Exploiting Latent Features of Text and Graphs

Thesis Proposal

Justin Sybrandt

Clemson University

April 30, 2019

Peer-Reviewed Work:

- Sybrandt, J., Shtutman, M., & Safro, I. (2017, August). Moliere: Automatic biomedical hypothesis generation system. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1633-1642). ACM.
    - Acceptance rate 8.8%.
    - 1 Non-author citation.
- Sybrandt, J., Shtutman, M., & Safro, I. (2018, December). Large-scale validation of hypothesis generation systems via candidate ranking. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 1494-1503). IEEE.
    - Acceptance rate 18%.
- Sybrandt, J., Carrabba, A., Herzog, A., & Safro, I. (2018, December). Are abstracts enough for hypothesis generation? In 2018 IEEE International Conference on Big Data (Big Data) (pp. 1504-1513). IEEE.
    - Acceptance rate 18%.

Pending Works:

- Sybrandt, J., & Safro, I. Heterogeneous Bipartite Graph Embeddings. In-submission and not available online.
- Sybrandt, J., & Shaydulin, R., & Safro, I. Partition Hypergraphs with Embeddings. Not available online.
- Aksenova M., Sybrandt J., Cui B., Sikirzhytski V., Ji H., Odhiambo D., Lucius M., Turner J. R., Broude E., Pena E., Lizzaraga S., Zhu J., Safro I., Wyatt M. D., Shtutman M. (2019). Inhibition of the DDX3 prevents HIV-1 Tat and cocaine-induced neurotoxicity by targeting microglia activation. https://www.biorxiv.org/content/10.1101/591438v1
- Locke, W., Sybrandt, J., Safro, I., & Atamturktur, S. (2018, November 12). Using Drive-by Health Monitoring to Detect Bridge Damage Considering Environmental and Operational Effects. https://doi.org/10.31224/osf.io/ntfdp

Other Works:

Peer-Reviewed Extended Abstracts:

- Aksenova, M., Sybrandt, J., Cui, B., Lucius, M., Ji, H., Wyatt, M., Safro, I., Zhu, J., & Shtutman, M. (2019). Inhibition of the DEAD Box RNA Helicase 3 prevents HIV-1 Tat- and cocaine-induced neurotoxicity by targeting microglai activation. In 2019 Meeting of the NIDA Genetic Consortium. Extended Abstract & Poster

- Sybrandt, J., & Hick, J. (2015). Rapid replication of multi-petabyte file systems. Work in progress in the 2015 Parallel Data Storage Workshop. Poster in 2015 Super Computing.

Online Work:

- Shaydulin, R., & Sybrandt, J. (2017). To Agile, or not to Agile: A Comparison of Software Development Methodologies. arXiv preprint arXiv:1704.07469.
  - 5 Non-author citations.

Overview

- Latent Variables
  - Unobservable qualities of a dataset.
- Text Embeddings
  - Transferable textual latent features.
  - Correspond to semantic properties of words.
- Graph Embeddings
  - Underlying network features.
  - Correspond to roles, communities, and unobserved node-features.

Outline

Hypothesis Generation Background

Drug Discovery Problem Overview

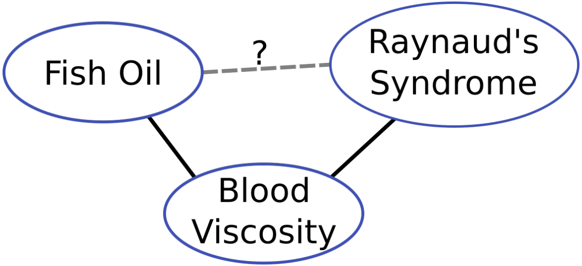
- Medical research is expensive and risky.
- Text mining can identify fruitful research directions before expensive experiments.

Wall Street Journal

Pfizer Ends Hunt for Drugs to Treat Alzheimer's and Parkinson's

Hypothesis Generation Overview

- PubMed contains over 27-million abstracts.
- 2-4k added daily.
- Hypothesis generation finds *implicitly* published relationships.



- Automatic Biomedical Hypothesis Generation System
- Basic Pipeline
  - Data collection
  - Network construction
  - Abstract identification
  - Topic modeling

Data Collection

- Titles & Abstracts

  Tumours evade immune control by creating hostile microenvironments that perturb T cell metabolism and effector function.

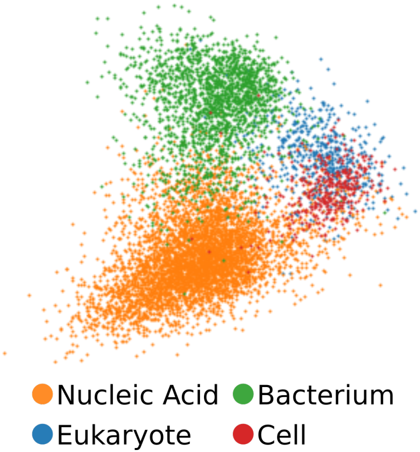
- Phrases (n-grams)
  - *"T cell metabolism"*
- Predicates
  - tumours $\rightarrow$ evade $\rightarrow$ immune control
- Unified Medical Language System

  Neoplasms

  tumor, tumour, oncological abnormality

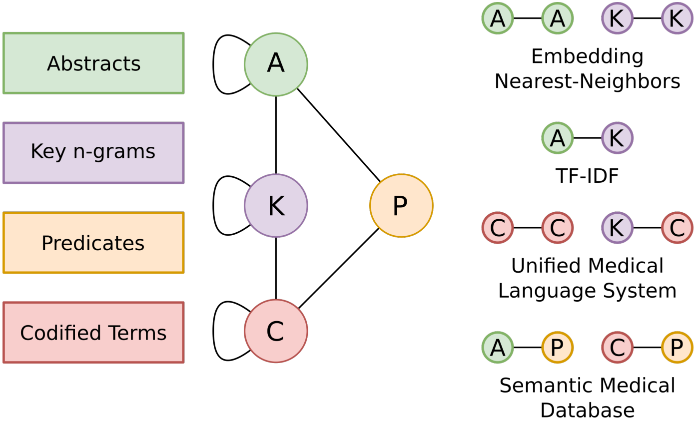
Medical Text Embedding

- Use fasttext to capture latent features [3].
- Semantically similar items are nearest neighbors.



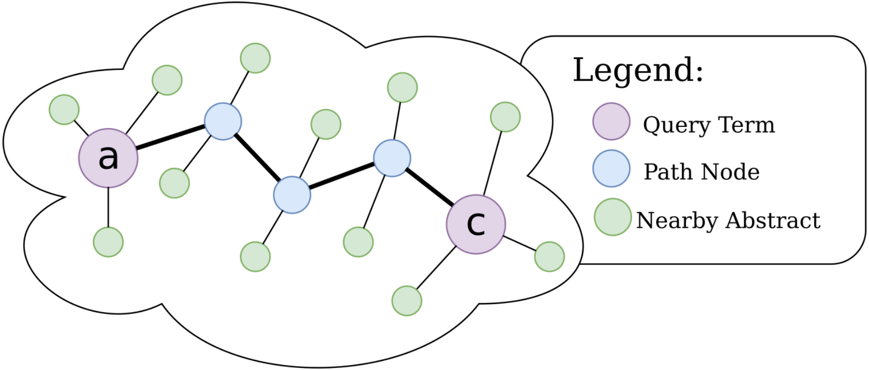
Network Construction

- Connections between data types.



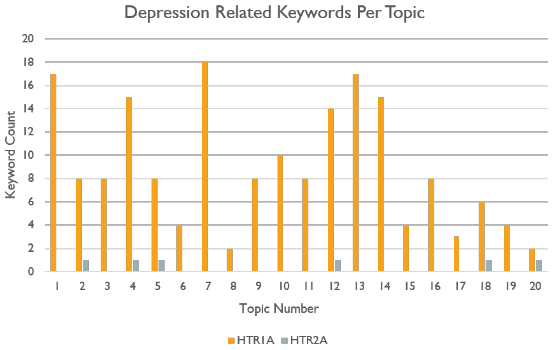
Abstract Identification

- Queries in form (a, c).
- Find shortest path.
- Identify abstracts near path.



Topic Modeling

- Run LDA topic modeling [2].
- Analyze word patterns across topics.
- Example: Venlafaxine interacts with HTR1A.

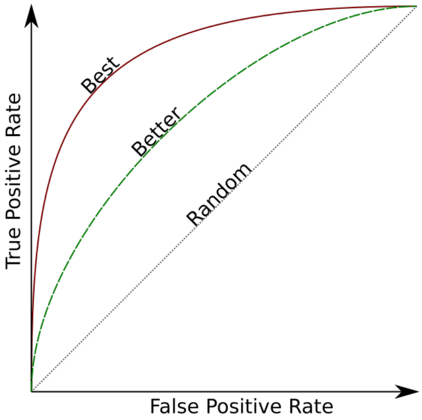


Automating Analysis

- Studying topics manually is infeasible.
- We propose *plausibility* ranking criteria.
  - Drug discovery is a ranking problem.
  - Allows for large-scale numerical evaluation.

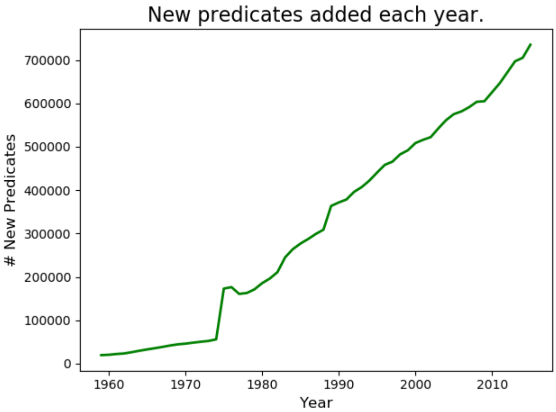
Validation Through Ranking

- Evaluate sorting criteria through ROC.
- "Synthetic" experiment, similar to drug discovery.
    - Identify recent discoveries.
    - Create negative samples.
    - Propose ranking criteria.



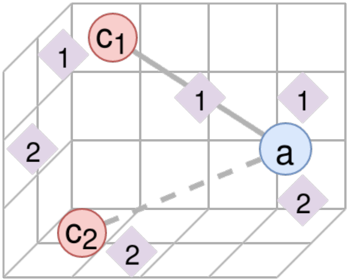
Validation Data

- Set "cut-date."
- Training Data:
  - All papers published prior.
- Validation Data:
  - Recent discoveries: SemMedDB pairs first occurring after.
  - Negative samples: Random UMLS pairs never occurring.



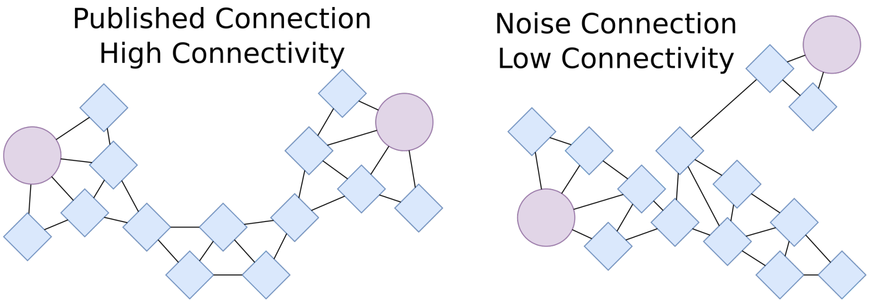
Proposed Ranking: Embedding

- Measure distance between query terms a and c.
- Calculate weighted centroid for each topic.
- Measure distances between terms and topics.



Proposed Ranking: Topics

- Create nearest-neighbors network from topic embeddings.
- Measure network statistics of shortest path $a - c$.



Combination Ranking

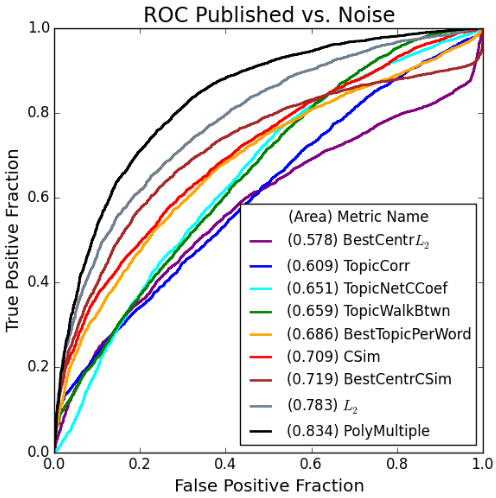
- Embedding measures
  - $\textsc{CSim}(a, c)$: Cosine Similarity of Embeddings
  - $L_2(a, c)$: Euclidean Distance of Embeddings
  - $\textsc{BestCentrCSim}(a, c, T)$: Maximum joint topic similarity.
  - $\textsc{BestCentrL}_2(a, c, T)$: Minimum joint topic distance.
  - $\textsc{BestTopPerWord}(a, c, T)$: Max of minimum joint similarity.
  - $\textsc{TopicCorr}(a, c, T)$: Correlation of Topic Distances

- Topic network measures
  - $\textsc{TopWalkLength}(a, c, T)$: Length of shortest path $a \sim c$
  - $\textsc{TopWalkBtwn}(a, c, T)$: Avg. $a \sim c$ betweenness centrality
  - $\textsc{TopWalkEigen}(a, c, T)$: Avg. $a \sim c$ eigenvalue centrality
  - $\textsc{TopNetCCoef}(a, c, T)$: Clustering coefficient of $\mathcal{N}$
  - $\textsc{TopNetMod}(a, c, T)$: Modularity of $\mathcal{N}$

$$\begin{aligned}\textsc{PolyMultiple}(a, c, T) = {} & \alpha_1 \cdot L_2^{\beta_1} + \alpha_2 \cdot \textsc{BestCenterL}_2^{\beta_2} \\ & + \alpha_3 \cdot \textsc{BestTopPerWord}(a, c, T)^{\beta_3} + \alpha_4 \cdot \textsc{TopCorr}(a, c, T)^{\beta_4} \\ & + \alpha_5 \cdot \textsc{TopWalkBtwn}(a, c, T)^{\beta_5} + \alpha_6 \cdot \textsc{TopNetCCoef}(a, c, T)^{\beta_6}\end{aligned}$$

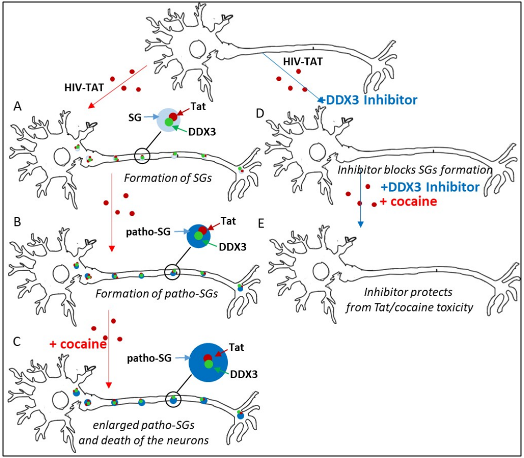
Validation Results

- Top ranking criteria:
  - PolyMultiple
  - L_2
  - BestCentrCSim



Testing in the Real World

- Apply ranking criteria for laboratory experiments.
- HIV-associated Neurodegenerative Disorder
  - 30% of HIV patents over 60 develop dementia.
  - We searched 40k gene relationships.
  - Identified DDX3X.



Are Abstracts Enough?

Are Abstracts Enough?

- Determine relationship between input and output.
  - Rebuild Moliere using different corpora.
  - Evaluate using ranking method.
- Explore effect of:
  - Corpus size.
  - Document length.
  - Abstracts vs. full-texts.

Full-text Challenges

- Expensive
  - Often requires licensing.
- Longer documents
  - 15.6x more words-per-document.
- Parsing
  - Figure, tables, references, PDFs.

Experiments

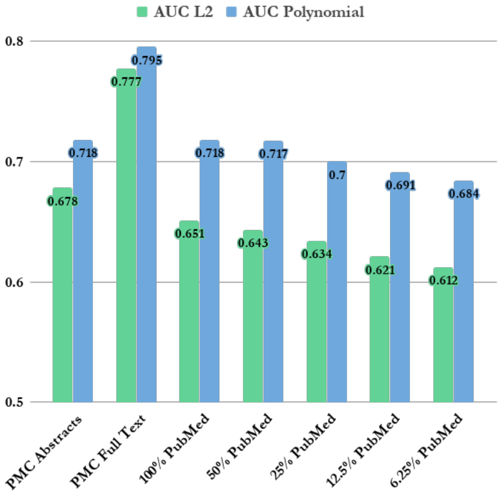
- Create separate *instances* of Moliere
  - From training vector space to POLYNOMIAL.
- Datasets
  - Iterative halves of PubMed.
  - PubMed Central full texts & abstracts.
- Evaluation
  - Create shared set of positive and negative hypotheses.
  - Cut year of 2015.
  - Rank and calculate ROC curves.

Dataset Details

Corpus	Total Words	Unique Words	Corpus Size	Median Words per Document
PMC Abstracts	109,987,863	673,389	1,086,704	102
PMC Full-Text	1,860,907,606	6,548,236	1,086,704	1594
PubMed	1,852,059,044	2,410,130	24,284,910	71
1/2 PubMed	923,679,660	1,505,672	12,142,455	71
1/4 PubMed	460,384,928	920,734	6,071,227	71
1/8 PubMed	229,452,214	565,270	3,035,613	71
1/16 PubMed	114,385,607	349,174	1,517,806	71

Abstract vs Full Text Results

- Summarize performance with L_2 and $\textsc{Polynomial}$ metrics.
  - $\textsc{Polynomial}$ evaluates whole system.
  - L_2 evaluates embedding.



Beyond Quality

- Full text improves performance quality by about 10%.
- Full text increases *runtime* from 2m to 1.5h.
- Topic modeling:
  - Primary runtime increase.
  - Less interpretable topics.

Effects

- Corpus size:
  - Longer corpus helps slightly.
- Document length:
  - Longer documents significantly improve word embeddings.
  - Increase runtime.
- Abstracts vs. Full text
  - Content in full texts not found in abstracts.

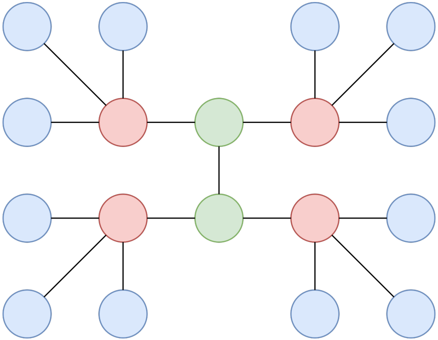
Hypothesis Generation Summary

- Moliere
  - Use embedding to make network, find abstracts, perform topic modeling.
- Validation via Candidate Ranking
  - Propose metrics to quality embedding and topic model qualities.
  - Evaluate recently published results, extend to real-world experiments.
- Are Abstracts Enough?
  - Compare performance of system across different corpus qualities.
  - Full texts improve performance at large runtime penalty.

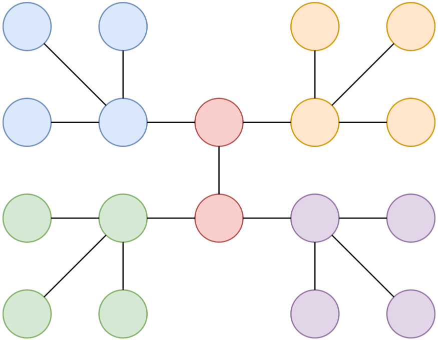
Graph Embedding Background

Two Different Graph Similarities [10]

- Structural Similarity

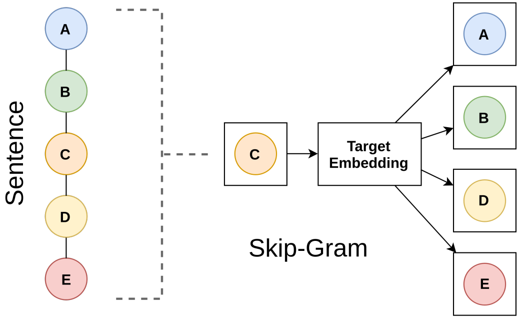


- Homophilic Similarity



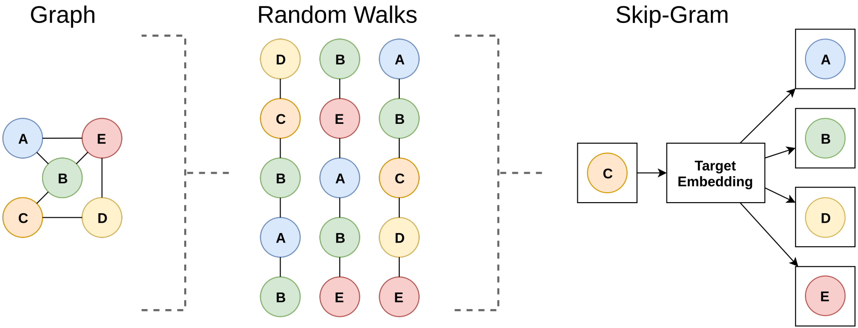
Skip-Gram Model [17]

- Sample *windows* centered on target word.
- Predict leading & trailing context from target embedding.
- Assumption: "Similar words share similar company."



Deepwalk [19]

- Sample random walks from graph.
- Interpret walks as "sentences."
- Apply Skip-Gram model.



LINE [21]

- Sample first- & second-order neighbors.
- Fit observed samples to embeddings.
  - Observed probability between u & v:

$$p(u, v) = \frac{w_{uv}}{\sum_{(i,j) \in E} w_{ij}}$$

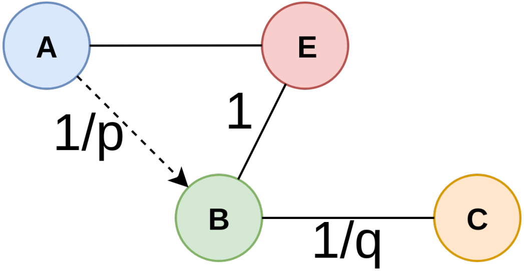
  - Predicted probability:

$$\hat{p}(u, v, \epsilon) = \sigma(\epsilon(u)^\intercal \epsilon(v))$$

  - Minimize KL-Divergence between p and $\hat{p}$.

Node2Vec [10]

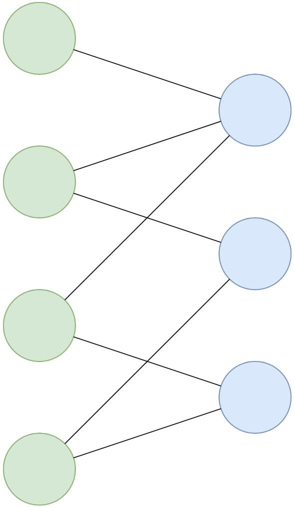
- Combine structural and homophilic.
  - Blends breadth- and depth-first walks.
  - Adds return- and out-parameters.



Heterogeneous Bipartite Graph Embedding

Heterogeneous Bipartite Graphs

- Contains two node types.
  - $G_B = (V, E)$
  - $V = A \cup B$
  - A and B are disjoint.
- Neighborhood $\Gamma(i)$.
  - If $i \in A$, then $\Gamma(i) \subseteq B$.



Proposed Methods

- Boolean Heterogeneous Bipartite Embedding
  - Weight all samples equally.
  - Sample direct and first-order relationships.
- Algebraic Heterogeneous Bipartite Embedding
  - Weight sampling with algebraic distance.
  - Sample direct, first-, and second-order relationships.

Both Methods:

- Enable type-specific latent features.
- Make only same-type comparisons.

Algorithm Outline

- Observed similarities:
  - $\mathbb{S}_A(i,j)$, $\mathbb{S}_B(i,j)$, $\mathbb{S}_{AB}(i,j)$
- Predicted similarities w.r.t. embedding ($\epsilon : V \to \mathbb{R}^k$):
  - $\widetilde{\mathbb{S}}_A(i,j,\epsilon)$, $\widetilde{\mathbb{S}}_B(i,j,\epsilon)$, $\widetilde{\mathbb{S}}_{AB}(i,j,\epsilon)$
- Optimize:
  - Minimize difference between $\mathbb{S}$ and $\widetilde{\mathbb{S}}$.

Boolean Heterogeneous Bipartite Embedding

Boolean Observations

- Observed cross-type relationships:

$$\mathbb{S}_{AB}(i,j) = \begin{cases} 1 & i \in \Gamma(j) \\ 0 & \text{otherwise} \end{cases}$$

- Observed same-type relationships:

$$\mathbb{S}_{A}(i,j) = \mathbb{S}_{B}(i,j) = \begin{cases} 1 & \Gamma(i) \cap \Gamma(j) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Boolean Predictions

- Predicted same-type relationships:

$$\widetilde{\mathbb{S}}_A(i, j, \epsilon) = \sigma\left(\epsilon(i)^\intercal \epsilon(j)\right)$$

- Predicted cross-type relationships:
  - Decompose into same-type relationships.

$$\widetilde{\mathbb{S}}_{AB}(i, j, \epsilon) = \mathop{\mathbb{E}}_{k \in \Gamma(j)}\left[\widetilde{\mathbb{S}}_A(i, k, \epsilon)\right] \mathop{\mathbb{E}}_{k \in \Gamma(i)}\left[\widetilde{\mathbb{S}}_B(k, j, \epsilon)\right]$$

Boolean Optimization

- Loss for a particular similarity:

$$O_X = \sum_{i,j \in V} \tilde{\mathbb{S}}_X(i,j,\epsilon) \log \left(\frac{\mathbb{S}_X(i,j)}{\tilde{\mathbb{S}}_X(i,j,\epsilon)} \right)$$

- Optimization:

$$\min_{\epsilon} O_A + O_{AB} + O_B$$

Algebraic Heterogeneous Bipartite Embedding

Algebraic Distance I

- Stationary iterative relaxation.
- Algebraic distance for hypergraphs [20], adapted for bipartite graphs:

$$a_0 \sim [0, 1]$$

$$a_{t+1}(i) = \lambda a_t(i) + (1 - \lambda)\frac{\sum_{j \in \Gamma(i)} a_t(j)|\Gamma(j)|^{-1}}{\sum_{j \in \Gamma(i)} |\Gamma(j)|^{-1}}$$

- Between each iteration, rescale to $[0, 1]$.

Algebraic Distance II

- Run $T = 20$ algebraic distance trials until stabilization.
- Summarize distances across all trials:

$$d(i,j) = \sqrt{\sum_{t'=1}^{T} \left(a_{\infty}^{(t')}(i) - a_{\infty}^{(t')}(j) \right)^2}$$

- Summarize similarity between nodes:

$$s(i,j) = \frac{\sqrt{T} - d(i,j)}{\sqrt{T}}$$

Algebraic Observations

- Same-type:
  - Two A nodes are similar if any B node is highly similar to both.

$$\mathbb{S}'_A(i,j) = \mathbb{S}'_B(i,j) = \max_{k \in \Gamma(i) \cap \Gamma(j)} \min\left(s(i,k), s(k,j)\right)$$

- Cross-type, sample both direct and second-order neighbors:
  - Decompose cross-type comparisons to same-typed neighborhoods.

$$\mathbb{S}'_{AB}(i,j) = \max\left(\max_{k \in \Gamma(j)} \mathbb{S}'_A(i,k), \max_{k \in \Gamma(i)} \mathbb{S}'_B(k,j)\right)$$

Algebraic Predictions

- Predicted same-type relationships:

$$\widetilde{\mathbb{S}}'_A(i, j, \epsilon) = \max\left(0, \epsilon(i)^\intercal \epsilon(j)\right)$$

- Predicted same- and cross-typed relationships:

$$\widetilde{\mathbb{S}}'_{AB}(i, j, \epsilon) = \mathop{\mathbb{E}}_{k \in \Gamma(j)}\left[\widetilde{\mathbb{S}}'_A(i, k, \epsilon)\right] \mathop{\mathbb{E}}_{k \in \Gamma(i)}\left[\widetilde{\mathbb{S}}'_B(k, j, \epsilon)\right]$$

Algebraic Optimization

- Loss for a particular similarity:

$$O_X^{'} = \mathop{\mathbb{E}}_{i,j \in V} \left(\mathbb{S}_X^{'}(i,j) - \tilde{\mathbb{S}}_X^{'}(i,j,\epsilon) \right)^2$$

- Optimization:

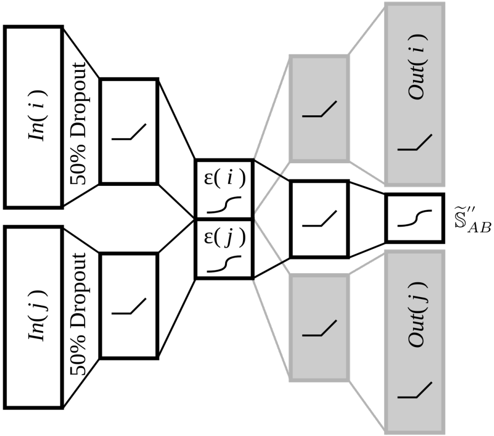
$$\min_{\epsilon} O'_A + O'_{AB} + O'_B$$

New Embedding Methods

- Boolean Heterogeneous Bipartite Embedding (BHBE)
  - Observes existence of relationships.
  - Predicts using $\sigma(\epsilon(i)^\intercal \epsilon(j))$.
  - Minimizes KL-Divergence.
- Algebraic Heterogeneous Bipartite Embedding (AHBE)
  - Observes relationships weighted through algebraic distance.
  - Predicts using $\max(0, \epsilon(i)^\intercal \epsilon(j))$.
  - Minimizes Mean-Squared Error.

Combination Embedding

- Learn joint representation to combine AHBE & BHBE.
  - Direct encoding predicts links through joint embedding.
  - Auto-regularized encoding also enforces all latent features are captured.

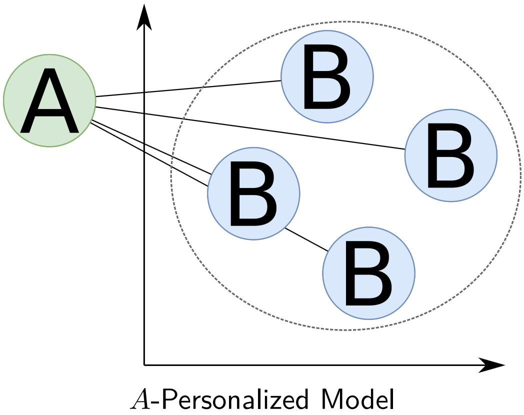


Evaluation I

- Link prediction task.
  - Select graph, delete % of edges.
  - Embed remaining graph.
  - Use embeddings to recover removed edges.
- Explore varying hold-out percentages.
  - From 10% to 90% splits.
  - Increments of 10%.

Evaluation II

- Models:
  - A- and B-Personalized.
    - Train a SVM for each node in the training set.
    - Each SVM detects an embedding region containing neighbors.
  - Unified.
    - Train a shallow neural network.
    - Predict links given one embedding of each type.

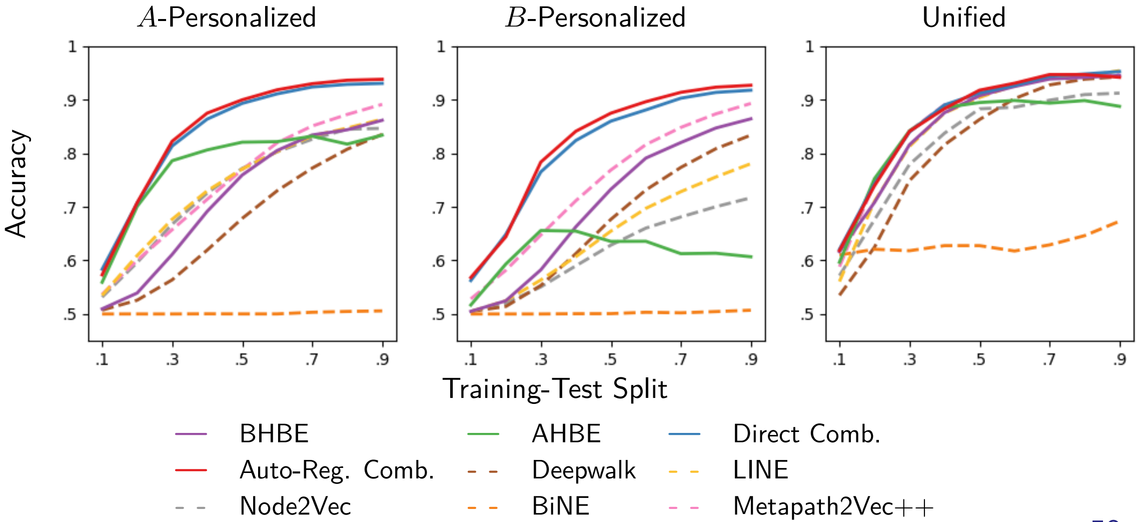


Benchmark

Graph	$\|A\|/\|B\|$	$\Gamma(i \in A)$ md	$\Gamma(i \in A)$ max	$\Gamma(j \in B)$ md	$\Gamma(j \in B)$ max	SR	LCP
Amazon	$16{,}716/5{,}000$	3	49	8	328	75.8	1.6
DBLP	$93{,}432/5{,}000$	1	12	8	7,556	174.7	81.7
Friendster	$220{,}015/5{,}000$	1	26	133	1,612	80.3	58.3
Livejournal	$84{,}438/5{,}000$	1	20	16	1,441	100.9	27.0
MadGrades	$11{,}951/6{,}462$	3	39	4	393	57.3	99.7
YouTube	$39{,}841/5{,}000$	1	54	4	2,217	113.3	80.6

Table: Graph summary. We report the median (md) and max degree for each node set, as well as the Spectral Radius (SR) and the percentage of the largest connected component (LCP).

MadGrades: UW. Instructor-Course Network ($|A| = 11,951,\ |B| = 6,462$)



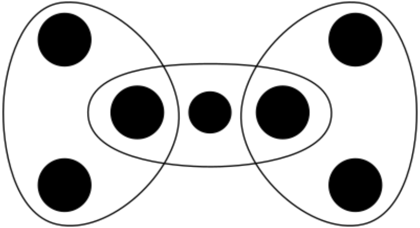
Results

- Algebraic HBE
  - Best detects trends among high-degree nodes.
  - Stability issues with larger graphs.
- Boolean HBE
  - Outperforms typical state-of-the-art methods, competitive with other heterogeneous methods.
  - Robust across trials.
- Combinations
  - BHBE and ABHE find different latent features.
  - Bootstrap performance above state-of-the-art.

Partition Hypergraphs with Embeddings

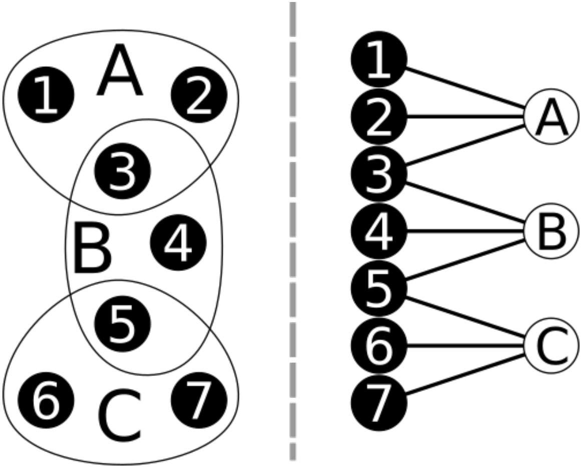
Hypergraphs

- Generalization: *hyperedges* contain any number of nodes.
- $H = (V, E)$
  - $V = \{v_1, v_2, \ldots, v_n\}$
  - $E = \{e_1, e_2, \ldots, e_m\}$
  - $e_i \subseteq V$



Hypergraph Star Expansion

- Map hyperedges to nodes.



Hypergraph Partitioning: Problem Description I

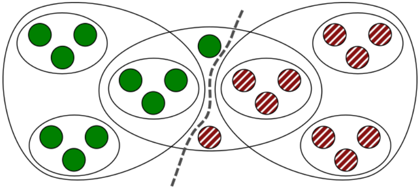
- Problem: Split V into k disjoint sets...
  - of approximately equal size.
  - minimizing an objective of cut hyperedges.

Hypergraph Partitioning: Problem Description II

- Partition:
  - $V = V_1 \cup V_2 \cup \cdots \cup V_k$
  - $\forall (V_i, V_j), V_i \cap V_j = \emptyset$
- $E_{\text{cut}} = \{e \in E : \nexists V_i, e \subseteq V_i\}$
- Metrics:

$$\lambda_{\text{cut}} := |E_{\text{cut}}|$$

$$\lambda_{k-1} := \sum_{e \in E_{\text{cut}}} |\{V_i : V_i \cap e \neq \emptyset\}| - 1$$

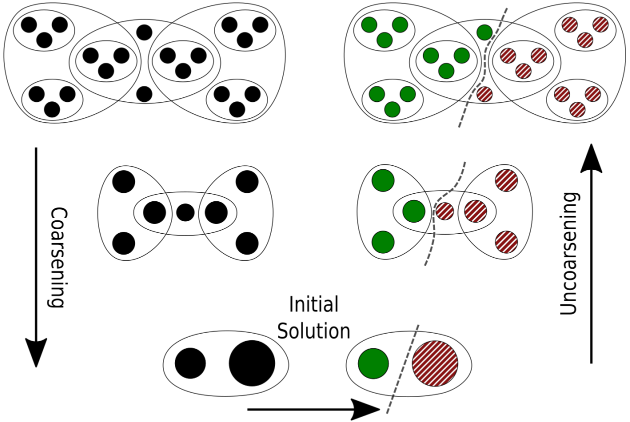


Hypergraph Partitioning: Problem Description III

- Hypergraph partitioning is NP-Hard...
  - to solve [15].
  - to approximate [5].

Multilevel Heuristic

- Steps:
  - Coarsen
  - Initial Solution
  - Uncoarsen: Interpolate & local search.
- Paradigms:
  - $(\log n)$-Level: Each level, pair almost all nodes.
  - n-Level: Each level, pair two nodes.

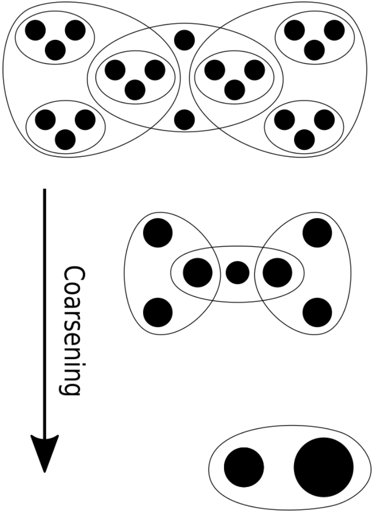


Coarsening

- Desired coarsening properties [12]:
  - Reduce number of nodes & hyperedges.
  - Remain structurally similar.

Contribution

Use hypergraph embeddings to better coarsen nodes.



Typical Coarsening

- Assigning all nodes & hyperedges uniform weights: $w_i = 1$.
- Measure similarities (e.g. hyperedge inner product).

$$S_E(u,v) = \sum_{e \in E | u,v \in e} \frac{w_e}{|e| - 1}$$

- Match nodes into (u, v).
- Contract u and v into x.
  - $w_x = w_u + w_v$
  - x participates in all hyperedges of u and v.

Embedding-based Coarsening

- Quantify similarity within embedding.

$$S_\epsilon(u,v) = \epsilon(u)^\intercal \epsilon(v)$$

- Prioritize nodes with highly similar neighbors.
- Measure node similarities:

$$S(u,v) = \frac{S_E S_\epsilon}{w_u w_v}$$

Embedding-based Coarsening Algorithm

- Sort V by each node's highest neighbor similarity.

$$\textsc{SortingCriteria}_u = \max_{v \in \Gamma(u)} S_\epsilon(u, v)$$

- In sorted order, pair nodes.

$$\textsc{Partner}_u = \operatorname*{argmax}_{v \in \Gamma(u)} \frac{S_E(u, v) S_\epsilon(u, v)}{w_u w_v}$$

- Merge (u, v) to coarse node x.
    - Assign $\epsilon(x)$ to be the centroid of its contained nodes.

Evaluation Method I

- Proposed Implementations:
  - Zoltan: $(\log n)$-Level, fast and highly parallel.
  - KaHyPar: n-Level, high-quality partitioning.
  - KaHyPar Flow: n-Level, best known algorithm.
- Considered Embeddings:
  - MetaPath2Vec++
  - Node2Vec
  - AHBE, BHBE
  - Combinations (AHBE+BHBE), (All)

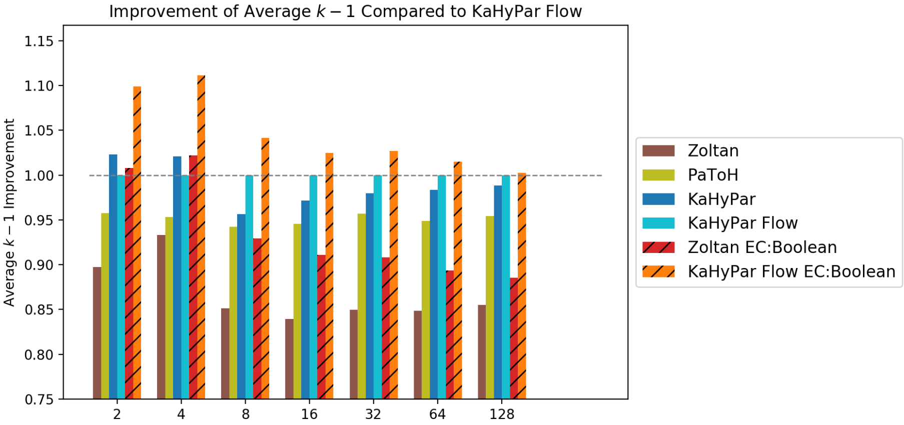
Evaluation Method II

- Baseline Algorithms:
  - hMetis [14]
  - PaToH [6]
  - Zoltan [8]
  - KaHyPar (w/ community-based coarsening) [13]
  - KaHyPar Flow (w/ flow-based refinement) [12]

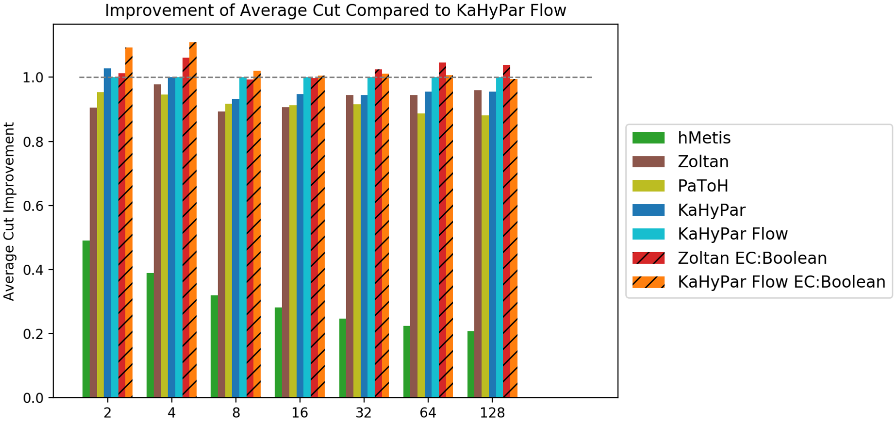
Evaluation Method III

- Benchmark Graphs
    - 86 graphs from SuiteSparse Matrix Collection.
    - 10 graphs designed to interfere with typical coarsening.
- k values: $2, 4, 8, \ldots, 128$.
- 20 trials per combination.
- Imbalance tolerance of 3%.

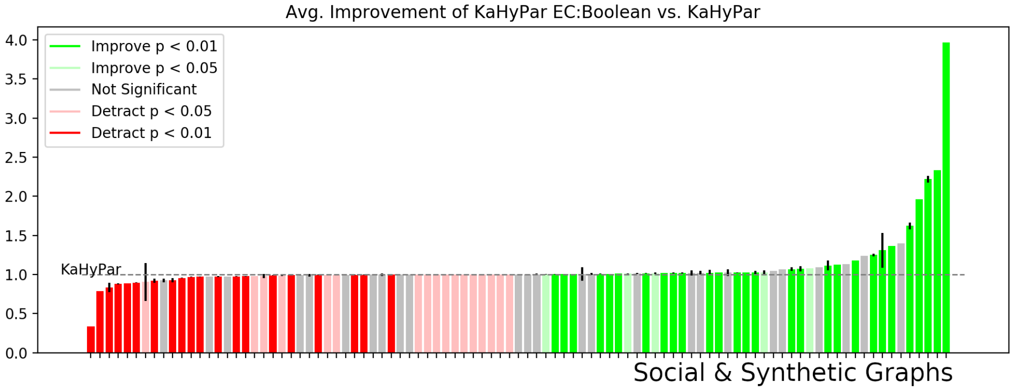
Partitioning Results I



Partitioning Results II



Partitioning Results III



Partitioning Summary

- Embeddings improve multilevel hypergraph partitioning.
  - Latent features indicate relevant structural similarities.
  - Embedding similarity prioritizes and matches nodes.
- Most important for small partition counts (k).
  - Embeddings capture key clusters.
  - Centroids during coarsening smooth some finer details.
- Latent features are most important for particular graphs.
  - Social networks.
  - Synthetic graphs.

Proposed Work

Future Directions

- Bias in Scientific Embeddings
  - Detect confirmation bias.
  - Normalize effect of “group think.”
- Hybrid Knowledge Graph Mining
  - Train text and graph embeddings.
  - Formulate hypothesis generation for deep learning.

Bias in Word Embeddings

- Recent work finds gender stereotypes in word embeddings [4].
- Biases exist in science.
  - Confirmation bias [18]
  - Over-interpreting noise [7].
  - P-hacking [11].
- Example P53
  - "[A]valanche of research" [22]
  - What connections to focus on? Which are noise?

Hybrid Knowledge Graph Mining for Hypothesis Generation

- Knowledge Graphs
  - Triplets, similar to SemMedDB Predicates
  - Typed relationships
- Specialized Techniques
  - Edge2Vec [9].
  - Use text to augment graphs [16].
  - SciBERT [1].

Attention-Based Hypothesis Generation

- Attention-mechanism creates interpretable results.
  - Assigns weights to relevant input.
- Unified deep-learning model.
  - Input: Embeddings for text and network features.
  - Potential Outputs:
    - Connection strength.
    - Connection type.
    - Automatic summary.

Timeline

Date	Accomplishment
April 2019	Dissertation proposal.
August 2019	Return from summer internship. Begin exploring bias in scientific embeddings.
November 2019	Complete analysis of bias in scientific embeddings. Begin exploring deep learning on knowledge graphs for hypothesis generation.
April 2020	Complete analysis of deep learning and knowledge graphs for hypothesis generation.
June 2020	Dissertation defense.

Table: Timeline of proposed work.

Acknowledgments

- NSF MRI #1725573
- NSF NRT-DESE #1633608

Bibliography I

[1] Beltagy, I., Cohan, A., and Lo, K.
Scibert: Pretrained contextualized embeddings for scientific text.
arXiv preprint arXiv:1903.10676 (2019).

[2] Blei, D. M., Ng, A. Y., and Jordan, M. I.
Latent dirichlet allocation.
993–1022.

[3] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T.
Enriching word vectors with subword information.
Transactions of the Association for Computational Linguistics 5 (2017), 135–146.

[4] Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T.
Man is to computer programmer as woman is to homemaker? debiasing word embeddings.
In *Advances in neural information processing systems* (2016), pp. 4349–4357.

Bibliography II

[5] Bui, T. N., and Jones, C.
Finding good approximate vertex and edge partitions is np-hard.
Information Processing Letters 42, 3 (1992), 153–159.

[6] Çatalyürek, Ü., and Aykanat, C.
Patoh (partitioning tool for hypergraphs).
Encyclopedia of Parallel Computing (2011), 1479–1487.

[7] de Groot, A. D.
The meaning of "significance" for different types of research [translated and annotated by eric-jan wagenmakers, denny borsboom, josine verhagen, rogier kievit, marjan bakker, angelique cramer, dora matzke, don mellenbergh, and han lj van der maas].
Acta psychologica 148 (2014), 188–194.

Bibliography III

[8] Devine, K. D., Boman, E. G., Heaphy, R. T., Bisseling, R. H., and Catalyurek, U. V.
Parallel hypergraph partitioning for scientific computing.
In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International* (2006), IEEE, pp. 10–pp.

[9] Gao, Z., Fu, G., Ouyang, C., Tsutsui, S., Liu, X., and Ding, Y.
edge2vec: Learning node representation using edge semantics.
arXiv preprint arXiv:1809.02269 (2018).

[10] Grover, A., and Leskovec, J.
node2vec: Scalable feature learning for networks.
In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (2016), ACM, pp. 855–864.

Bibliography IV

[11] Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., and Jennions, M. D.
The extent and consequences of p-hacking in science.
PLoS biology 13, 3 (2015), e1002106.

[12] Heuer, T., Sanders, P., and Schlag, S.
Network Flow-Based Refinement for Multilevel Hypergraph Partitioning.
In *17th International Symposium on Experimental Algorithms (SEA 2018)* (2018), pp. 1:1–1:19.

[13] Heuer, T., and Schlag, S.
Improving coarsening schemes for hypergraph partitioning by exploiting community structure.
In *16th International Symposium on Experimental Algorithms, (SEA 2017)* (2017), pp. 21:1–21:19.

[14] Karypis, G.
hmetis 1.5: A hypergraph partitioning package.
http://www. cs. umn. edu/˜ metis (1998).

Bibliography V

[15] Lengauer, T.
Combinatorial algorithms for integrated circuit layout.
Springer Science & Business Media, 2012.

[16] Li, C., Lai, Y.-Y., Neville, J., and Goldwasser, D.
Joint embedding models for textual and social analysis.
In *The Workshop on Deep Structured Prediction* (2017).

[17] Mikolov, T., Chen, K., Corrado, G., and Dean, J.
Efficient estimation of word representations in vector space.
arXiv preprint arXiv:1301.3781 (2013).

Bibliography VI

[18] Munafò, M. R., Nosek, B. A., Bishop, D. V., Button, K. S., Chambers, C. D., Du Sert, N. P., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., and Ioannidis, J. P.
A manifesto for reproducible science.
Nature human behaviour 1, 1 (2017), 0021.

[19] Perozzi, B., Al-Rfou, R., and Skiena, S.
Deepwalk: Online learning of social representations.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), ACM, pp. 701–710.

[20] Shaydulin, R., Chen, J., and Safro, I.
Relaxation-based coarsening for multilevel hypergraph partitioning.
Multiscale Modeling & Simulation 17, 1 (2019), 482–506.

Bibliography VII

[21] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q.
Line: Large-scale information network embedding.
In *Proceedings of the 24th International Conference on World Wide Web* (2015), International World Wide Web Conferences Steering Committee, pp. 1067–1077.

[22] Vogelstein, B., Lane, D., and Levine, A. J.
Surfing the p53 network.
Nature 408, 6810 (2000), 307.

Outline