Learning with a collection of matrices, and tensors

Clément Grimal (LIG PhD student)

7^{th} January, 2010

Supervisors: Éric Gaussier and Gilles Bisson

1. The clustering problem
   - Dyadic co-clustering
   - Multi-way co-clustering
     - Multi-type co-clustering
     - High-order co-clustering
2. Existing solutions
   - Solutions for multi-type co-clustering
   - Solutions for high-order co-clustering
3. Proposed solutions for multi-type co-clustering
   - Chain
   - Alternate
   - Merge
   - Few results

Dyadic co-clustering

Classical problem

2 types of objects linked by a simple relationship:

$$\boxed{\text{documents}} \xrightarrow{\text{contain}} \boxed{\text{words}}$$

Classical representation

A co-occurrences matrix:

$$\begin{array}{cc} & \text{words} \\ \text{documents} & \begin{bmatrix} 1 & 0 & 2 & 1 & \cdots \\ 0 & 2 & 3 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array}$$

Multiple interrelated types of objects

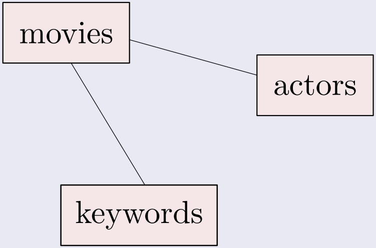
How can we co-cluster multiple types of data objects?

- How are they linked to each others?
- How can we build a model to represent their relationships?

Multi-type co-clustering

Example

3 types of objects linked by pairwise relationships:



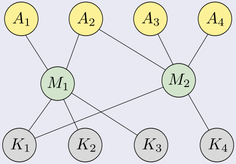
Multi-type co-clustering

Representations

A set of co-occurrences matrices (one for each relationship)

$$\begin{array}{cc} & \text{actors} \\ \text{movies} & \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots \\ 0 & 1 & 1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array} \qquad \begin{array}{cc} & \text{keywords} \\ \text{movies} & \begin{bmatrix} 1 & 1 & 1 & 0 & \cdots \\ 1 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{array}$$

or a *n*-partite graph



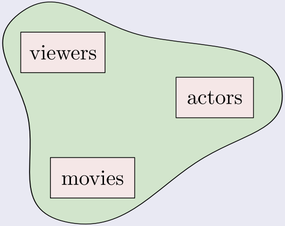
High-order co-clustering

Going further...

What if the 3 types of objects are linked by a triadic relationship?

Example

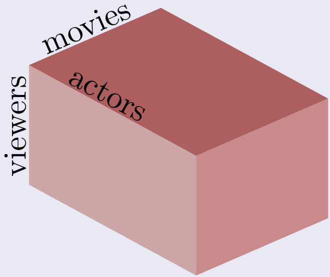
Viewers are giving feedback on the performance of **actors** in different **movies**.



High-order co-clustering

Representations

A *n*-way tensor



or an hyper-graph.

Solutions for multi-type co-clustering

List of solutions

- Multi-Latent Semantic Analysis (MLSA)
- Relational Summary Network (RSN)
- Multi-way Distributional Clustering (MDC)
- Linked Matrix Factorization (LMF)
- Spectral clustering on Multi-type relational data
- Multi-way Relation Graph Clustering (MRGC)

Linked Matrix Factorization

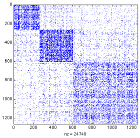
From...

W. Tang, Z. Lu, and I. S. Dhillon, Clustering with Multiple Graphs, to appear in Proceedings of the IEEE International Conference on Data Mining (ICDM), December 2009.

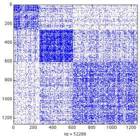
Problem

How to combine information coming from different sources?

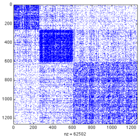
abstract



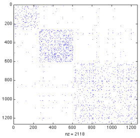
title



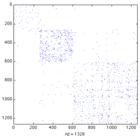
keywords



author



citation



Linked Matrix Factorization

Model

$A \approx P \Lambda P^T$ where P is an N x d matrix and Λ is a d x d symmetric matrix.

$$\mathcal{G} = \frac{1}{2} \sum_{m=1}^{M} \| A^{(m)} - P \Lambda^{(m)} P^T \|_F^2 + \dots$$

Finally find the clusters with P.

Solutions for high-order co-clustering

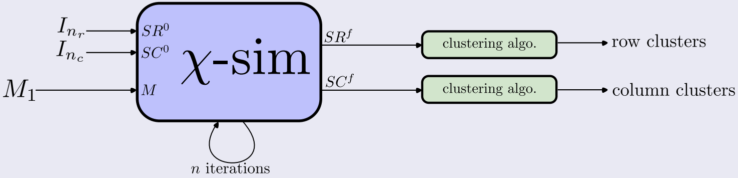
List of solutions

- Tensorial Probabilistic Latent Semantic Analysis (T-PLSA)
- Non-negative tensor factorization
- Hyper-graph partitioning
- Multi-way clustering using Bregman divergence

χ-Sim algorithm

Principle

From a co-occurrences matrix, computes the rows and the columns similarities. Then, perform a clustering algorithm on both similarity matrices.

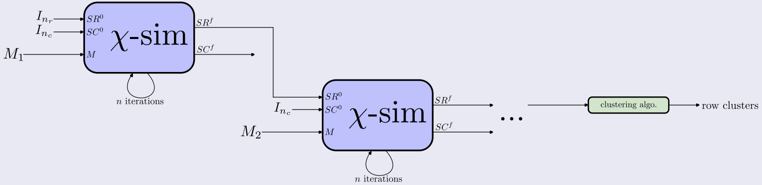


How can we use for multi-type (or high-order) co-clustering?

Chain

Principle

Compute the similarity matrix from a first data matrix, and use it to initialize the algorithm with a second matrix.



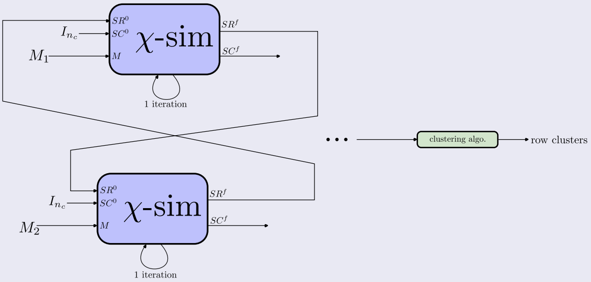
Problems

- How do we choose the order of the matrices?
- How many matrices do we use?
- How many iterations do we perform for each matrix?

Alternate

Principle

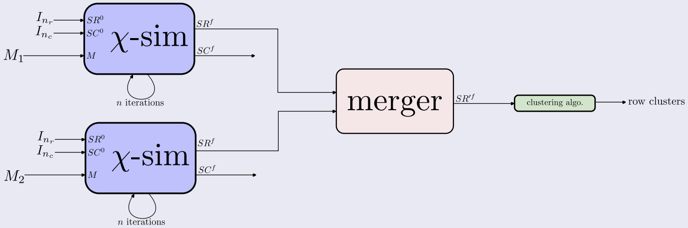
Similar to the previous one, but performing only one iteration on a matrix.



Merge

Principle

Compute the similarity matrices from several data matrices, and merge them before performing the clustering algorithm on it.



How do we merge? (average, min, max...)

Few results

Using only one matrix

	M_1	M_2
1 iteration	31,6 %	26,8 %
2 iterations	31,2 %	40,9 %

Chain

	$M_1 \rightarrow M_2$
1 iteration - 1 iteration	37,6 %
1 iteration - 2 iterations	40,5 %

Merge

	$min\_merge(M_1, M_2)$	$max\_merge(M_1, M_2)$
1 iteration - 1 iteration	50,2 %	31,2 %