\mathcal{C}_{A_1}$ |
|---|
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |
| $o_4, o_5, o_6$ |

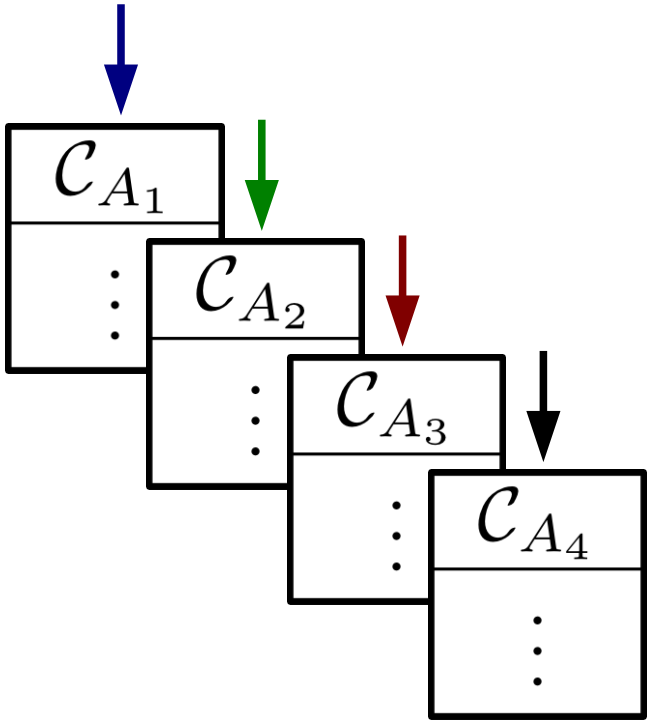
# Cartification

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |

| $\mathcal{C}_{A_1}$ |
|---|
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |
| $o_4, o_5, o_6$ |
| $o_4, o_5, o_6$ |
| $o_4, o_5, o_6$ |

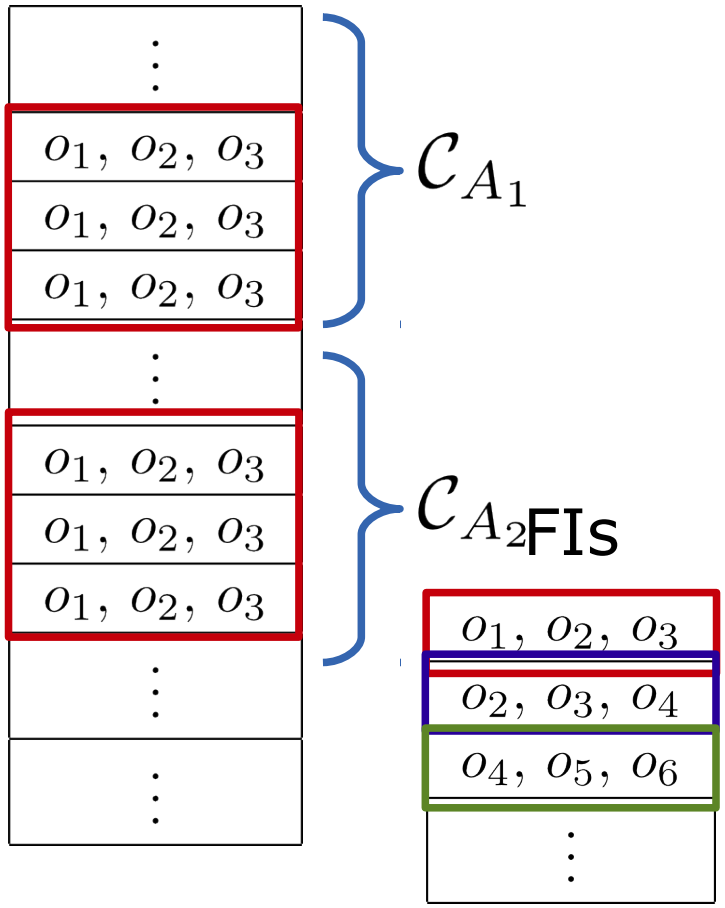
# Cartification

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |



# Frequent Itemset Mining

Cartified DB




Original DB

?

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |

# Cartification

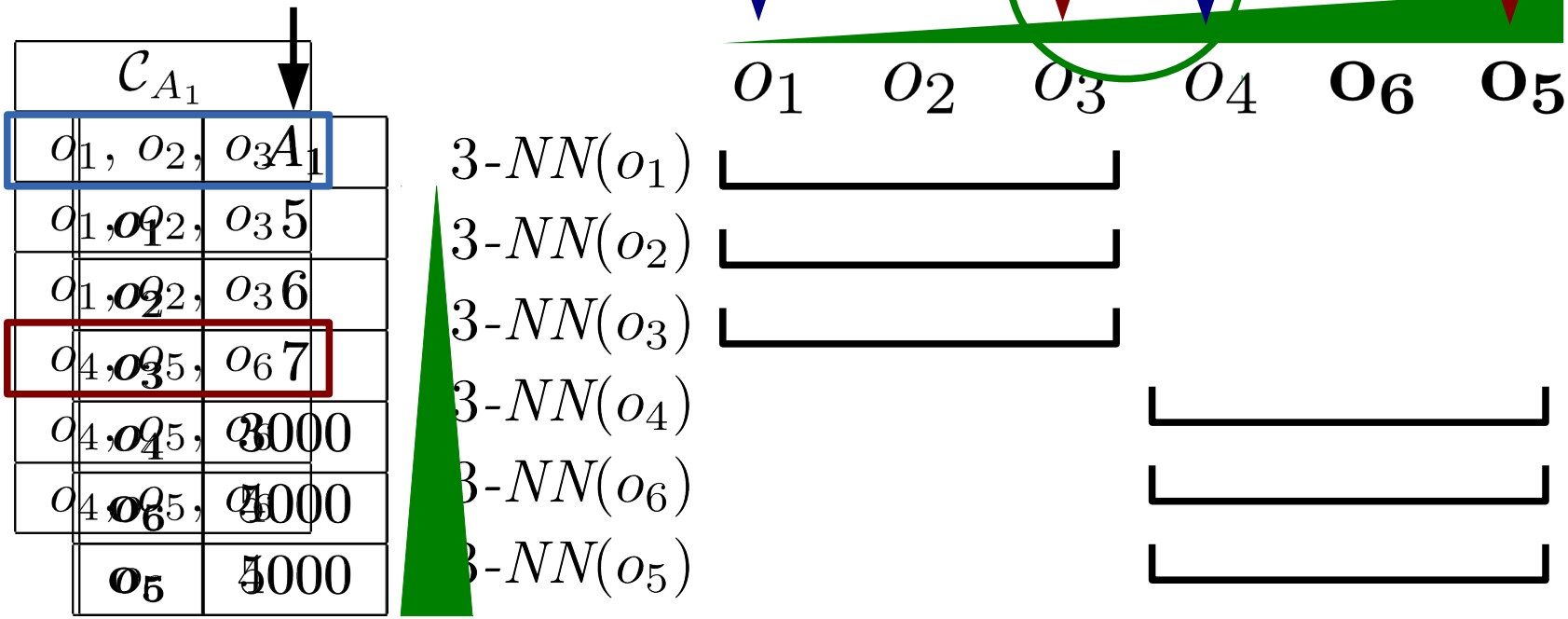
- Frequent Itemset Mining solves our problem.
- It is not scalable.

# Take 2

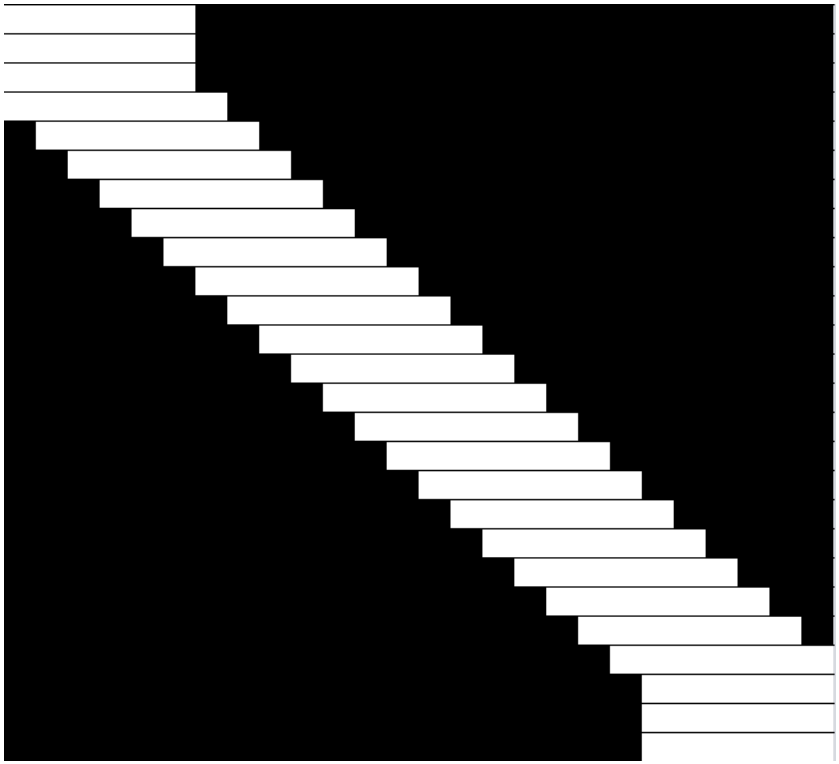
| | $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|---|
| $o_1$ | 5 | 6 | 1000 | 2000 |
| $o_2$ | 6 | 8 | 170 | 150 |
| $o_3$ | 7 | 6 | 140 | 160 |
| $o_4$ | 3000 | 4000 | 160 | 110 |
| $o_5$ | 5000 | 4200 | 150 | 5 |
| $o_6$ | 4000 | 3200 | 140 | 5000 |

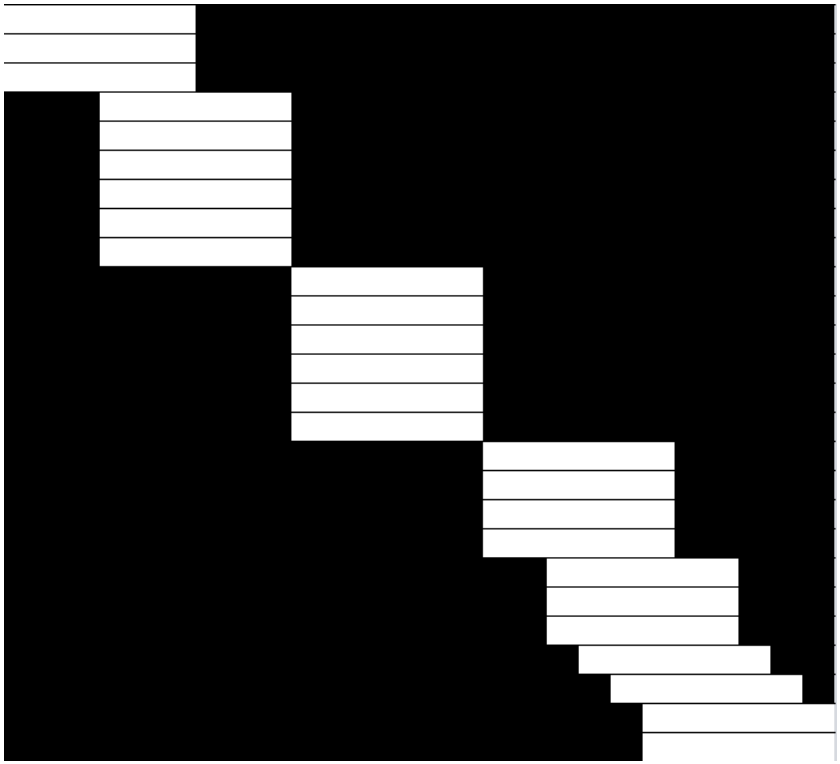
| $\mathcal{C}_{A_1}$ |
|---|
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |
| $o_1, o_2, o_3$ |
| $o_4, o_5, o_6$ |
| $o_4, o_5, o_6$ |
| $o_4, o_5, o_6$ |

# Take 2

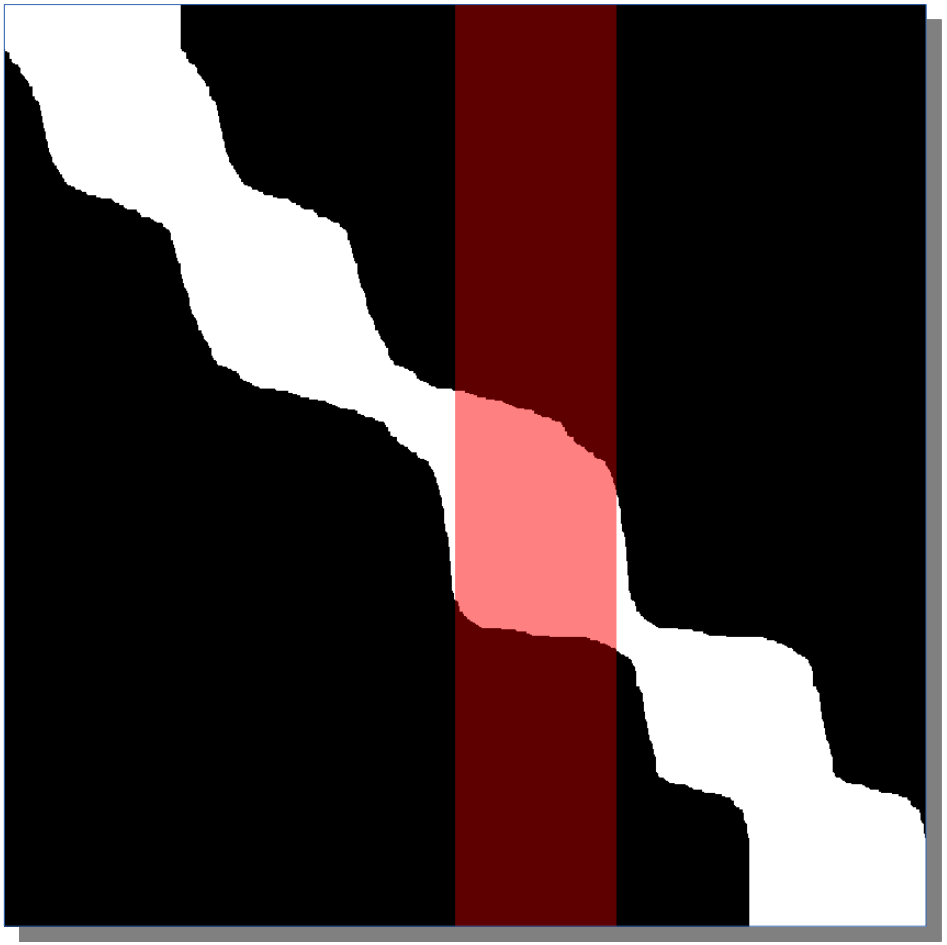

$\mathcal{C}_{A_1}$
$o_1, o_2, o_3$ $A_1$
$o_1, o_2, o_3$ 5
$o_1, o_2, o_3$ 6
$o_4, o_5, o_6$ 7
$o_4, o_5, o_6$ 3000
$o_4, o_5, o_6$ 4000
$o_5$ 4000
3-$NN(o_1)$
3-$NN(o_2)$
3-$NN(o_3)$
3-$NN(o_4)$
3-$NN(o_6)$
3-$NN(o_5)$
?
$o_1$ $o_2$ $o_3$ $o_4$ $o_6$ $o_5$

# Uniform vs. Clusters



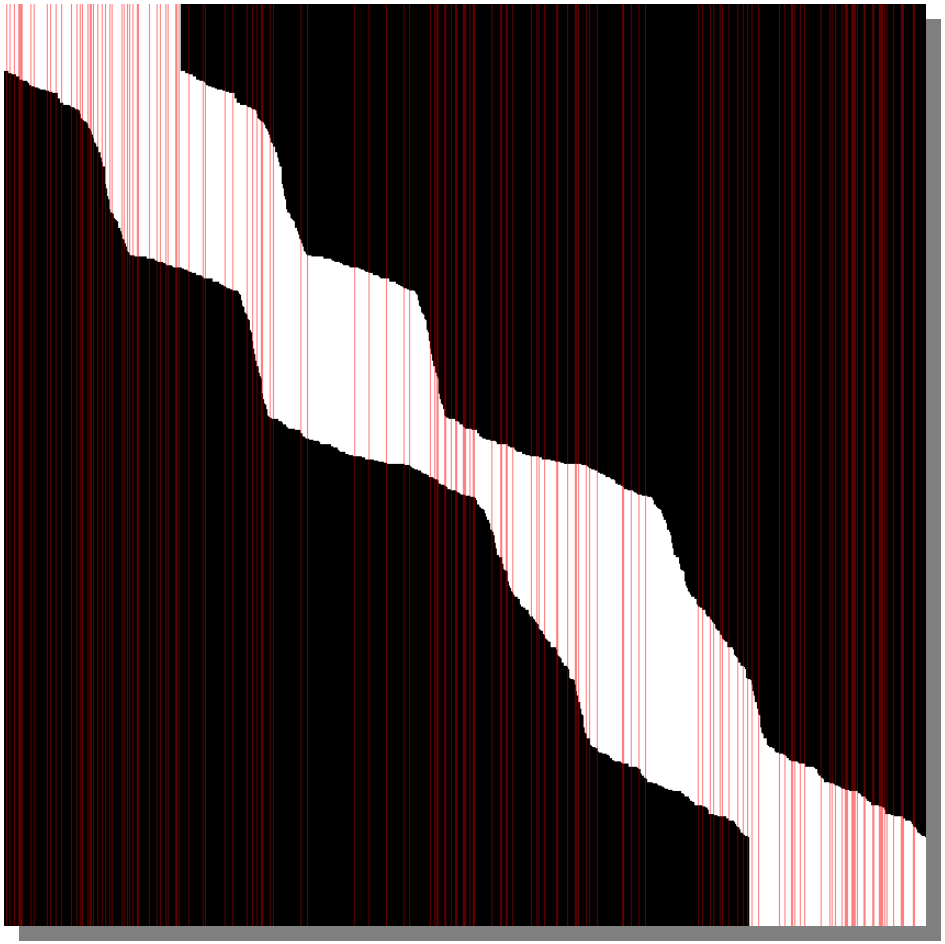


# Running example



Dim 1 ✓

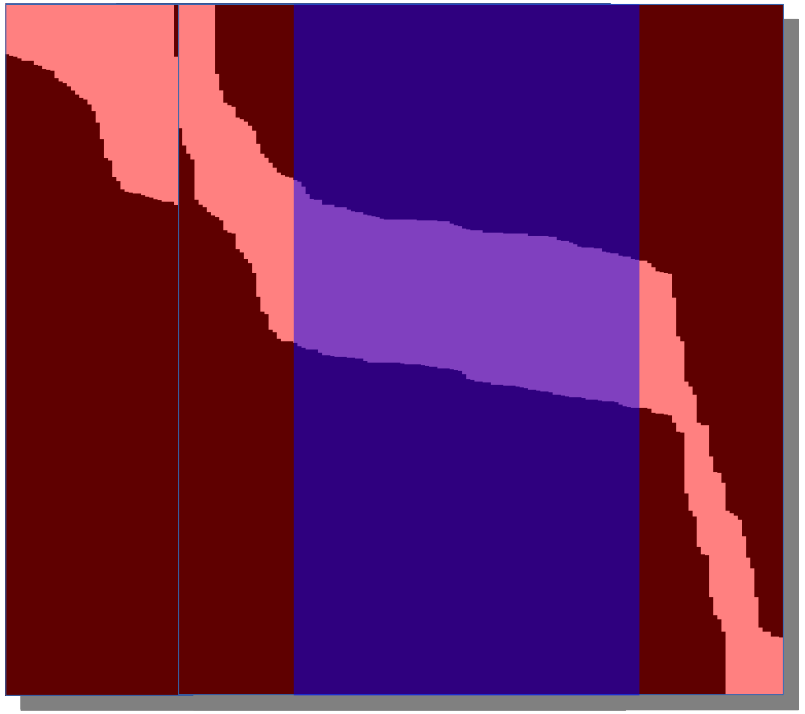
# Running example



Dim 1 ✓

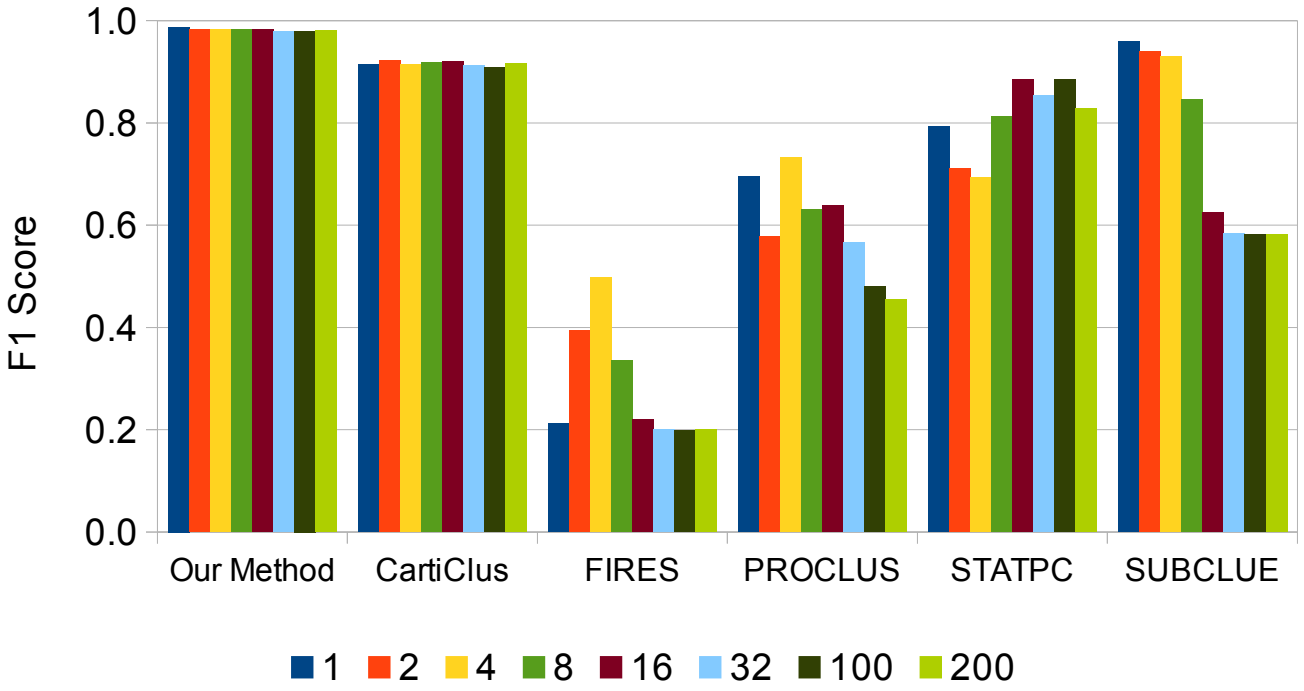
? Dim 2

# Running example

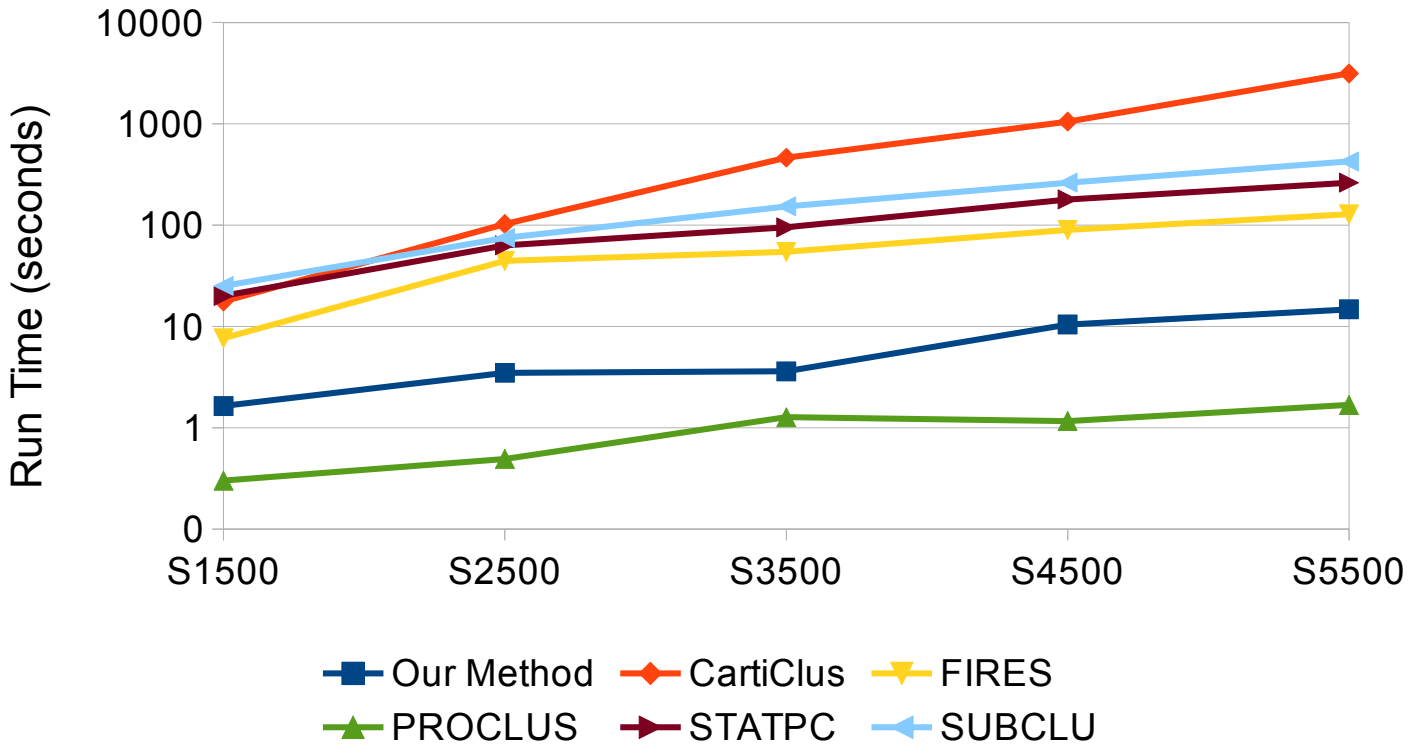


Dim 1 ✓
? Dim 2 ✗
? Dim 3 ✗
? Dim 4 ✓

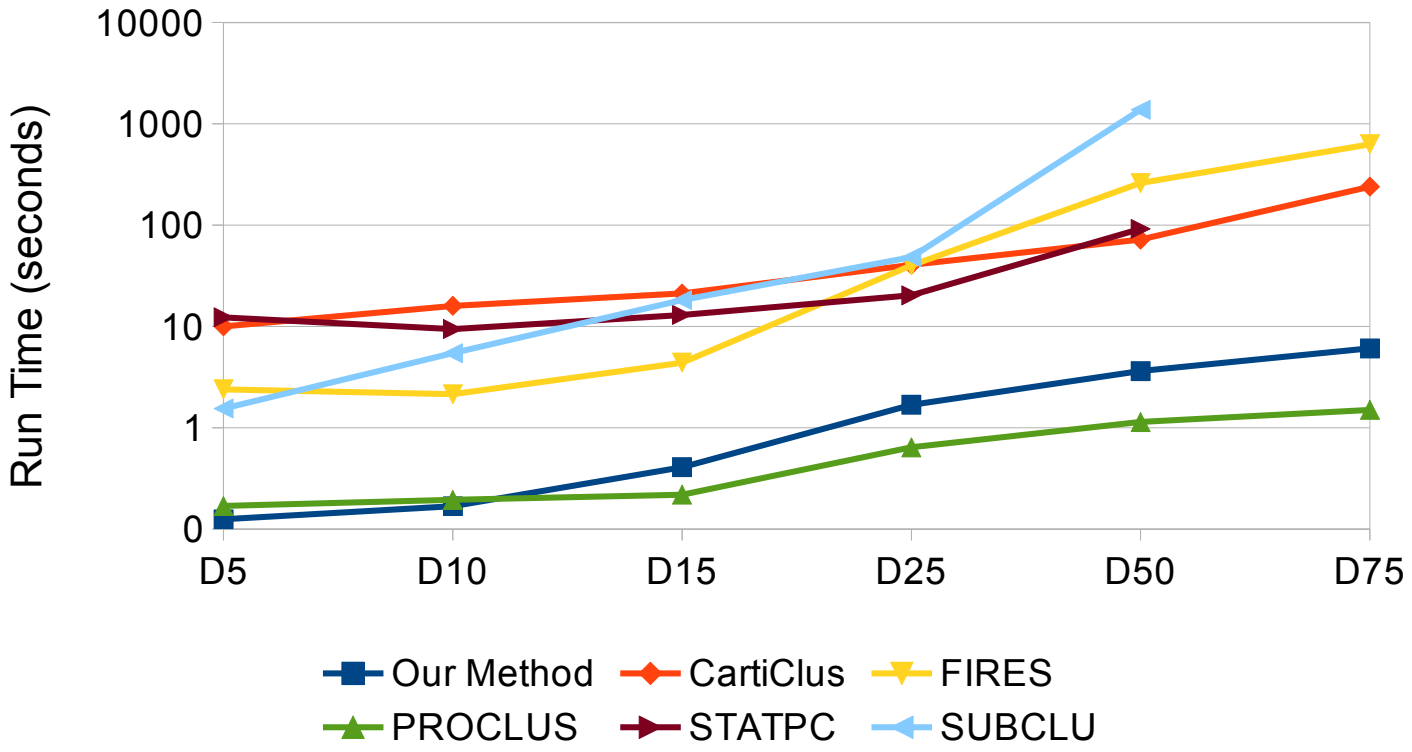
# Experiments



# Experiments



# Experiments



# Real World – MovieLens

| |
|---|
| Star Wars: A New Hope (a.k.a. Star Wars) (1977)<br>Star Wars: The Empire Strikes Back (1980)<br>Star Wars: Return of the Jedi (1983) |
| LotR: The Fellowship of the Ring, The (2001)<br>LotR: The Two Towers, The (2002)<br>LotR: The Return of the King, The (2003) |
| Back to the Future (1985)<br>Terminator, The (1984)<br>Terminator 2: Judgment Day (1991) |
| Die Hard (1988)<br>Terminator, The (1984)<br>Terminator 2: Judgment Day (1991) |
| Usual Suspects, The (1995)<br>Pulp Fiction (1994)<br>Silence of the Lambs, The (1991) |

# Real World - Movielens

| | |
|---|---|
| Star Wars: A New Hope (1977)<br>Star Wars: The Empire Strikes Back (1980)<br>Star Wars: Return of the Jedi (1983)<br>LotR: The Fellowship of the Ring, The (2001)<br>LotR: The Two Towers, The (2002)<br>LotR: The Return of the King, The (2003) | Brazil (1985)<br>Dr. Strangelove (1964)<br>Clockwork Orange, A (1971)<br>2001: A Space Odyssey (1968)<br>Blade Runner (1982)<br>Alien (1979) |
| Chinatown (1974)<br>Rear Window (1954)<br>North by Northwest (1959)<br>Vertigo (1958)<br>Psycho (1960)<br>Silence of the Lambs, The (1991) | Third Man, The (1949)<br>Citizen Kane (1941)<br>Godfather: Part II, The (1974)<br>Chinatown (1974)<br>Godfather, The (1972)<br>Taxi Driver (1976) |

# Conclusion

- Preserves neighborhood information
- Combines different similarity measures gracefully
- Finds relevant features and discards noise
- Fast
- Produce explainable results

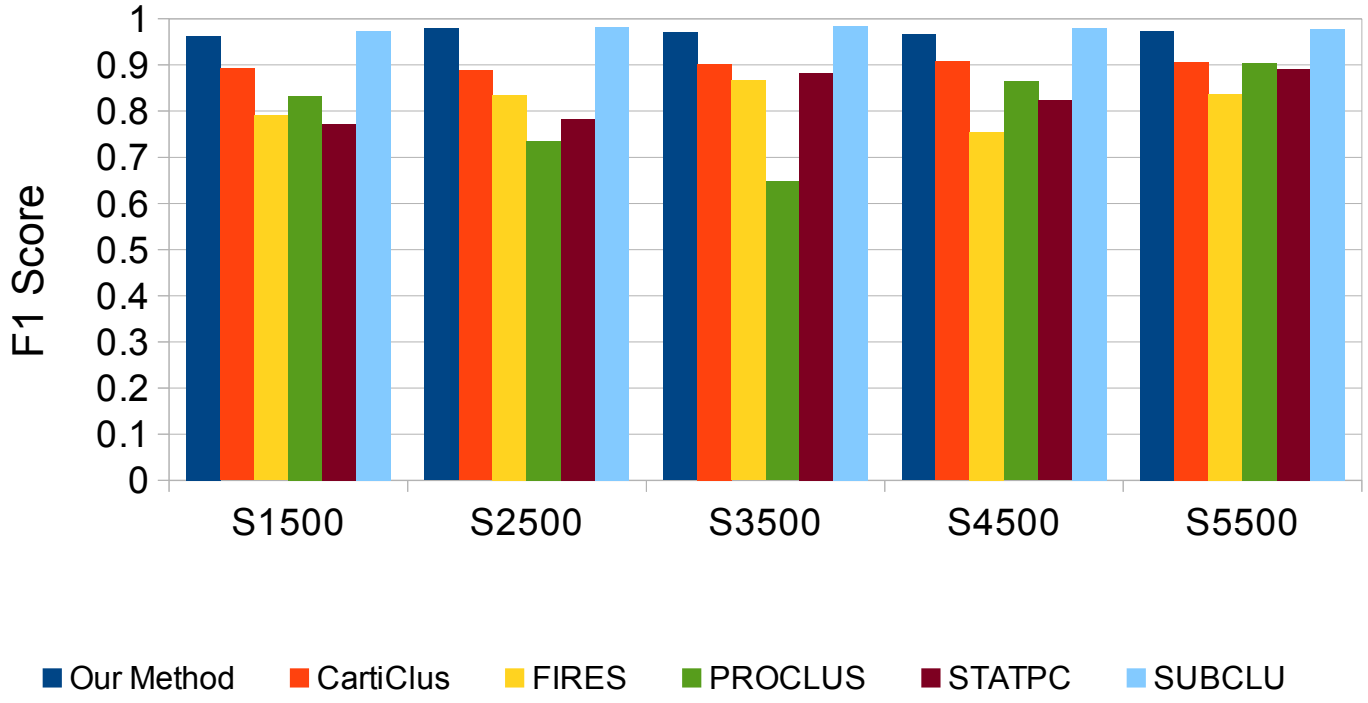
→ *Code and the data is available at our website.*

**Thank you!**

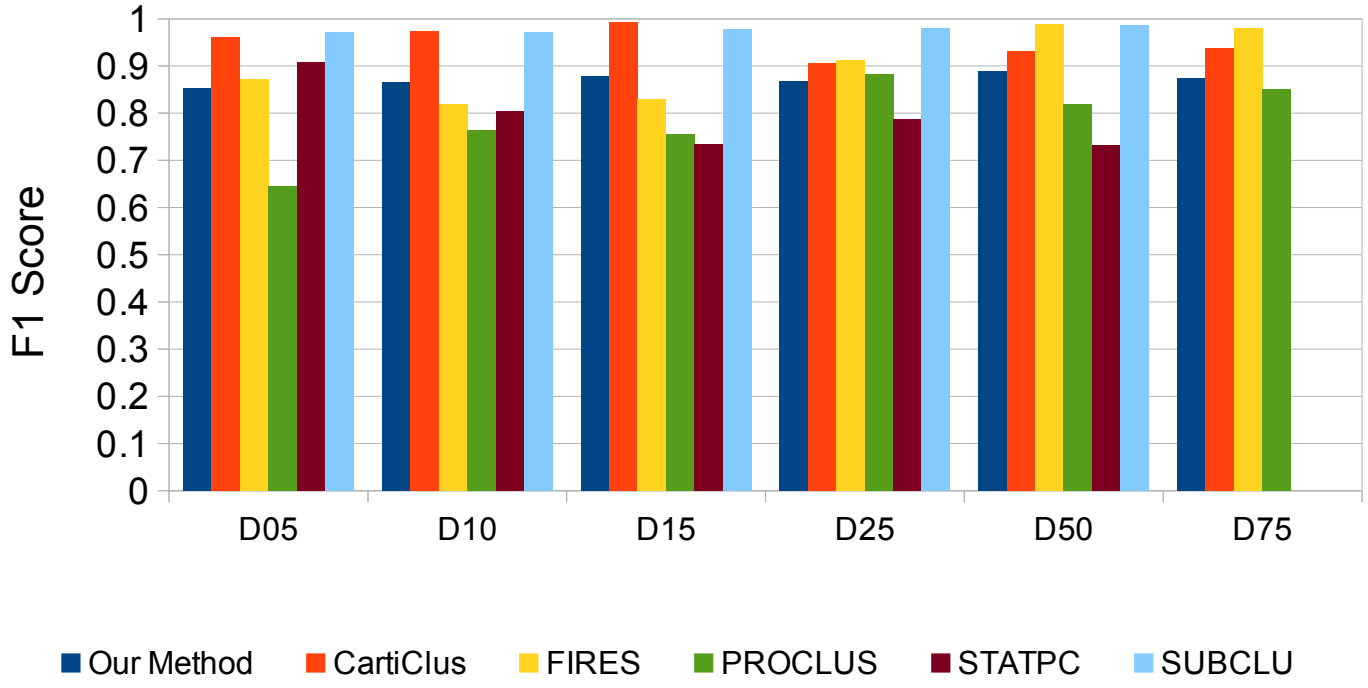
# Real World – Gene Expression

| | Alon | Nutt |
|---|---|---|
| **Our method** | 0.78 | 0.78 |
| **PROCLUS** | 0.46 | 0.49 |
| **FIRES** | 0.52 | 0.55 |
| **SUBCLU** | 0.58 | n/a |
| **STATPC** | n/a | n/a |
| **CartiClus** | n/a | n/a |
| *# of Objects* | 62 | 50 |
| *# of Dims* | 2000 | 1377 |

# More Experiments



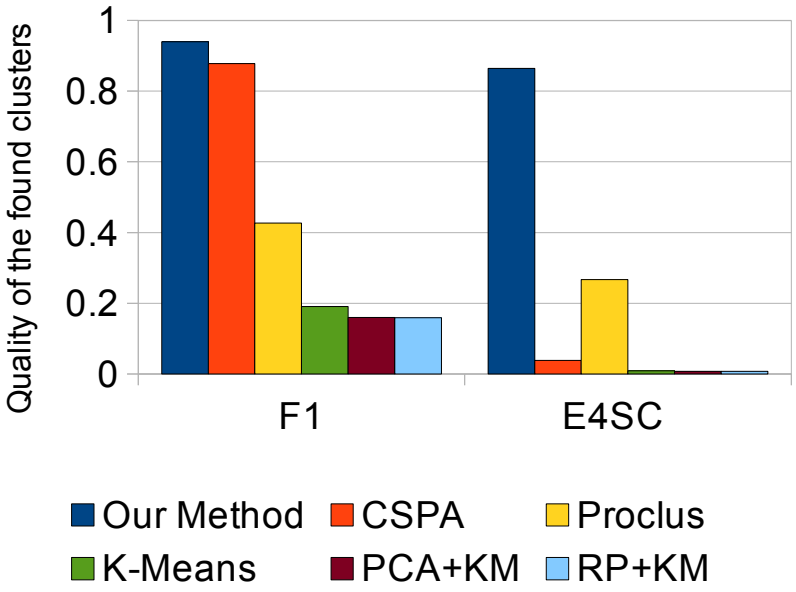
# More Experiments



# Experiments

- Evaluate:
  - **Subspace cluster detection**
  - Noise Robustness
  - Scalability
- Competitors:
  - Subspace clustering: **PROCLUS**
  - Clustering: **K-Means**
  - Dimensionality Reduction: **PCA** and **Random Projection**
  - Clustering Ensemble: **CSPA**

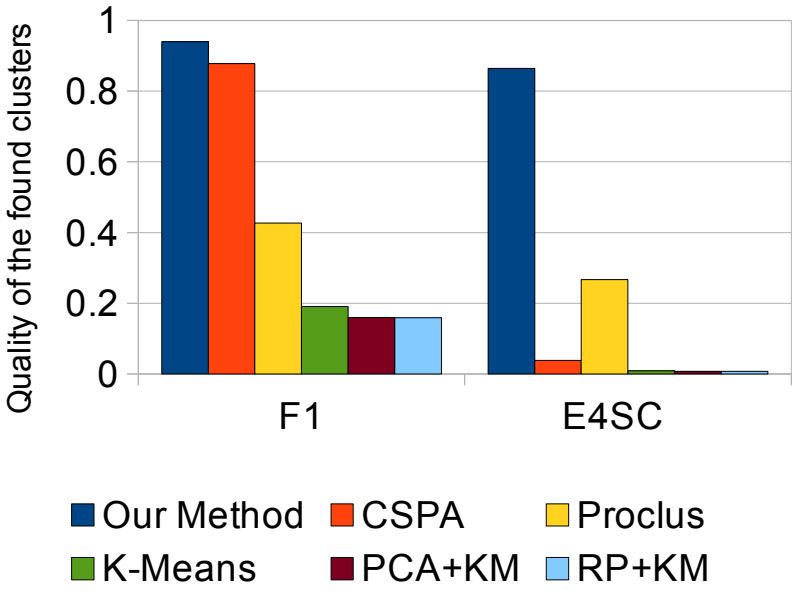
# Results




10 clusters in 10 dimensions

200 irrelevant dimensions

# Results




10 clusters in 10 dimensions

200 irrelevant dimensions

- Very effective on finding relevant dimensions.