Algebraic Graph Theory on Hypergraphs

Michael Levet
Introduction

• What is Algebraic Graph Theory? Why do we care?
• Graph structure vs. Graph algorithms
Spectral Graph Theory- Simple Graphs

- Adjacency Matrix
  - Characteristic Polynomial
  - Trace-Eigenvalue Proof
  - Use of eigenvalues to quickly study graph structure
Linear Algebra and Hypergraphs

• No clear definition for adjacency matrix
• Module over Ring R (called R-Module)
  – Two operations: + and *
  – Abelian Group on +
  – *: M x R -> M
  – Multiplication distributes and is associative
• Tensor Algebra
  – Given M and N as modules over commutative rings R and S containing $1_R$ and $1_S$, one can form a third module P such that, given m in M and n in N, mn is in P.
Hypermatrices

• Hypermatrix- Tensor over specific basis
• Matrix $M_{n \times m}$ in $F^{n \times m}$, while vector $m_{nm}$ in $F^{nm}$
• Hyper matrix of form $F^{a \times b \times c \times \ldots \times n}$
• $[A_1 | A_2 | A_3 | \ldots | A_n]$
Adjacency Hypermatrix

• Requires $k$-uniform hypergraph
• Dimension: $|V|^k$
• Analogous to a multi-dimensional array
• $\text{MatrixA}[1][1][3] = 1$: Edge containing exactly $\{v_1, v_3\}$
• $\text{MatrixB}[2][5][7][6] = 0$: No edge containing exactly $\{v_2, v_5, v_6, v_7\}$
• Symmetry- Elements given by permutations of index set have same value.
• Analogue to Square Matrices- Cubical (think $Q_n$)
Hyperdeterminant and Eigenvalues

• Over field $\mathbb{C}$, $\lambda$ is an eigenvalue of hypermatrix $A$ if the following is satisfied: $\text{det}(A - \lambda I) = 0$.

• The det function is defined as a hyperdeterminant

• Analogue of Eigenvectors ($Ax = \lambda x$): $Ax^{k-1} = \lambda x_j^{k-1}$

• Think of $x^{k-1}$ as a basis of cardinality $|k-1|$ (i.e., $x^i$ is an index for a vector coordinate)

• Rather than single equation as in linear algebra, multilinear analogue is system of $(k-1)$ equations
A SIMPLE HYPERGRAPH MIN CUT ALGORITHM
Applicability/Application

- Undirected weighted hypergraph
  - Circuit chip design
  - Skatepark design
- Undirected unweighted hypergraph
  - Clustering
The Cut Function

\[ E \subseteq 2^V, \ A \subseteq V \]

\[ w(A) = \sum \{ w(e) \mid e \in E, \ e \cap A \neq \emptyset, \ e \cap V \setminus A \neq \emptyset \} \]
Basic Graph Example

http://tracer.lcc.uma.es/problems/maxcut/maxcut.htm
Algorithm

```plaintext
MinimumCutPhase(G, w, a)
A ← {a}
while A ≠ V
    add to A the most tightly connected vertex
store the cut-of-the-phase and shrink G by merging the two vertices added last

MinimumCut(G, w, a)
while |V| > 1
    MinimumCutPhase(G, w, a)
    if the cut-of-the-phase is lighter than the current minimum cut
        then store the cut-of-the-phase as the current minimum cut
```
Runtime

$O(|V||E| + |V|^2 \log |V|)$
COMMUNITY DETECTION IN HYPERGRAPHS

Yaser Keneshloo
A hypergraph $H = (V, E)$ is a set of $V$ of nodes and a family $E$ of subsets of $V$ called edges.

If $\exists e \in E: v_i \in e \land v_j \in e$, nodes $v_i$ and $v_j$ are adjacent.

If size of all edges in $E$ equals $r$, $H$ is a $r$-uniform hypergraph.
- If $r = 2$, $H$ is a simple graph

A hypergraph is $k$-partite if $V$ can be partitioned in $k$ sets

If $k = r$, $H$ is a $k$-partite, $k$-uniform hypergraph, also known as $k, k$-hypergraph

COMMUNITY DETECTION IN GRAPH

- The identification of closely connected groups of nodes in complex network
- Reveal the macrostructure and identify functional modules within a network
- Different quality measure can be developed to determine how GOOD a community can be defined.
- Modularity Measure:
  \[
  Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - P_{ij}] \delta(g_i, g_j)
  \]
  - \(m\) is the number of edges
  - \(A\) is the adjacency matrix
  - \(P\) is the expected adjacency matrix, \(P = \frac{k_ik_j}{2m}\), where \(k_i\) is the degree of node \(i\)
  - \(g\) is a clustering
  - \(\delta\) is Kronecker delta and is 1 if \(f(g_i = g_j)\)
CHALLENGES IN HYPERGRAPHS

• Connectivity
  – Rewards adjacent nodes
  – Not suitable for partite graphs

• Community structure
  – Partite graphs may have their own community in each domain

• Hyper-Incidence
  – The binary concept of an edge “connecting two nodes” need to be generalized
• The Mutara’s Modularity
  – A modularity for simple graphs
  – Solve the first two challenges
• 3-Bipartite Modularity
  – Project a triple \((d,u,t)\) to separate edges \((d,u),(d,t),(u,t)\)
  – Uses the Mutara’s Modularity to calculate total modularity
  – Prone to information loss
  – Solve all the challenges!
• Consider each hyper-edge as a node
• Create a weighted line graph from original graph
• Define a similarity measure between hyper-edges
• It applies Infomap algorithm to detect communities
• With communities in line graph, each hyperedge in original graph gets into a single-community which applies automatically assigns overlapping membership to all communities

REFERENCES


Hypergraph Based Clustering in High-Dimensional Data Sets: A Summary of Results

Mohammed Davoodi
What is Clustering?

• Clustering in data mining is a discovery process that groups a set of data.
• Similarities within the cluster is maximized and similarities with other clusters is minimized.
• Clustering has a lot of applications, e.g. business.
What's the problem?

- Clustering techniques take \( n \) data points and \( m \) variables, and find similarities.
- Traditional techniques get inaccurate results with large number of variables.
- This paper proposes a solution to this using Hypergraph's to model the data and a \( k \) partitioning algorithm to divide it into datasets.
How it works

• Apriori algorithm is used to determine related items and insert them into the hypergraph as a hyperedge.

• Edges are assigned weight using essential rules which is the average confidence( conf(x->y) = sup(x U y)/ sup(x) ) of all the items of the edge and has a singleton right hand side.
How it works (cont.)

- Now that the hypergraph has been constructed, the graph must now be split into partitions.
- HMETIS, a partitioning algorithm is used to cut the edges with the minimum weight and create clusters.
- Once the hypergraph has been cut to \( k \) parts, a fitness algorithm is used to eliminate bad clusters.

\[
fitness(C) = \frac{\sum_{e \subseteq C} Weight(e)}{\sum_{|e \cap C| > 0} Weight(e)}
\]
How it works (cont.)

• Now that we have good partitions, each is examined to filter out vertices that are not highly connected to the rest of the vertices of the partition.

$$\text{connectivity}(v, C) = \frac{|\{e|e \subseteq C, v \in e\}|}{|\{e|e \subseteq C\}|}$$
The Results

• Two datasets were examined to see the accuracy of the method.

• Stocks within the S&P500 index were examined. Stocks within the same industry group tend to trade similarly.

• Categorizing protein sequences functionality.
Directions for Future Work

- HMETIS cannot determine the number of partitions to split the hypergraph into.
- The right parameters are necessary to find good clusters.
Mining Social Networks and Their Visual Semantics from Social Photos

Michel Crampes and Michel Plantié
Summary

• Given a collection of photos and a set of people, the authors present a method to build personalized photo albums

• Case Study: Distributing photos taken at a wedding
3 Strategies
3 Strategies

1. Send the photos to people who appear in the respective photos (too constraining)
3 Strategies

1. Send the photos to people who appear in the respective photos (too constraining)
2. Send every photo to every person (too loose)
3 Strategies

1. Send the photos to people who appear in the respective photos (too constraining)
2. Send every photo to every person (too loose)
3. Photo-Tribe Distribution
   1. Identify subsets of people with a common interest (Tribes)
   2. Identify a set of photos of interest to a tribe and send the photos to all members of the tribe
Definitions

[.] is a set (also symbolized as a capital letter): for instance X is the set [x].
Car(.) is a cardinality of a set.
x_i is an individual among all people in a set X (x_j ∈ X)
C is the set of concepts c (c ∈ C)
c is a concept, i.e. a set of photos, (c ∈ C) whose intention is the set of people it contains
[c|[x_i, x_j]] is the set of concepts containing the couple [x_i, x_j].
[c|x_i v x_j] is the set of concepts containing at least x_i or x_j.
Car(c|[x_i, x_j]) is the cardinality of the concept containing [x_i, x_j].
Social Graph

Undirected Weighted Graph \( G = (V, E, w) \)

\( V \) is the set of people

\( E = (u, v) \) indicates social interaction between \( u \) and \( v \)

\( w : E \to \mathbb{R}^+ \cup \{0\} \) is a weight function
Weight Functions

Simple force \( (x_i, x_j) = \frac{\text{Car}(c | [x_i, x_j])}{\text{Car}(C)} \)

Proximity \( (x_i, x_j) = \frac{\sum_{c|x_i, x_j} \left( \frac{2}{\text{Car}(c | [x_i, x_j])} \right)}{\text{Car}(C)} \)

Cohesion \( (x_i, x_j) = \frac{\text{Car}([c | x_i, x_j])}{\text{Car}([c | x_i \lor x_j])} \)
Tribes

- A set of people with a common interest
- A tightly connected subgraph of G
- Found by computing dense subgraphs in G
Tribes

• Convert G into a hypergraph
• Tribes are hyperedges of G
• The same person can belong to multiple tribes
Photo-Tribe (High Level View)

• Create Social Graph from photos
  – Use desired weight function
Photo-Tribe (High Level View)

• Create Social Graph from photos
  – Use desired weight function
• Compute tribes
Photo-Tribe (High Level View)

• Create Social Graph from photos
  – Use desired weight function

• Compute tribes

• For each tribe t
  – Find a set of photos P appealing to t
  – Send P to each person in t
## Performance

<table>
<thead>
<tr>
<th></th>
<th>simple force</th>
<th>cohesion</th>
<th>proximity</th>
</tr>
</thead>
<tbody>
<tr>
<td>recall</td>
<td>57.05%</td>
<td>50.28%</td>
<td>55.49%</td>
</tr>
<tr>
<td>précision</td>
<td>95.19%</td>
<td>97.21%</td>
<td>95.79%</td>
</tr>
<tr>
<td>fmeasure</td>
<td>71.34%</td>
<td>66.28%</td>
<td>70.27%</td>
</tr>
</tbody>
</table>
A new method for mining disjunctive emerging patterns in high-dimensional datasets using hypergraphs

Renato Vimieiro, Pablo Moscato, 2014

Presentation by Tyler Kahn
Disjunctive emerging pattern

- as opposed to a conjunctive emerging pattern
  - shared features that form a pattern
  - depends on a high sample to feature ratio
  - issues dealing with heterogeneity of samples (ex. a single disease caused by dysregulation of molecular pathways $A \parallel B$)
Disjunctive emerging pattern

- Let $A_1, A_2, \ldots, A_n$ be a set of $n$ categorical attributes
- Let $F$ denote the set of features in a dataset (the union of all possible values for the attributes)
- $f(s) = \bigcup A_i(s)$
- Classes $S^+$ and $S^-$
- A disjunctive emerging pattern is a subset of features $X \subseteq F$, fulfilling the following constraints:
  - $X$ includes at least one value from each attribute
  - $X$ occurs in at least $\alpha$ samples with positive class label
  - $X$ occurs in at most $\beta$ samples with negative class label
- A DEP is maximal if there is no proper subset of $X$ that is also a DEP
Borders

- Maximal DEPs are not enough to describe interesting DEPs
- Need also minimal DEPs
- A border is a pair of collections \((L, R)\) where \(L\) and \(R\) are sets of DEPs
- \(R\) can be determined by \(L\) by finding the minimal transversals of a hypergraph
Algorithm 1. An algorithm for mining jumping DEP borders.

Output: $(\mathcal{L}, \mathcal{R})$ the border of the set of jumping DEPs

```
begin
\mathcal{L} \leftarrow \min_{s} \{ f(s) | s \in S^+ \land \forall n \in S^- [f(n) \neq f(s)] \};
\mathcal{R} \leftarrow \emptyset;
foreach L_k \in \mathcal{L} do
\mathcal{E} \leftarrow \{ f(s) - L_k | s \in S^- \land (f(s) - L_k) \neq \emptyset \}
V \leftarrow \bigcup \mathcal{E};
MT \leftarrow \text{minimalTransversals}(V, \mathcal{E});
\mathcal{R} \leftarrow \mathcal{R} \cup \{ F - X | X \in MT \};
end
end
```
Hierarchical, model-based risk management of critical infrastructures

A Paper by F. Baiardi, C. Telmon, D. Sgandurra
Motivation

- Large number of infrastructure components
- Large number of interdependencies among the components
- Simplification desired
Infrastructure hypergraph

- The infrastructure hypergraph is a labeled directed hypergraph
- Components of the system are nodes
- Dependencies in the system are hyperarcs
- The tail label is the attribute of the tail component to be controlled
- The head label is the attribute of the head component that is controlled
Infrastructure hypergraph

- confidentiality, c
  - ability of reading the component state

- integrity, i
  - ability of updating the component state

- availability, a
  - ability of managing the component, of determining who can invoke its operations
Users and attacks

- Users have rights (legal and illegal)
  - set of component attributes \( U \) controls
- Transitive closure to determine full control of User by including dependencies
  - algorithm given
- Attacks have 3 parts
  - target component
  - preconditions (rights)
  - postconditions (new rights)
- complex attack is a sequence of attacks to achieve a goal
Evolution graphs

- model attack evolution with a directed acyclic graph
  - nodes are states
  - there is an arc from two nodes if an elementary attack transitions from one to the other
  - labeled by the user and attack

- can be autonomous or multiple users colluding
Risk

- Risk Analysis
  - Evaluate the probability of each evolution using the complexity and resources required

- Risk mitigation, implement countermeasures
  - remove a vulnerability
  - update dependencies
  - update initial rights of some users
  - increase resources required for attack
Hierarchical model

- Real infrastructure is large, analysis is computation intensive
- allow components of the infrastructure to be hierarchically decomposed
- model will be a high level abstraction as a simple hypergraph with the components expandable as further nodes.
- conditions mentioned which ensure reduction in computation cost
Hierarchical Model

Fig. 7. An example, (a) A high-level description of a (subset) of a system and (b) a hierarchical decomposition of the hand-held device.

Fig. 4. Hierarchical decomposition of a component. (a) An infrastructure hypergraph and (b) hypergraph after the decomposition of C1.
Hierarchical Decomposition

- Hyperarcs allow decomposition of components
- Decomposability Condition 1: No low-level evolution results in new rights for any user.
- Decomposability Condition 2: A low-level evolution may result in new rights for a user only if these rights do not enable any elementary attack of a high-level evolution.
- Decomposability Condition 3: A low-level evolution may result in new rights for a user only if these rights do not increase the number of elementary attacks of high-level evolutions the user can implement.
- These are all sufficient conditions but not necessary conditions.
- Decomposability Condition 4: Further countermeasures are introduced to stop any low-level evolution that violates the previous condition.
- Defining a necessary and sufficient condition is complex. It depends on user, vulnerabilities of and attacks against components not involved in the decomposition.
- Extend to decomposing multiple components to a hypergraph.
Relevance

- General model founded on hypergraph
- Might allow for hypergraph algorithms to determine risk
- Statistical hypergraph prediction of attacks
- Allows for hypergraph research on an explicit or general example of infrastructure
“Evolutionary Dynamic for Inter-Group Cooperation”

AUTHORS: MIHAI SUCIU, NOEMI GASKO

PRESENTER: BRENDAN AVENT
Fast Version – Prisoner’s Dilemma

Rules:
- Each player independently chooses either Cooperate or Defect simultaneously
- Payoffs to players are listed as (P1, P2) in each entry in the table
- $T > R > P > S$, with $T + S < 2R$

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>(R, R)</td>
<td>(S, T)</td>
</tr>
<tr>
<td>Defect</td>
<td>(T, S)</td>
<td>(P, P)</td>
</tr>
</tbody>
</table>

- Generalization: Public Goods game
Exremely Fast Version – Evolutionary Game Theory

Examines how cooperation emerges amongst selfish (self-interested) individuals

Prisoner’s Dilemma is of interest here

Via competition, EGT allows successful strategies to propagate through the players, while failing strategies die off
Hypergraph for inter-group cooperation

Proposed hypergraph model:
- Vertices represent the places
- Hyperedge represents a group of players

Simulations were run for with:
- Rule: $R_i, i \in \{1,2,3,4\}$ (next slide)
- Payoff values: $R = 2, P = 1, S = 0, T = b \geq 2$
- Percentage of defectors: $p_D$

No intra-group competition

Fig. 1. The hypergraph representation in the fourth round, $p_D = 0.01, b = 2.5$. Nodes in hyperedge are identified by different shades.
Rule list and scoring measure

$R_1$: For each player $i$ in a given hyperedge $E_a$, randomly choose another hyperedge $E_b$ and have $i$ compete against a randomly chosen number (determined at the beginning of the round) of opponents in that hyperedge.

$R_2$: For each player $i$, randomly choose a predefined number of opponents from another randomly chosen group.

$R_3$: Randomly choose a hyperedge $E_b$, and then for each player $i$ in $E_a$, $i$ competes against every player in $E_b$.

$R_4$: For each player $i$, randomly choose 1 opponent to compete against (standard situation, no hypergraph necessary).

Frequency of cooperation ($F$) is the scoring measure of cooperation:

$$F = \frac{\text{number of cooperators}}{\text{number of players}}$$
Quick Results

Fig. 2. Evolution of frequency of cooperation in the first hyperedge in two different simulations. Players play according to the rule $R_1$, $p_0 = 0.01$, $b = 2.5$.

Fig. 5. Evolution of frequency of cooperation for a game played between ten groups, each group has ten players. Players play according to the rule $R_1$, $p_0 = 0.01$, $b = 2.5$. Even if the number of defectors is greater than previous cases (one defector in each group) the groups cooperate for more rounds.
Quick Conclusions

Inter-group cooperation is hard.

When the number of defectors is relatively small, cooperation is not affected.

Cooperation is independent of the number of groups and the chosen $b$.

Once a defector strategy becomes adopted by cooperators, the strategy begins to spread exponentially.
Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content

Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, Xiaofei He

Presented by Antuan Byalik
Goal

- Create a music recommendation system that is better than pure user rating
- Incorporate social media information to get better recommendations
Why Hypergraphs?

• The paper lists two main problems when using social media in this context:
  • “There are many different types of objects and relations in music social communities, which makes it difficult to develop a unified framework taking into account all objects and relations.”
  • “In these communities, some relations are much more sophisticated than pairwise relation, and thus cannot be simply modeled by a graph.”
Notation and Formal Definition

- $G(V, E, w)$ – denotes a Hypergraph where
  - $V$ – set of vertices, $E$ – set of hyperedges, $w$ – weight function
- $D(e)$ – degree of a hyperedge = cardinality of that edge’s set
- Unified Hypergraph – Hypergraph with multi-type vertices and hyperedges
  - Vertices/edges represents users/groups/songs
The Approach

- Focus on six objects and nine relations
- Construct a unified Hypergraph
The Approach

- $E_1$ for each pairwise friendship with weight $= 1$
- $E_2$ with vertices corresponding to all users in a group including the group itself for every group with weight $= 1$
- $E_3$ for each user-track combination with weight $= \text{frequency}$
- $E_4/E_5/E_6$ for tracks/albums/artists with three vertices in each for user, tag and resource - weight $= 1$
The Approach

• E7/E8 represents album with all its tracks and similarly artist contains all albums by that artist with all weights = 1

• E9 is the hyperedge set for the knn graph on acoustic-based music similarities with the weight = similarity of the two tracks (internally defined formula for this)

• Lastly, construct the vertex-hyperedge incidence matrix
Methodology

• Offline training – construct the Hypergraph and compute incidence matrix
• Online recommendation – query based on a user and compute rankings
  • Use vertex for user in combination with a defined cost function
  • This gives sorted list of ‘closest’ results to a particular query
Conclusions

• Found to work very well on Last.fm and Pandora where data sets were taken and tested
• Similar process could be used for movies and pictures
  • Mentions that in something like Facebook the weights might have to be adjusted based on how important social connection is to preference
HYPERGRAPH-BASED IMAGE REPRESENTATION

Article by Alain Bretto & Luc Gillibert
(2005, Université de Caen, France)

Presentation by Matt Dallmeyer
HYPERGRAPH-BASED IMAGE REPRESENTATION

• Paper begins with basic definitions:
  • Hypergraph, hyperedge, star, degree, etc.
  • Only considering undirected, connected, simple hypergraphs

• Image Adaptive Hypergraph Model:
  • Each pixel is a vertex, with adjacent pixels as its neighborhood
  • Vertices are connected in hypergraph if the difference in intensity (color, brightness, etc.) is below a calculated threshold
  • Applications in image processing are demonstrated
NOISE DETECTION / CANCELLATION

- With this representation, hyperedges with cardinality 1 are often noisy hyperedges
- The paper describes an algorithm to detect such noisy hyperedges
- This makes it easy to sample the surrounding pixels in order to correct the noise

![Natural image - Image corrupted - Noise detection]

**Fig. 2.** Example of IANH-based noise detection and cancellation.
SEGMENTATION

- Paper describes algorithm to partition image using hypergraph representation.
- Useful for image processing programs like the ‘Warholizer,’ which fills in similar areas of an image with solid colors.

Fig. 3. Example of IANH-based image segmentation.

Source: warholize.me
EDGE DETECTION

• With the help of some established algorithms, the paper details how to use the hypergraph representation to distinguish visible edges within an image.

• Useful for image processing and for object recognition.

**Fig. 4.** Example of IANH-based edge detection.
HYPERGRAPH-BASED IMAGE REPRESENTATION

- Curious how various filters/effects in Photoshop worked from a computer science standpoint
- I was also interested to learn that similar image representation is useful for 3-D object recognition