Algebraic Techniques for Analysis of Large Discrete-Valued Datasets*

Mehmet Koyutürk[1], Ananth Grama[1],

& Naren Ramakrishnan[2]

1. Dept. of Computer Sciences, Purdue University

{koyuturk, ayg} @cs.purdue.edu

2. Dept. of Computer Sciences, Virginia Tech

naren@cs.vt.edu

*This work was supported in part by National Science Foundation grants EIA-9806741, ACI-9875899 and ACI9872101

Motivation

- Handling large discrete-valued datasets
  - Extracting relations between data items
  - Summarizing data in an error-bounded fashion
  - Clustering of data items
  - Finding concise interpretable representations for clustered data
- Applications
  - Association rule mining
  - Classification
  - Data partitioning & clustering
  - Data compression

Algebraic Model

- Sparse matrix representation
  - Each column corresponds to an item
  - Each row corresponds to an instance
- Document-Term matrix (Information Retrieval)
  - Columns: Terms
  - Rows: Documents
- Buyer-Item matrix (Data Mining)
  - Columns: Items
  - Rows: Transactions
- Rows contain patterns of interest!

Basic Idea

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} = \mathbf{x}\mathbf{y}^T$$

$\mathbf{x}$: presence vector
$\mathbf{y}$: pattern vector

- Not all such matrices are rank 1 (cannot be represented accurately as a single outer product)
- We must find the best outer product
  - Concise
  - Error-bounded

An Example

- Consider the universe of items
  - {bread, butter, milk, eggs, cereal}
- And grocery lists
  - {butter, milk, cereal}
  - {milk, cereal}
  - {eggs, cereal}
  - {bread, milk, cereal}
- These lists can be represented by a matrix as follows:

An Example (contd.)

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

- This rank-1 approximation can be interpreted as follows:
  - Item set {milk, cereal} is characteristic to three buyers
  - This is the most dominant pattern in the data

Rank-1 Approximation

- *Problem:* Given discrete matrix $\mathbf{A}_{mxn}$, find discrete vectors $\mathbf{x}_{mx1}$ and $\mathbf{y}_{nx1}$ to
- **Minimize** $||\mathbf{A}-\mathbf{x}\mathbf{y}^T||^2_F$ **,** the number of non-zeros in the error matrix
  - NP-hard!
- Assuming continuous space of vectors and using basic algebraic transformations, the above minimization reduces to:
- **Maximize** $(\mathbf{x}^T\mathbf{A}\mathbf{y})^2 / ||\mathbf{x}||^2||\mathbf{y}||^2$

Background

- Singular Value Decomposition (SVD) [Berry et.al., 1995]
  - Decompose matrix into $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T$
    - **U**, **V** orthogonal, Σ contains singular values
  - Decomposition based on underlying patterns
  - Latent Semantic Indexing (LSI)
- Semi-Discrete Decomposition (SDD) [Kolda & O'Leary, 2000]
  - Restrict entries of **U** and **V** to {-1,0,1}
  - Can perform as well as SVD in LSI using less than one-tenth the storage[Kolda & O'Leary, 1998]

Background (contd.)

- Centroid Decomposition [Chu & Funderlic, 2002]
  - Decomposition based on spatial clusters
    - Centroid corresponds to the collective trend of a cluster
  - Data characterized by correlation matrix
  - Centroid method: Linear time heuristic to discover clusters
  - Two drawbacks for discrete-attribute data
    - Continuous in nature
    - Computation of correlation matrix requires quadratic time

Background (contd.)

- Principal Direction Divisive Partitioning (PDDP) [Boley, 1998]
  - Recursively splits matrix based on principal direction of vectors(rows)
  - Does not force orthogonality
  - Takes advantage of sparsity
  - Assumes continuous space

Alternating Iterative Heuristic

- In continuous domain, the problem is:

$$\textbf{minimize } F(d,\mathbf{x},\mathbf{y})=\|\mathbf{A}-d\mathbf{x}\mathbf{y}^T\|_F^2$$

$$F(d,\mathbf{x},\mathbf{y})=\|A\|_F^2-2d\,\mathbf{x}^T\mathbf{A}\mathbf{y}+d^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2 \quad (1)$$

- Setting $\partial F/\partial d = 0$ gives us the minimum of this function at

$$d^*=\mathbf{x}^T\mathbf{A}\mathbf{y}/\|\mathbf{x}\|^2\|\mathbf{y}\|^2$$

  (for positive definite matrix **A**)
- Substituting d^* in (1), we get equivalent problem: **maximize** $(\mathbf{x}^T\mathbf{A}\mathbf{y})^2 / \|\mathbf{x}\|^2\|\mathbf{y}\|^2$
- This is the optimization metric used in SDD's alternating iterative heuristic

Alternating Iterative Heuristic

- Approximate binary optimization metric to that of continuous problem
- Set $\mathbf{s}=\mathbf{Ay}/||\mathbf{y}||^2$, maximize $(\mathbf{x}^T\mathbf{s})^2/||\mathbf{x}||^2$
- This can be done by sorting s in descending order and assigning 1's to components of x in a greedy fashion
- Optimistic, works well on very sparse data

- Example

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{aligned}
&\mathbf{y}_0 = [1\ 0\ 0\ 0] \\
&\Rightarrow \mathbf{s}^x{}_0 = \mathbf{Ay} = [1\ 1\ 0]^T \\
&\Rightarrow \mathbf{x}_0 = [1\ 1\ 0]^T \\
&\Rightarrow \mathbf{s}^y{}_0 = \mathrm{A}^T\mathrm{y} = [2\ 2\ 0\ 0]^T
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow \mathbf{y}_1 = [1\ 1\ 0\ 0] \\
&\Rightarrow \mathbf{s}^x{}_1 = \mathbf{Ay} = [2\ 2\ 0]^T \\
&\Rightarrow \mathbf{x}_1 = [1\ 1\ 0]^T
\end{aligned}$$

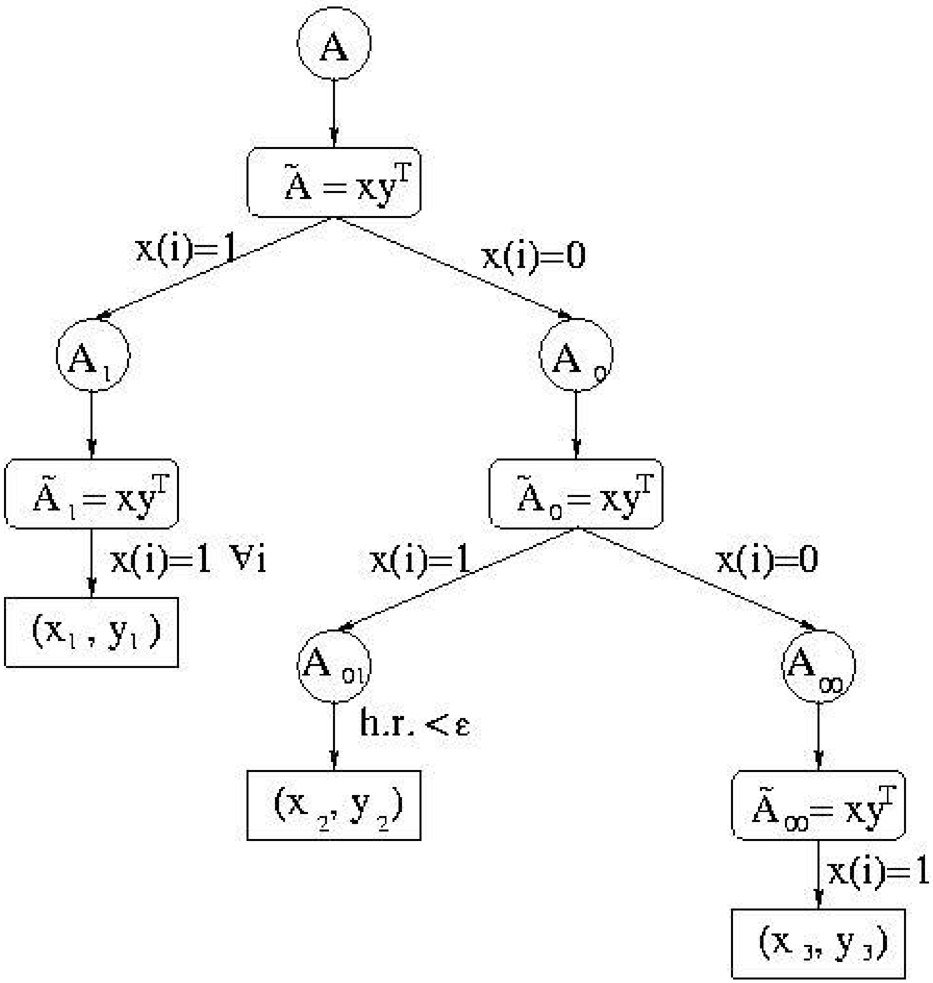
Initialization of pattern vector

- Crucial to find appropriate local optima
- Must be performed in at most $q(nz(\mathbf{A}))$ time
- Some possible schemes
  - **Center:** Initialize **y** as the centroid of rows, obviously cannot discover a cluster.
  - **Separator:** Bipartition rows on a dimension, set center of one group as initial pattern vector.
  - **Greedy graph growing:** Bipartition rows with starting from one row and growing a cluster centered on that row in a greedy manner, set center of that cluster as initial pattern vector.
  - ☑ **Neighborhood:** Randomly select a row, identify set of all rows that share a column with it, set center of this set as initial pattern vector. Aims at discovering smaller clusters, more successful.

Recursive Algorithm

- At any step, given rank-one approximation $\mathbf{A} \approx \mathbf{x}\mathbf{y}^T$, split $\mathbf{A}$ to $\mathbf{A}_1$ and $\mathbf{A}_0$ based on rows
  - if $\mathbf{x}_i = 1$ row i goes into $\mathbf{A}_1$
  - if $\mathbf{x}_i = 0$ row i goes into $\mathbf{A}_0$
- Stop when
  - Hamming radius of $\mathbf{A}_1$ is less then some threshold
  - all rows of $\mathbf{A}$ are present in $\mathbf{A}_1$
    - if Hamming radius of $\mathbf{A}_1$ greater than threshold, partition based on hamming distances to pattern vector and recurse

Recursive Algorithm



- Example: set $\varepsilon=1$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Rank-1 Appx.:

$\mathbf{y} = [1\ 1\ 1\ 0]$

$\mathbf{x} = [1\ 1\ 1]^T$

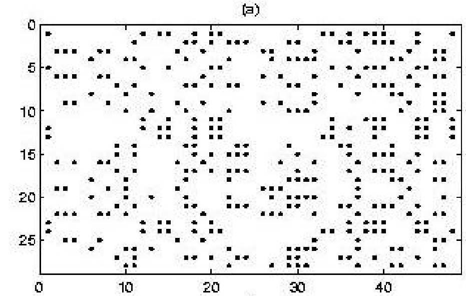
$\Rightarrow$ h.r. $= 2 > \varepsilon$

$$\Rightarrow \quad \mathbf{A} \to \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}$$

Effectiveness of Analysis

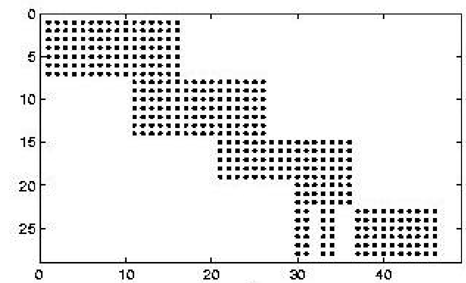
Input: 4 uniform patterns intersecting pairwise, 1 pattern on each row (overlapping patterns of this nature are particularly challenging for many related techniques)

(a)



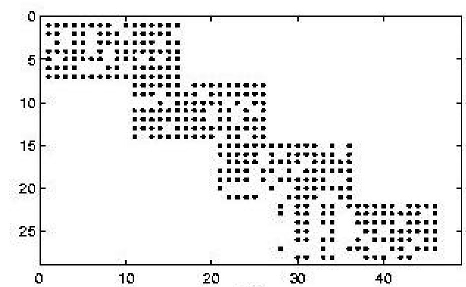
(i)

Detected patterns



(ii)

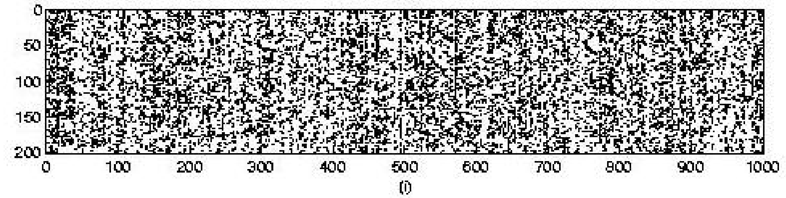
Input permuted to demonstrate strength of detected patterns



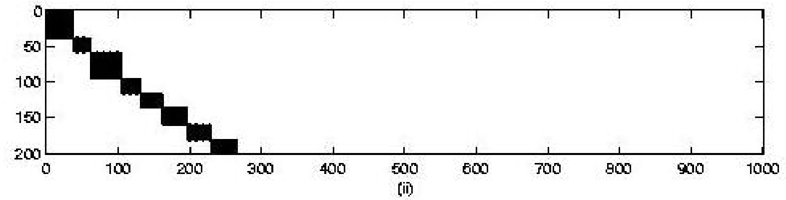
(iii)

Effectiveness of Analysis

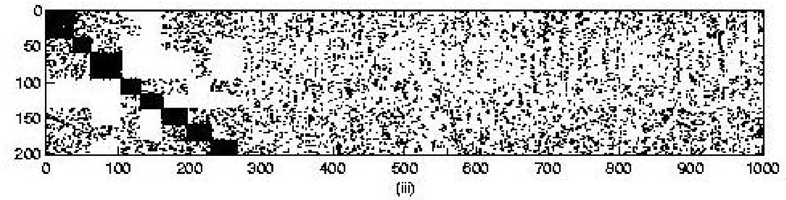
Input: 10 gaussian patterns, 1 pattern on each row



Detected patterns

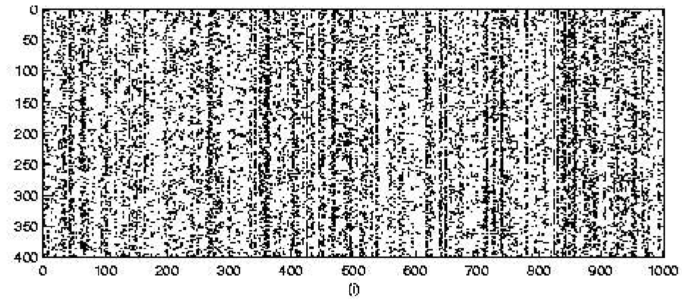


Permuted input

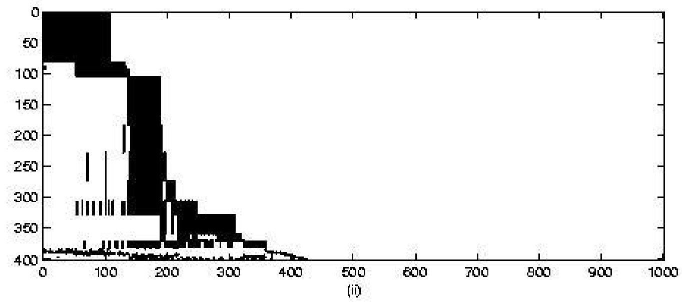


Effectiveness of Analysis

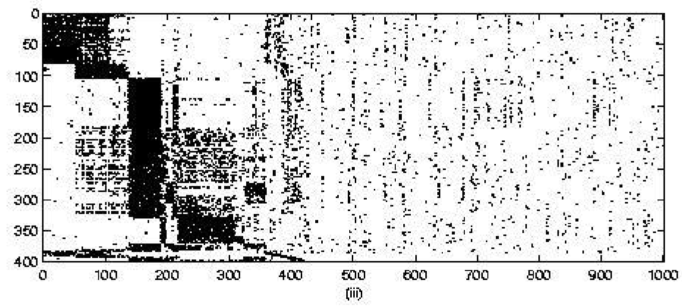
Input: 20 gaussian patterns, 2 patterns on each row



Detected patterns



Permuted input



Application to Data Mining

- Used for preprocessing data to reduce number of transactions for association rule mining
- Construct matrix **A**:
  - Rows correspond to transactions
  - Columns correspond to items
- Decompose **A** into $\mathbf{XY}^T$
  - **Y** is the compressed transaction set
  - Each transaction is weighted by the number of rows containing the pattern (# of non-zeros in the corresponding row of **X**)

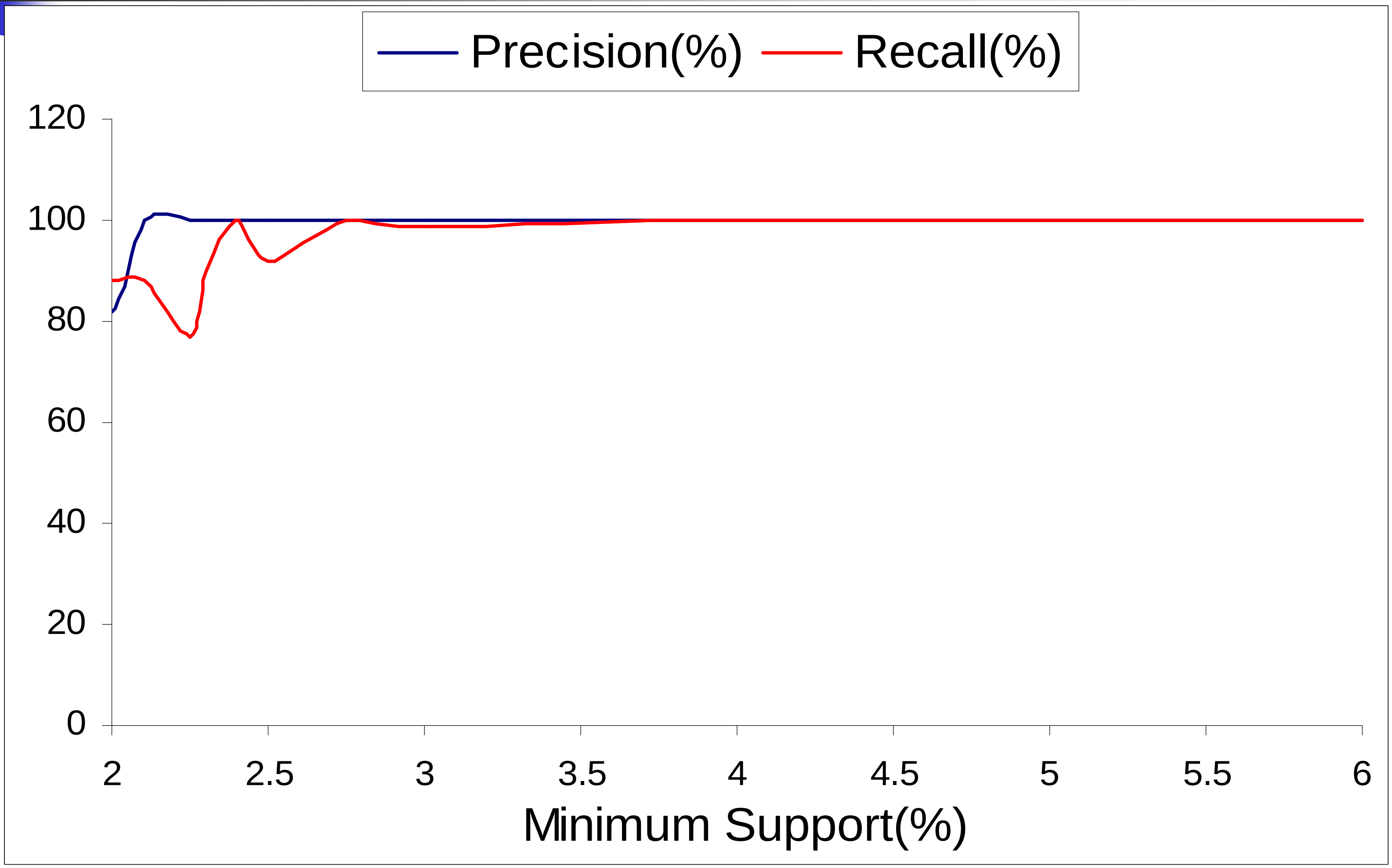
Application to Data Mining (contd.)

- Transaction sets generated by IBM Quest data generator
  - Tested on 10K to 1M transactions containing 20(L), 100(M), and 500(H) patterns
- A-priori algorithm ran on
  - Original transaction set
  - Compressed transaction set
- Results
  - Speed-up in the order of hundreds
  - Almost 100% precision and recall rates

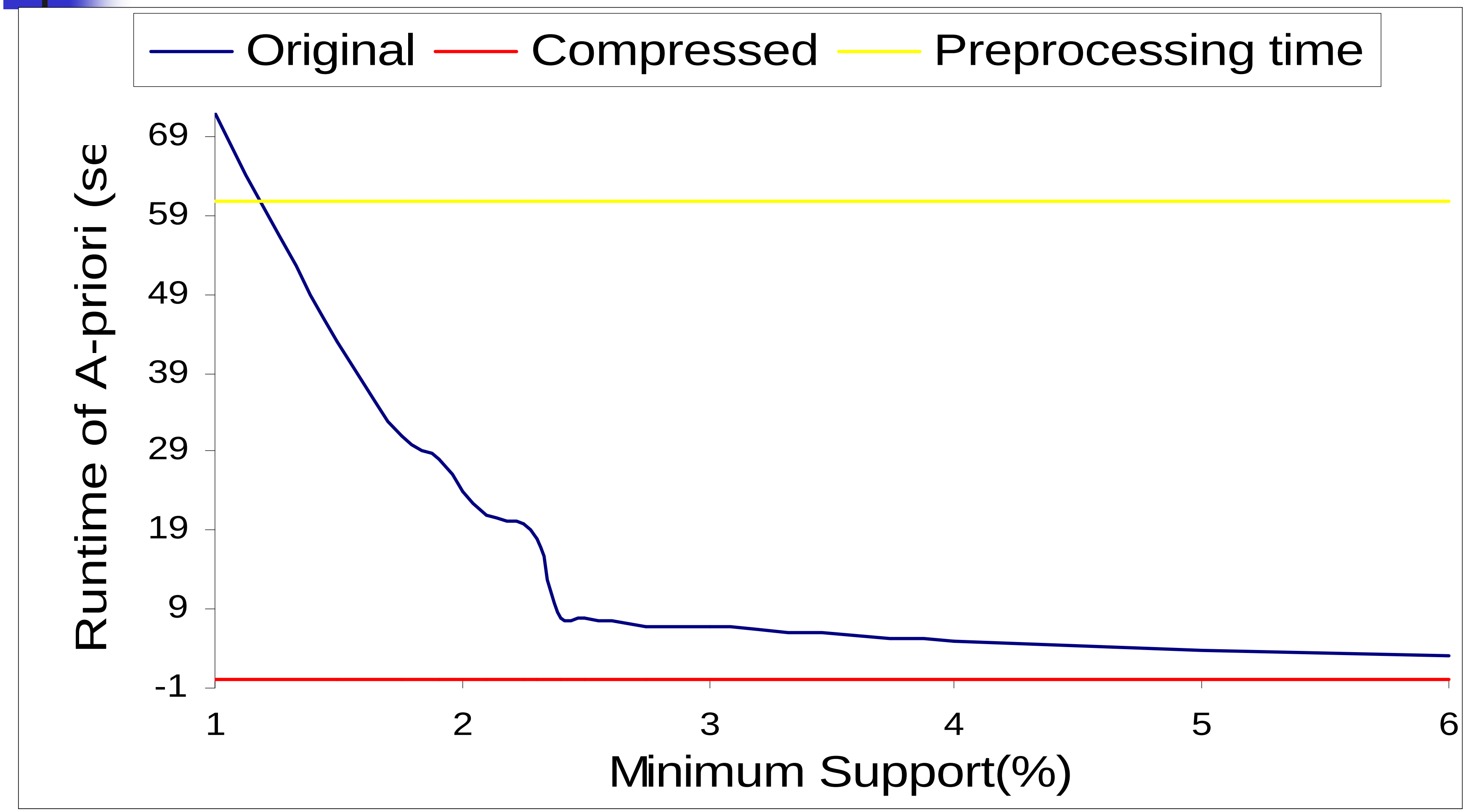
Preprocessing Results

Data	# trans.	# items	# pats.	# sing. vectors	Prepro. time (secs.)
M10K	7513	472	100	512	0.41
L100K	76025	178	20	178	3.32
M100K	75070	852	100	744	4.29
H100K	74696	3185	500	1445	12.04
M1M	751357	922	100	1125	60.93

Precision & Recall on M100K



Speed-up on M100K



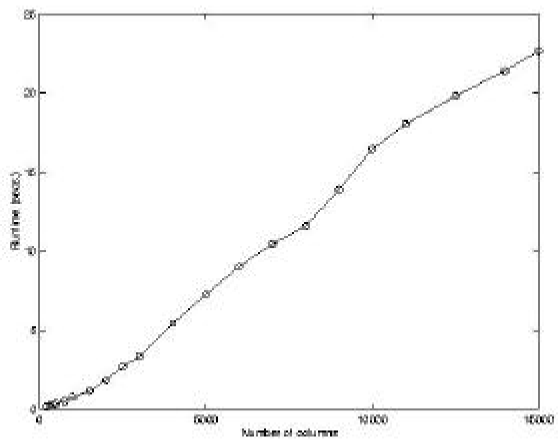
Run-time Scalability

-Rank-1 approximation requires *O(nz(***A***))* time

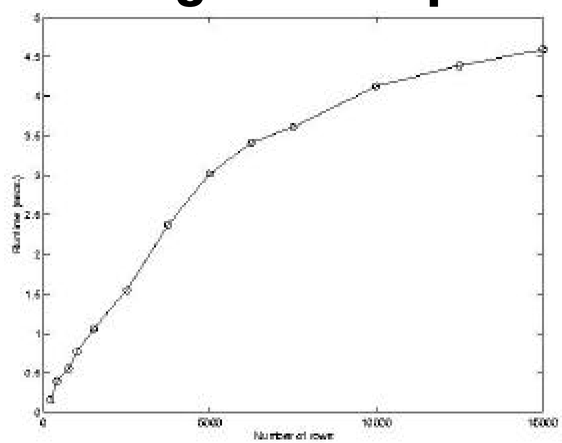
-Total run-time at each level in the recursive tree can't exceed this since total # of non-zeros at each level is at most *nz(***A***)*

⇒ Run-time is O(*k*X*nz(***A***))* where *k* is the number of discovered patterns

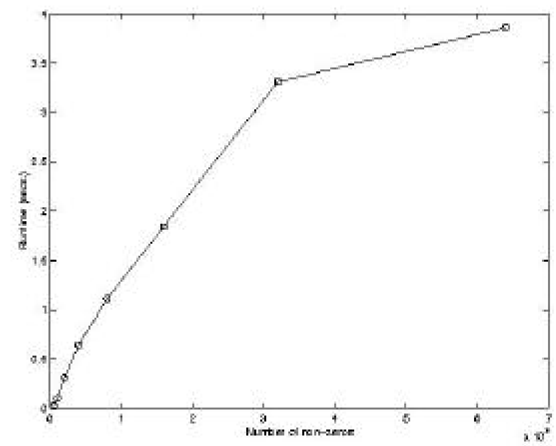
Run-time on data with 2 gaussian patterns on each row



runtime *vs* # columns



runtime *vs* # rows



runtime *vs* # nonzeros

Conclusions and Ongoing Work

- Scalable to extremely high-dimensions
  - Takes advantage of sparsity
- Clustering based on dominant patterns rather than pairwise distances
- Effective in discovering dominant patterns
- Hierarchical in nature, allowing multi-resolution analysis
- Current work
  - Parallel implementation

References

- [Berry et.al., 1995] M. W. Berry, S. T. Dumais, and G. W. O'Brien, Using linear algebra for intelligent information retrieval, *SIAM Review,* 37(4):573-595, 1995.
- [Boley, 1998] D. Boley, Principal direction divisive partitioning (PDDP),*Data Mining and Knowledge Discovery,* 2(4):325-344, 1998.
- [Chu & Funderlic, 2002] M. T. Chu and R.E. Funderlic, The centroid decomposition: relationships between discrete variational decompositions and SVDs, *SIAM J. Matrix Anal. Appl.,* 23(4):1025-1044, 2002.
- [Kolda & O'Leary, 1999] T. G. Kolda and D. O'Leary, Latent semantic indexing via a semi-discrete matrix decomposition, In *The Mathematics of Information Coding, Extraction and Distribution,* G. Cybenko et al., eds., vol. 107 of IMA Volumes in Mathematics and Its Applications. Springer-Verlag, pp. 73-80, 1999.
- [Kolda & O'Leary, 2000] T. G. Kolda and D. O'Leary, Computation and uses of the semidiscrete matrix decomposition, *ACM Trans. On Math. Software,* 26(3):416-437, 2000.