Improve visual perception and human understanding of Big Data using graph/hypergraph based visualisation

PhD student: Xavier Ouvrard
xavier.ouvrard@etu.unige.ch
PhD supervisor: Pr. S. Marchand-Maillet
CERN supervisor: Dr. J.-M. Le Goff

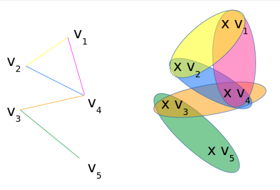
CERN Project

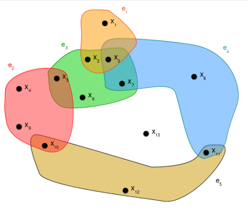
- **PhD:**
  - started in 10.2016 @ UniGe
  - done within the **Collaboration Spotting** project @ CERN
- **Aim:**
  Developping a generic web application to visually query datasets by enabling raw or processed links in datasets

  cf: ERCIM News 111 Le Goff et al. [2017], Agocs et al. [2017]
- **Different** applications:
  - With JRC, EC : TIM: http://www.timanalytics.eu/
  - Used in ARIADNE, LHCb
  - Collaboration with Wigner Institute (Hungary)
  - Other applications queuing

Problematic of the PhD

- In project, before PhD:
  - **Datasets**
    - mostly textual data
    - stored as **labelled graphs**.
  - **Collaborations** are built choosing a **reference**.
  - Multiple **facets** of dataset are visualized.
  - Represented by **pairwise relationships** => graphs
- But ...
  - **Collaborations**:
    - sets of elements
    - n**-adic relationships**
  - **Hypergraphs** fits for n-adic relationships





RQs induced

- One global RQ:
  How to visually render collaborations so it allows smooth interaction with the data for knowledge discovery?

- Different facets of the global RQ:
  - **Modelisation of dataset with collaborations**:
    - **Are hypergraphs pertinent to achieve interactive navigation and visualisation of facets in an information space?**
  - **Visualisation of hypergraphs and KD**:
    => implies answering theoretical RQ on hypergraphs:
    - **How to extend the concept of adjacency in a hypergraph?**
    - **How to coarsen a hypergraph?**

Table of contents

Collaborations, KD and visualisation

- Global research question:
  How to visually render collaborations so it allows smooth interaction with the data for knowledge discovery ?

- Different **tasks** attached to the visualisation and knowledge discovery
  - Showing the **global** and **local** structure of the data
  - Showing the **links** in between data
  - Showing **temporal** evolution
- Requirements
  - Visualisation should be **quickly computed** (web browsing)
  - Give **optimal information** depending on the task to be solved
- Evaluation
  User evaluation through task solving with baseline

Information space, facets and hypergraphs I

RQ1: **Are hypergraphs pertinent to achieve interactive navigation and visualisation of facets in an information space?**

Hypergraph framework designed in Ouvrard et al. [2017a].

1. Building the (extended) schema hypergraph
=> metadata level
2. Building the reachability hypergraph
=> metadata level
3. Building the navigation hypergraph
=> metadata level
4. Building the visualisation hypergraph
=> data level

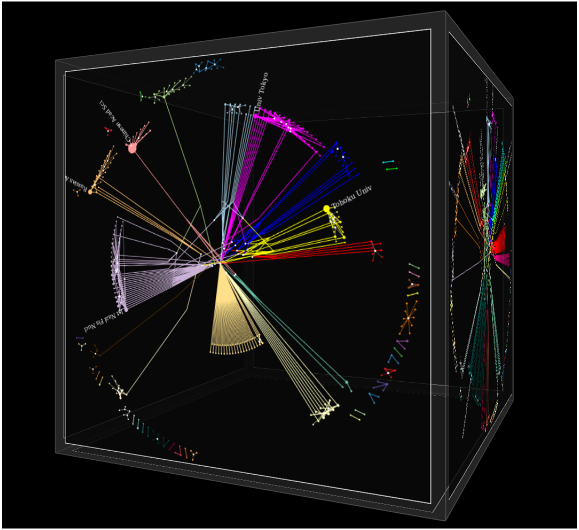
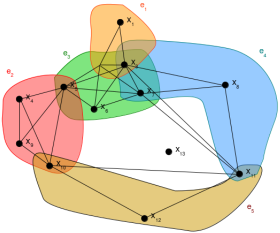


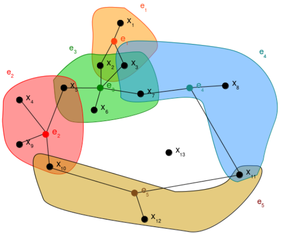
Figure 1: DataHyperCube

Table of contents

Visualizing hypergraphs I

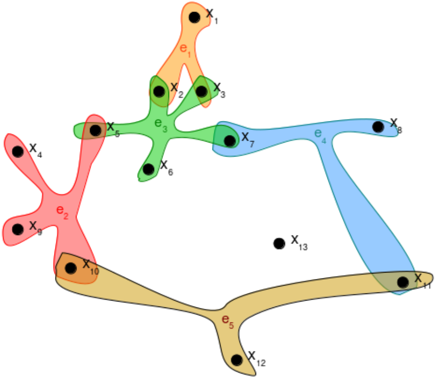


Venn diagram representation and 2-section of the hypergraph



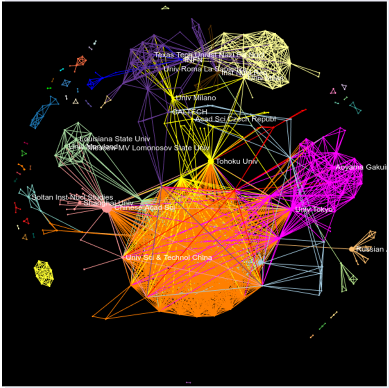
Venn diagram representation and extra-node representation of hypergraph

Visualizing hypergraphs II

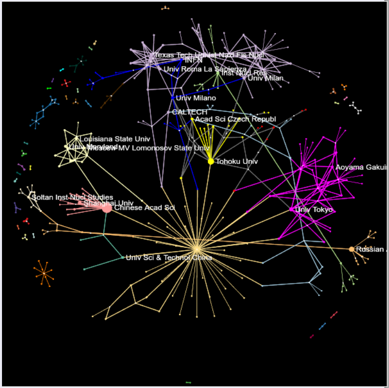


“PaintSplash” representation of hypergraph

Visualizing hypergraphs III



Clique view



Extra-node view

Figures from Ouvrard et al. [2017b]

Visualization of large hypergraphs

Research question:
How to visualize large graphs with maximal knowledge discovery, nice layouts in a time acceptable for the user?

Scaling up: raises many challenges

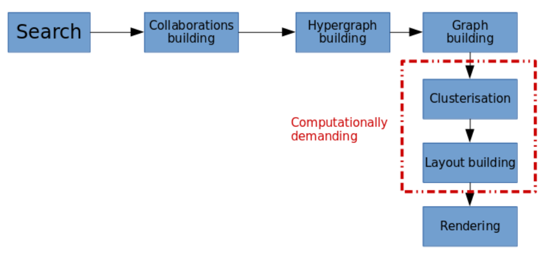
- hypergraphs should have nice aesthetics
- they should give meaningfull information
- allow to interact with data
- compute fast in a reasonnable time (0.5-10 s).

State of the Art: vast domain, one good reference: Tamassia [2013]

Results: article under writting

Evaluation: using metrics for visualization + user evaluation

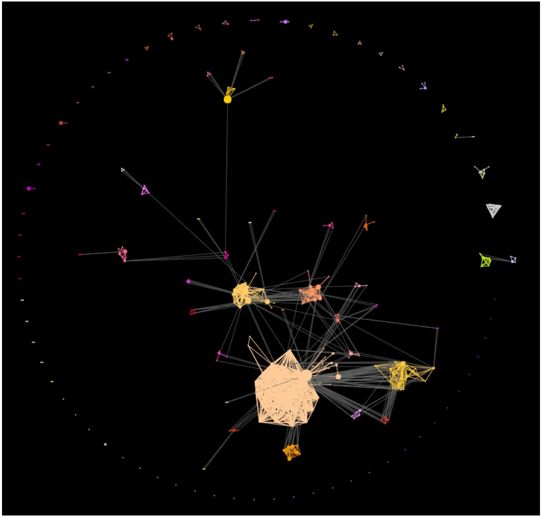
Challenge of computations



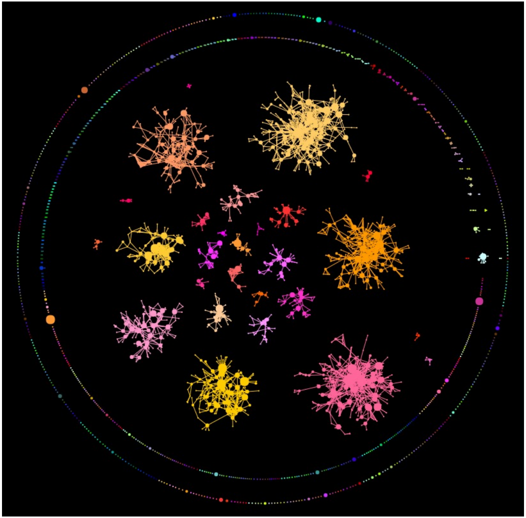
Layout building process I

- Force directed algorithms don't scale up well
- Clusters based on modularity have problem in scaling up
- **Divide and conquer** approach:
  - Working on different quotient hypergraphs at different levels
  - Work separately on each component
  - Gathering

Layout building process II



Layout building process III



Layout building process IV

Advantages:

- better identification of communities
- force-directed algorithm is more efficient

Remaining problems:

- Overlapping issues
  => necessity to spread information
- Nodes are not always well distributed
- Main nodes are not always clearly identifiable
- Main collaborations are not highlighted sufficiently

=> finding way of coarsening the hypergraph

Table of contents

Coarsening of hypergraphs I

- Research question:
  How to coarsen a hypergraph?

- Task to be solved:
  - spot out the important structures of a hypergraph
- Important for
  - spraying the information shown
  - give focus on important information
- Different way of spotting out the important nodes
  - k-core approach
  - diffusion approach
  - exchange approach
  - spectral approach
- Evaluation
  - on random hypergraphs with chosen parameters,
  - on open dataset

Table of contents

Adjacency in hypergraphs I

Pu [2013] makes clear distinction between:

- **pairwise relationship**: relation between two entities
- co-occurence extends pairwise relations to a p-**adic relationship**

But in this article:

- co-occurence is represented by a 2D-matrix: pairwise view
- co-occurence is used for text.

Adjacency in hypergraphs II

In graph:

- Two nodes are said **adjacent** if it exists an edge linking them
  => **pairwise relationship**
- Nodes incident to one given edge are said **e-adjacent.**
  => also **pairwise relationship**
- E-adjacency and adjacency are **equivalent in graphs**

Adjacency in hypergraphs III

Extending to hypergraph:

- Two nodes are said **adjacent** if it exists an hyperedge linking them
  => pairwise relationship
- Taking all nodes from one hyperedge: they are all incident to this hyperedge and these nodes are said to be **e-adjacent.**
  => **n-adic relationship**
- There is **no more equivalence** in between the two notions in the general case

Adjacency in hypergraphs IV

- Research question:
  How to extend the concept of adjacency in a hypergraph?

- Motivated by the diffusion approach
- Adjacency = pairwise concept cf Pu [2013]
- In hypergraphs adjacency is more complex

- In Ouvrard and Marchand-Maillet [2017]:
  - definition of k-**adjacency**
  - definition of **e-adjacency**
  - building of an **e-adjacency tensor**

Table of contents

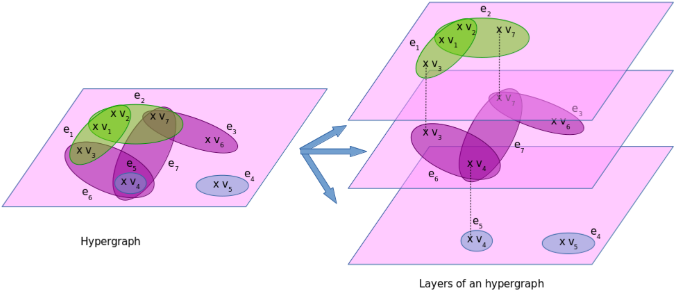
Expectations.

Slides from this part are built from Ouvrard and Marchand-Maillet [2017]

The tensor should be:

- invariant to vertex permutations either globally or at least locally.
- allow the retrieval of the hypergraph in its original form.
- the sparsest possible in between two possible choices.
- allow the retrieval of the degrees of the nodes

Decomposition of hypergraphs I



Decomposition of the hypergraph in layers

Decomposition of hypergraphs II

- Defining a direct summation of hypergraphs:
  Sum of two hypergraphs $\mathcal{H}_1 = (V_1, E_1)$ and $\mathcal{H}_2 = (V_2, E_2)$:

  $$\mathcal{H}_1 + \mathcal{H}_2 = (V_1 \cup V_2, E_1 \cup E_2) \, .$$

  Direct sum if $E_1 \cap E_2 = \emptyset$. In this case the sum is written $\mathcal{H}_1 \oplus \mathcal{H}_2$.

Decomposition of hypergraphs III

- Layers are k-uniform hypergraphs

$$\mathcal{H} = \bigoplus_{k=1}^{k_{\max}} \mathcal{H}_k.$$

- Each layer can be modeled by a hypercubic and hypersymmetric tensor of order k and dimension n
=> **family of e-adjacency tensors** $\mathcal{A}_{\mathcal{H}} = (\mathcal{A}_k)$

Tensor, hypermatrix and homogeneous polynomials I

Technical part:

- allow to associate a tensor to a hypermatrix by choosing a basis
- a hypermatrix of order k and dimension n which is symmetrical and cubical to an homogeneous polynomial of degree k

Tensor, hypermatrix and homogeneous polynomials II

Applied to hypergraphs:

- Hypergraph

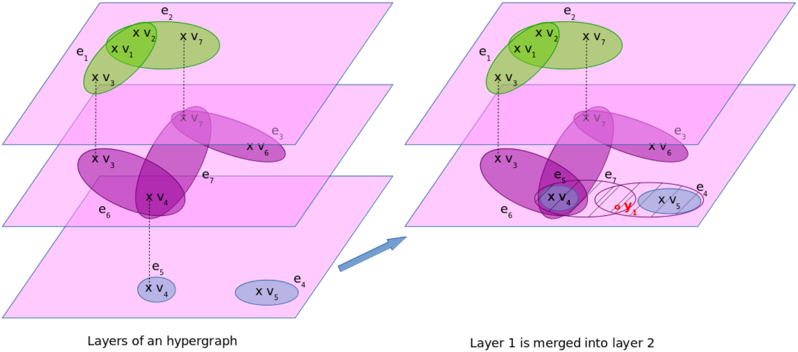
=> layers of k-uniform hypergraphs
=> family of unnormalized tensor $\mathcal{A}_{\mathcal{H}} = (\mathcal{A}_k)$ and hypermatrix $(\boldsymbol{A}_k)$ symmetric and cubical
=> family of homogeneous polynomials P_k
=> homogeneous polynomials can be reduced and ordered

Gathering layers I

- Key idea behind:

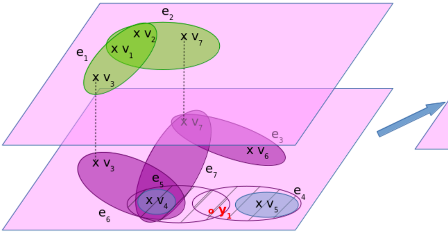
$$(k) + 1 = (k + 1)$$

- Layer k: k-uniform hypergraph
- Adding one element to each hyperedge allows to have $k+1$-uniform hypergraph
- Merging of layer k and layer $k+1$ is now possible
- Iterative process

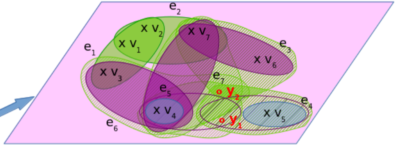


Additional element in the lower level and merging (step 1)

Gathering layers III



Merged layer 1 and 2



Merged layer 1 and 2 into layer 3

Additional element in the lower level and merging (step 2)

Gathering layers IV

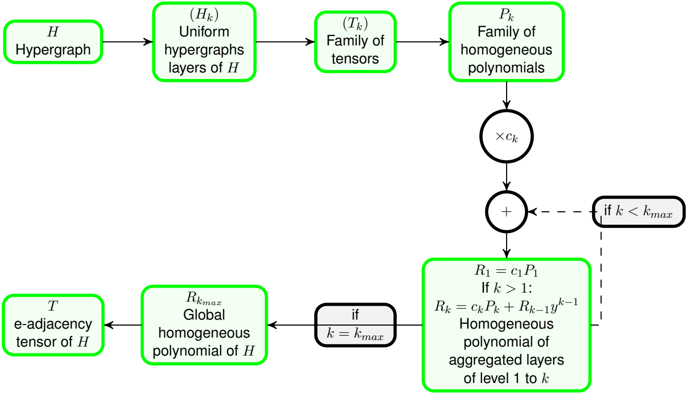


Table of contents

Coarsening the hypergraph

- Spectral techniques for hypermatrix exist: cf Qi and Luo [2017]
- Apply them to the e-adjacency tensor => can help in finding good approximation of the original hypergraph
- Tücker decomposition of tensors => higher order SVD could help reduce the hypergraph
- Diffusion/Exchange in hypergraphs

Bibliography I

A. Agocs, D. Dardanis, J.-M. Le Goff, and D. Proios. Interactive graph query language for multidimensional data in collaboration spotting visual analytics framework. *ArXiv e-prints*, December 2017.

Claude Berge and Edward Minieka. *Graphs and hypergraphs*, volume 7. North-Holland publishing company Amsterdam, 1973.

Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 2013. doi: 10.1007/978-3-319-00080-0. URL `http://dx.doi.org/10.1007/978-3-319-00080-0`.

Jean-Marie Le Goff, Adam Agocs, Dimitris Dardanis, Richard Forster, Xavier Ouvrard, and André Rattinger. Collaboration spotting: A visual analytics platform to assist knowledge discovery. *ERCIM News*, (111), October 2017.

Xavier Ouvrard and Stéphane Marchand-Maillet. Adjacency matrix and co-occurrence tensor of general hypergraphs: Two well separated notions. *arXiv preprint to be published*, 2017.

Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. An hypergraph based framework for modelisation and visualisation of high dimension multi-facetted data. *to be published*, 2017a.

Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Networks of collaborations: Hypergraph modeling and visualisation. *arXiv preprint arXiv:1707.00115*, 2017b.

Li Pu. Relational learning with hypergraphs. 2013.

L. Qi and Z. Luo. *Tensor Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974751. URL http://epubs.siam.org/doi/abs/10.1137/1.9781611974751.

Roberto Tamassia, editor. *Handbook on Graph Drawing and Visualization*. Chapman and Hall/CRC, 2013. ISBN 978-1-5848-8412-5. URL https://www.crcpress.com/Handbook-of-Graph-Drawing-and-Visualization/Tamassia/9781584884125.

Hypergraphs

- Hypergraphs introduced by Berge and Minieka [1973].
- Definition of Bretto [2013]:

A **hypergraph** $\mathcal{H}$ on a finite set $V = \{v_1\,;\, v_2;\, ...\,;\, v_n\}$ is a family of hyperedges $E = \{e_1, e_2, ..., e_p\}$ where each **hyperedge** is a non-empty subset of V

Order of $\mathcal{H}$: $|V| = n$
Size of $\mathcal{H}$: $|E| = p$

- Two **visions** of hypergraphs
  - **set of elements** of power set of nodes => set view
  - **extension of graphs** => n-adic relationship view

Adjacency in hypergraphs I

- k nodes are said k**-adjacent** if it exists a hyperedge that hold them.
  - $k = 2$: usual notion of adjacency is retrieved
  - If k vertices are k-adjacent then each subset of this k vertices of size $l \leqslant k$ is l-adjacent.
- The nodes belonging to one given hyperedge are said **e-adjacent.**
- In k-uniform hypergraph:
  k-adjacency is equivalent to e-adjacency.
- **In general hypergraphs**: nodes that are k-adjacent with $k < k_{\text{max}} = \max_{e \in E} |e|$ have to co-occur potentially with other nodes => k**-adjacency is distinct from e-adjacency**.

Example I

Reminder: $e_1 = \{v_1, v_2, v_3\}$, $e_2 = \{v_1, v_2, v_7\}$, $e_3 = \{v_6, v_7\}$, $e_4 = \{v_5\}$, $e_5 = \{v_4\}$, $e_6 = \{v_3, v_4\}$ and $e_7 = \{v_4, v_7\}$.

- Using:
  - the degree-normalized adjacency tensor $\mathcal{A}_k = \frac{1}{(k-1)!}\mathcal{A}_{k\,raw}$
  - and the normalizing coefficients $c_k = \frac{1}{k}$
- It follows:
  - $R_1(z) = z_4 + z_5$
  - $R_2(z) = z_4 y_1 + z_5 y_1 + \frac{2!}{2}(z_3 z_4 + z_6 z_7 + z_4 z_7)$
  - $R_3(z) =$ $z_4 y_1 y_2 + z_5 y_1 y_2 + z_3 z_4 y_2 + z_6 z_7 y_2 + z_4 z_7 y_2 + z_1 z_2 z_3 + z_1 z_2 z_7$

Example II

Reminder: $e_1 = \{v_1, v_2, v_3\}$, $e_2 = \{v_1, v_2, v_7\}$, $e_3 = \{v_6, v_7\}$, $e_4 = \{v_5\}$, $e_5 = \{v_4\}$, $e_6 = \{v_3, v_4\}$ and $e_7 = \{v_4, v_7\}$.

- e-adjacency tensor of $\mathcal{H}$ is a symmetric cubical tensor of order 3 and dimension 9, described by:

$$r_{489} = r_{589} = r_{349} = r_{679} = r_{479} = r_{123} = r_{127} = \frac{1}{3!}.$$

The other remaining not null elements are obtained by permutation on the indices.

- Degree of one vertex: $\deg(v_4) = 3!\,(r_{489} + r_{349} + r_{479}) = 3$.