Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 70 (2018) 65–70

www.elsevier.com/locate/endm

Hypergraph Modeling and Visualisation of Complex Co-occurence Networks

X. Ouvrard [a,b,1], J.M. Le Goff [a] and S. Marchand-Maillet [b]

[a] *CERN, CH-1211 Geneva 23*

[b] *University of Geneva, CUI, 7 route de Drize, Battelle A, CH-1227 Carouge*

Abstract

Finding inherent or processed links within a dataset allows to discover potential knowledge. The main contribution of this article is to define a global framework that enables optimal knowledge discovery by visually rendering co-occurences (i.e. groups of linked data instances attached to a metadata reference) - either inherently present or processed - from a dataset as facets. Hypergraphs are well suited for modeling co-occurences since they support multi-adicity whereas graphs only support pairwise relationships. This article introduces an efficient navigation between different facets of an information space based on hypergraph modelisation and visualisation.

Keywords: hypergraph modeling, data visualisation, data mining

1 Introduction

Having insight into non-numerical data calls for the gathering of instances: classically (multi-entry) frequency arrays of occurences are used. To get further insight into data instances of a given type, one can regroup them using

[1] This document is part of X. Ouvrard article work supervised by Pr. S. Marchand-Maillet and J.M. Le Goff and founded by a doctoral position at CERN.

[2] Email: xavier.ouvrard@cern.ch

https://doi.org/10.1016/j.endm.2018.11.011

their links to instances of another type - used as reference. It generates a family of co-occurence sets that can be viewed as a facet of the information space. Navigating accross the different facets is achieved by iterating this process between different types of interest while keeping the same reference type: any of these types can be used as a reference. We use a publication dataset as breadcrumb trail example.

Previous approaches using a reference to articulate the different facets of an information space exist [1,2,3]. [4] proposes a graph-based framework which provides insights into the different facets of an information space based on user-selected perspectives, combining type of reference and of co-occurences. [5] shows how the keeping of m-adic relationships can help in gaining understanding in the network evolution.

This article provides a hypergraph-based framework that supports interactions between the different facets of an information space for optimal knowledge discovery. The dataset - mostly textual - refers to physical entities with unique individual references. Data instances are attached to metadata instances. We suppose that there is no metadata instance that doesn't have a data instance attached to it.

2 Modeling co-occurences in datasets

Hypergraphs suits well the storage of co-occurence information with references. A **hypergraph** $\mathcal{H} = (V, E)$ is a hyperedge family $E = \{e_i : e_i \subseteq V \wedge i \in [\![p]\!]\}$ [3] over the vertex set $V = \{v_i : i \in [\![n]\!]\}$ [6]. A hypergraph where the hyperedges are distinct one-to-one is said **with no repeated hyperedge**. In [7], a hypergraph is a triple $\mathcal{H} = (V, E, i)$ with V a vertex set, E a hyperedge set and $i : E \to \mathcal{P}(V)$ [4] an incidence function. Considering a map $w : E \to \mathbb{R}^{++}$ the hypergraph $\mathcal{H}_w = (V, E, w)$ - or $\mathcal{H}_w = (V, E, i, w)$ - is said **weighted**.

2.1 Allowing navigation

Relational database schema are hypergraphs of metadata instances where the hyperedges gather table metadata: normalized forms are linked to the properties of the hypergraphs modeling them [8]. In graph databases, the schema [5] represents the relationships between the vertex types. The **schema hypergraph** $\mathcal{H}_{\text{Sch}} = (V_{\text{Sch}}, E_{\text{Sch}}, i_{\text{Sch}})$ represents these relationships as hyperedges.

[3] $[\![k; n]\!]$ corresponds to $\{i : i \in \mathbb{N} \wedge k \leqslant i \leqslant n\}$ and $[\![n]\!]$ to $[\![1, n]\!]$.
[4] $\mathcal{P}(V)$ is the power set of V
[5] although not required [9]

Each data instance stored in the dataset is labeled using a labeling function on the vertices of V_{Sch}. Hyperedges of the schema itself can be labeled by another labeling function over another label set.

Types of visual or referencing interest are selected in a subset U of V_{Sch} to generate $\mathcal{H}_X = (V_X, E_X, i_X)$ the **extracted schema hypergraph** where $V_X = U$, $E_X = \{e \cap U : e \in E_{\text{Sch}}\}$ and $i_X = i_{\text{Sch}}|_{E_X}$.

From $\mathcal{H}_X$, we build the **reachability hypergraph** $\mathcal{H}_R = (V_R, E_R, i_R)$ with $V_R = V_X$ as vertex set, the hyperedges of $\mathcal{H}_R$ are the connected components $E_{\text{cc}}\,(\subset E_X)$ of $\mathcal{H}_X$ - regrouped in C_X, the set of connected components of $\mathcal{H}_X$ - and $\forall e_R \in E_R : i_R\,(e_R) = \bigcup_{E_{\text{CC}} \in C_X} \bigcup_{e \in E_{\text{CC}}} i_X(e)$.

Last at the level of metadata, the **navigation hypergraph** is built by choosing a nonempty subset of possible reference vertices R_{ref} in a hyperedge $e_R \in E_R$. Non empty subsets of R_{ref} allow to generate possible hyperedges of the navigation hypergraph $\mathcal{H}_N = (V_N, E_N)$ where $V_N = e_R$, $E_N = \{e_R \backslash R : R \subseteq R_{\text{ref}} \wedge R \neq \emptyset\}$. Navigation is possible without changing reference inside a hyperedge of $\mathcal{H}_N$.

In a publication dataset, typical metadata is: *publication id*, title, abstract, authors, affiliations, addresses, *author keywords*, *subject categories*, *countries*, *organisations*,...[6] A possible navigation hyperedge is: {author keywords, organisations, country, subject category} with publication id as reference.

2.2 Facet visualisation hypergraphs

Each physical entity d in a dataset $\mathcal{D}$ is described by a unique physical reference r and a set of data instances of different types $\alpha \in V_{\text{Sch}}$. The types are obtained from the metadata - for instance in publications: organisation, author keywords, country. We write $A_{\alpha,r} = \{a_1, ..., a_{\alpha_r}\}$ the set of values of type α that are attached to d. $A_{\alpha,r}$ is possibly the emptyset if no value of type α is attached to d. Hence d is fully described by: $(r, \{A_{\alpha,r} : \alpha \in V_{\text{Sch}}\})$.

In the navigation hypergraph, each hyperedge $e_N \in E_N$ describes accessible facets relatively to a reference type. A facet will show co-occurences of a chosen type $\alpha \in e_N$ built relatively to reference instances of type $\rho \in V_N \backslash e_N$ (α/ρ as short). For example, in a publication dataset, with organisation as reference, one can retrieve all subject categories that are common to a given organisation.

Performing a search on the dataset will retrieve a set $\mathcal{S}$ of physical references r. A facet will be represented by the visualisation hypergraph of

[6] Metadata of interest for visualisation or referencing are in italic

co-occurences of type α/ρ. The set of all values of type ρ is defined by $\Sigma_\rho = \bigcup_{r\in\mathcal{S}} A_{\rho,r}$. Each value of type ρ is mapped to a set of physical references in which they appear, using $r_\rho : v \in \Sigma_\rho \mapsto R_v$ where $R_v = \{r : v \in A_{\rho,r}\}$. The set of values of type α relatively to the reference v is $\bigcup_{r\in R_v} A_{\alpha,r} = e_{\alpha,v}$.

Hence the **raw visualisation hypergraph** for the facet of type α/ρ attached to the search $\mathcal{S}$ is $\mathcal{H}_{\alpha/\rho,\mathcal{S}} = \left(\bigcup_{r\in\mathcal{S}} A_{\alpha,r}, (e_{\alpha,v})_{v\in S_\rho}\right)$.

Some hyperedges can possibly point to the same subset of vertices. In this case, we build a reduced visualisation weighted hypergraph from the raw visualisation hypergraph. We define: $g_\alpha : v \mapsto e_{\alpha,v}$ and $\mathcal{R}$ the equivalence relation such that: $\forall v_1 \in \Sigma_\rho, \forall v_2 \in \Sigma_\rho$: $v_1 \mathcal{R} v_2 \Leftrightarrow g_\alpha(v_1) = g_\alpha(v_2)$.

Considering $\overline{v} \in \Sigma_\rho/\mathcal{R}$ [7], we write $\overline{e_{\alpha,\overline{v}}} = g_\alpha(v)$ where $v \in \overline{v}$.

$\overline{E_\alpha} = \{\overline{e_{\alpha,\overline{v}}} : \overline{v} \in \Sigma_\rho/\mathcal{R}\}$ is the support set of the multiset [8] $\{\{e_{\alpha,v} : v \in \Sigma_\rho\}\}$: $\overline{e_{\alpha,\overline{v}}} \in \overline{E_\alpha}$ is of multiplicity $w_\alpha(\overline{e_{\alpha,\overline{v}}}) = |\overline{v}|$ in this multiset.

It yields: $\{\{e_{\alpha,v} : v \in \Sigma_\rho\}\} = \left\{\overline{e_{\alpha,\overline{v}}}^{w_\alpha\left(\overline{e_{\alpha,\overline{v}}}\right)} : \overline{v} \in \mathcal{S}_\rho/\mathcal{R}\right\}$

Let $\tilde{g_\alpha} : \overline{v} \in \Sigma_\rho/\mathcal{R} \mapsto e \in \overline{E_\alpha}$, then $\tilde{g_\alpha}$ is bijective. $\tilde{g_\alpha}^{-1}$ allows to retrieve the class associated to a given hyperedge; hence the associated values of Σ_ρ to this class - which will be important for navigation. The references associated to $e \in \overline{E_\alpha}$ are $\bigcup_{v\in\tilde{g_\alpha}^{-1}(e)} r_\rho(v)$. The **reduced visualisation weighted hypergraph** for the search $\mathcal{S}$ is defined as $\mathcal{H}_{\alpha/\rho,w_\alpha,\mathcal{S}} = \left(\bigcup_{r\in\mathcal{S}} A_{\alpha,r}, \overline{E_\alpha}, w_\alpha\right)$.

2.3 Navigability through facets

Keeping the same search $\mathcal{S}$ and reference ρ, the sets $R_v, v \in \Sigma_\rho$ remain the same between the different facets: considering another type $\alpha' \in e_N$ and using the same reference ρ, another visualisation hypergraph $\mathcal{H}_{\alpha'/\rho}$ is built.

Let α being the current type and $\mathcal{H}_{\alpha/\rho,w_\alpha}$ being the current visualisation hypergraph. Focusing on a subset of vertices $A \subseteq A_{\alpha,S}$, we retrieve the corresponding hyperedge subset $\overline{E_\alpha}\big|_A = \left\{e : e \in \overline{E_\alpha} \wedge (\exists x \in e : x \in A)\right\}$ of $\overline{E_\alpha}$ which contains at least one element of A. Using $\tilde{g_\alpha}^{-1}$ we get for each $e \in \overline{E_\alpha}\big|_A$ the class associated to the hyperedge $\overline{v}$, building the set $\overline{V}\big|_A = \left\{\tilde{g_\alpha}^{-1}(e) : e \in \overline{E_\alpha}\big|_A\right\}$. The references of type ρ used to build the co-occurences are: $\mathcal{V}_{\rho,A} = \left\{v : \forall \overline{v} \in \overline{V}\big|_A : v \in \overline{v}\right\}$. From each element v of $\mathcal{V}_A$, the set of

[7] $\Sigma_\rho/\mathcal{R}$ is the quotient set of Σ_ρ by $\mathcal{R}$

[8] In a multiset repetitions of elements are allowed. For further details [10].

physical references R_v is retrieved, considering $r_\rho|_{\mathcal{V}_{\rho,A}}$ as the restriction of r_ρ to $\mathcal{V}_{\rho,A}$. It yields to the physical reference set: $\mathcal{S}_A = \bigcup_{v \in \nu_{\rho,A}} R_v$.

From these physical references, one can switch to another facet of the same search with the same reference type ρ. Let α' be the targetted type. Then only $\mathcal{H}_{\alpha'/\rho}\big|_A = \left(\bigcup_{r \in \mathcal{S}_A} A_{\alpha',r}, (e_{\alpha',v})_{v \in \mathcal{V}_{\rho,A}} \right)$ will be processed as raw visualisation hypergraph, using $\mathcal{S}_A$ as reference search set in the former paragraph. To obtain the related reduced weighted version we use the same approach as above. The set of co-occurences retrieved include all occurences that have co-occured with one of the element selected in the first facet.

Of course if $A = A_{\alpha,S}$ the reduced visualisation hypergraph will contain all the instances of type α' attached to physical entities of the search $\mathcal{S}$.

Ultimately, by building a multi-dimensional network organised around types, one can retrieve very valuable information from combined data sources. This process can be extended to any number of data sources as long as they share one or more types. Otherwise the reachability hypergraph is not connected and only separated navigations will be possible.

3 Conclusion

Using the connected components of the extracted schema we have enabled the possibility of navigating the dataset. An application of the hypergraph modeling framework is the DataHedron shown in Figure 1: it enables easy navigation between facets of the information space. It is a 2.5D representation of the information space where each DataHedron face embeds a visualisation hypergraph. Navigation through facets is articulated via the references that links one facet with another. The link by references is realised on one face of the DataHedron. Combining this framework with search tools allows to have deep insight into a dataset.

References

[1] M. Dörk, N. H. Riche, G. Ramos, S. Dumais, Pivotpaths: Strolling through faceted information spaces, IEEE Transactions on Visualization and Computer Graphics 18 (12) (2012) 2709–2718.

[2] J. Zhao, C. Collins, F. Chevalier, R. Balakrishnan, Interactive exploration of implicit and explicit relations in faceted datasets, IEEE Transactions on

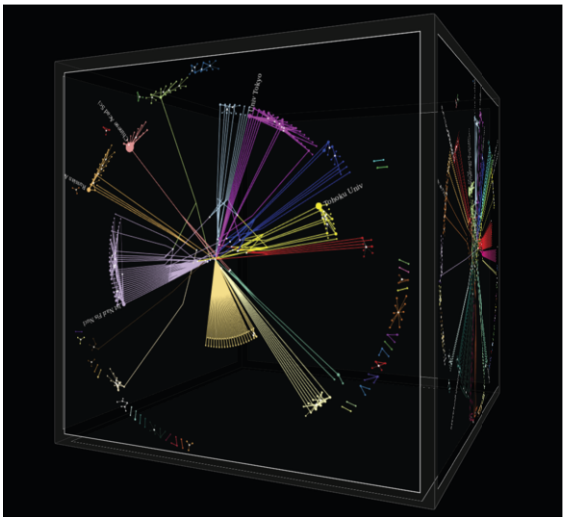

Figure 1. The DataHedron.

Visualization and Computer Graphics 19 (12) (2013) 2080–2089.

[3] S. Hadlak, H. Schumann, H.-J. Schulz, A survey of multi-faceted graph visualization, in: EuroVis, 2015, pp. 1–20.

[4] A. Agocs, D. Dardanis, J.-M. Le Goff, D. Proios, Interactive graph query language for multidimensional data in collaboration spotting visual analytics framework, ArXiv e-printsarXiv:1712.04202.

[5] C. Taramasco, J.-P. Cointet, C. Roth, Academic team formation as evolving hypergraphs, Scientometrics 85 (3) (2010) 721–740.

[6] A. Bretto, Hypergraph theory, An introduction. Mathematical Engineering. Cham: Springer.

[7] J. Stell, Relations on hypergraphs, Relational and Algebraic Methods in Computer Science (2012) 326–341.

[8] R. Fagin, Degrees of acyclicity for hypergraphs and relational database schemes, Journal of the ACM 30 (3) (1983) 514–550.

[9] R. C. McColl, D. Ediger, J. Poovey, D. Campbell, D. A. Bader, A performance evaluation of open source graph databases, PPAA '14, ACM, 2014, pp. 11–18. doi:10.1145/2567634.2567638.

[10] A. Radoaca, Properties of multisets compared to sets, in: Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2015 17th International Symposium on, IEEE, 2015, pp. 187–188.