Node Representation Learning

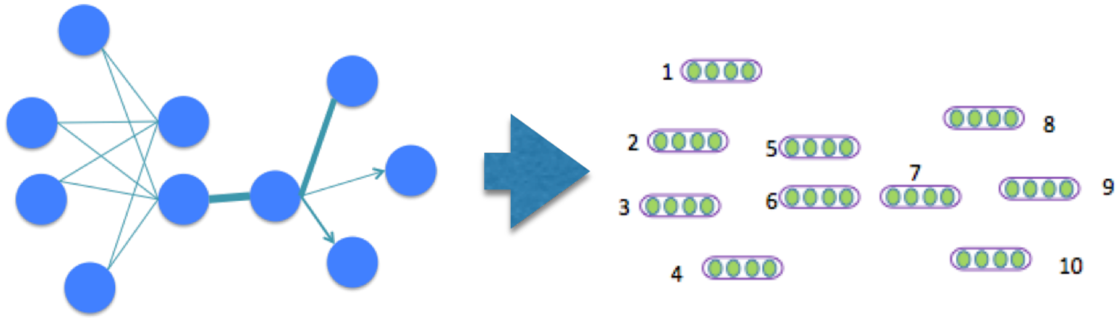
Jian Tang
HEC Montreal
CIFAR AI Chair, Mila
Email: jian.tang@hec.ca

Outline

- Node Representation Methods
  - LINE, DeepWalk, node2vec
- Graph and High-dimensional Data Visualization
  - LargeVis
- Knowledge Graph Embedding
  - RotatE (Sun et al., ICLR'19)
- A High-performance Node Representation System (Zhu et al., WWW'19)

Problem Definition: Node Embedding

- Given a network/graph G=(V, E, W), where V is the set of nodes, E is the set of edges between the nodes, and W is the set of weights of the edges, the goal of ***node embedding*** is to represent each node *i* with ***a vector*** $\vec{u}_i \in R^d$, which preserves the structure of networks.



Networks

Node representations

Related Work

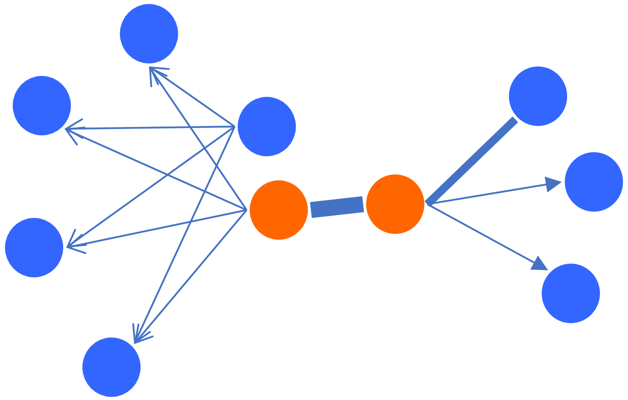
- Classical graph embedding algorithms
  - MDS, IsoMap, LLE, Laplacian Eigenmap, …
  - Hard to scale up
- Graph factorization (Ahmed et al. 2013)
  - Not specifically designed for network representation
  - Undirected graphs only
- Neural word embeddings (Bengio et al. 2003)
  - Neural language model
  - word2vec (skipgram), paragraph vectors, etc.

LINE: Large-scale Information Network Embedding (Tang et al., Most Cited Paper of WWW 2015)

- Arbitrary types of networks
  - Directed, undirected, and/or weighted
- Clear objective function
  - Preserve the first-order and second-order proximity
- Scalable
  - Asynchronous stochastic gradient descent
  - Millions of nodes and billions of edges: a coupe of hours on a single machine

Jian Tang, Meng Qu, Mingzhe Wang, Jun Yan, Ming Zhang and Qiaozhu Mei. **LINE: Large-scale Information Network Embedding**. WWW'15

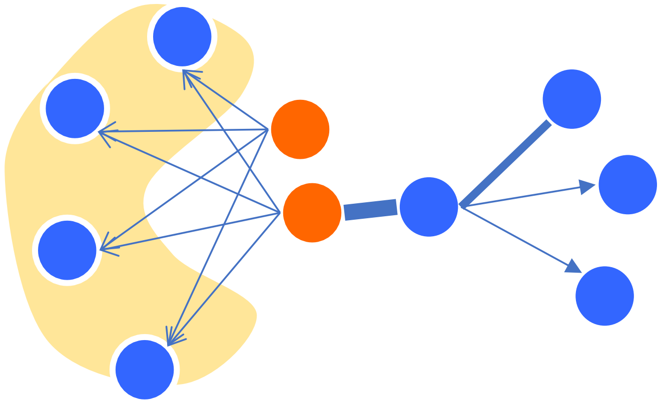
First-order Proximity



- The local pairwise proximity between the nodes
- However, many links between the nodes are not observed
  - Not sufficient for preserving the entire network structure

Second-order Proximity

“The degree of overlap of two people’s friendship networks correlates with the strength of ties between them” --Mark Granovetter



“You shall know a word by the company it keeps” --John Rupert Firth

- Proximity between the neighborhood structures of the nodes

Preserving the First-order Proximity (LINE 1st)

- Distributions: : (defined on the undirected edge i - j)

Empirical distribution of first-order proximity:

$$\hat{p}_1(v_i, v_j) = \frac{w_{ij}}{\sum_{(m,n)\in E} w_{mn}}$$

Model distribution of first-order proximity:

$$p_1(v_i, v_j) = \frac{\exp(\vec{u}_i^T \vec{u}_j)}{\sum_{(m,n)\in V\times V} \exp(\vec{u}_m^T \vec{u}_n)}$$

$\vec{u}_i$: Embedding of i

- Objective:

$$O_1 = KL(\hat{p}_1, p_1) = -\sum_{(i,j)\in E} w_{ij} \log p_1(v_i, v_j)$$

Preserving the Second-order Proximity (LINE 2nd)

- Distributions: (defined on the directed edge i -> j)

Empirical distribution of neighborhood structure:

$$\hat{p}_2(v_j \mid v_i) = \frac{w_{ij}}{\sum_{k \in V} w_{ik}}$$

Model distribution of neighborhood structure:

$$p_2(v_j \mid v_i) = \frac{\exp(\vec{u}'^{T}_{i} \vec{u}_j)}{\sum_{k \in V} \exp(\vec{u}'^{T}_{k} \vec{u}_i)}$$

- Objective:

$$O_2 = \sum_i KL(\hat{p}_2(\cdot \mid v_i), p_2(\cdot \mid v_i)) = -\sum_{(i,j) \in E} w_{ij} \log p_2(v_j \mid v_i)$$

Optimization Tricks

- Stochastic gradient descent + Negative Sampling
  - Randomly sample an edge and multiple negative edges
- The gradient w.r.t the embedding with edge (i, j)

$$\frac{\partial O_2}{\partial \vec{u}_i} = w_{ij} \frac{\partial \log \hat{p}_2(v_j \mid v_i)}{\partial \vec{u}_i}$$

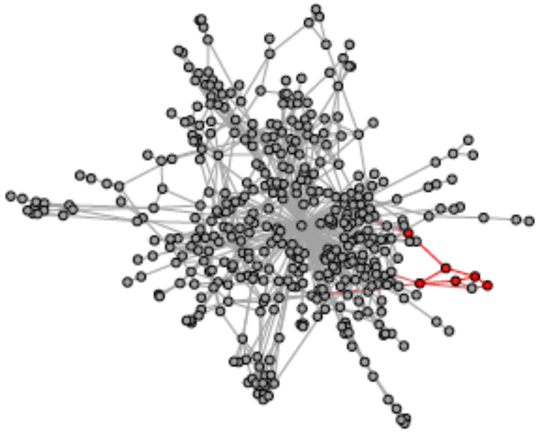
- Problematic when the variances of weights of the edges are large
  - The variance of the gradients are large
- Solution: edge sampling
  - Sample the edges according to their weights and treat the edges as binary
- Complexity: O(d*K*|E|)
  - Linear to the dimensionality d, the number of negative samples K, and the number of edges

Discussion

- Embed nodes with few neighbors
  - Expand the neighbors by adding higher-order neighbors
  - Breadth-first search (BFS)
  - Adding only second-order neighbors works well in most cases
- Embed new nodes
  - Fix the embeddings of existing nodes
  - Optimize the objective w.r.t. the embeddings of new nodes

DeepWalk (Perozzi et al. 2014)

- Learning node representations with the technique for learning word representations, i.e., Skipgram
- Treat *random walks on networks* as sentences



Random walk generation (generate node contexts through random search)

$$\begin{array}{l} 3 \\ \left.1\right]^{v_j} \\ 5 \\ 1 \\ \vdots \end{array}$$

$$p(v_j \mid v_i) = \frac{\exp(\vec{u}'^{T}_{i}\,\vec{u}_j)}{\sum_{k\in V}\exp(\vec{u}'^{T}_{k}\,\vec{u}_i)}$$

Predict the nearby nodes in the random walks

Bryan Perozzi, Rami Al-Rfou, Steven Skiena. DeepWalk: Online Learning of Social Representations. KDD’14

Node2Vec (Grover and Leskovec, 2016)

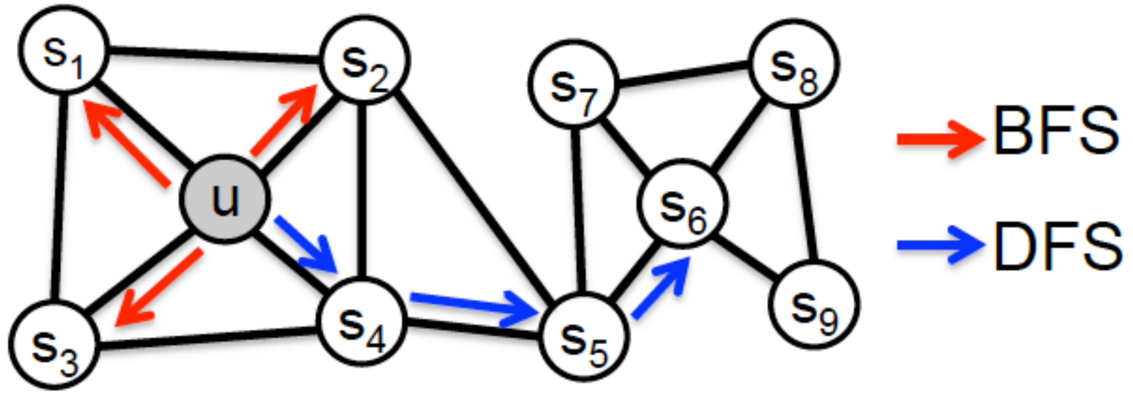
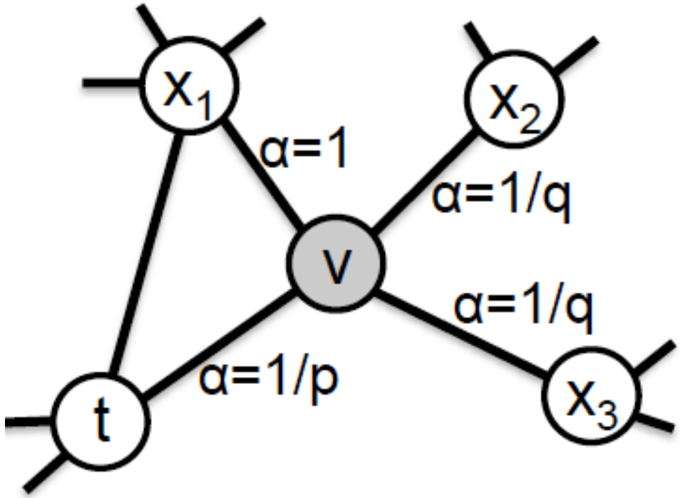


Figure 1: BFS and DFS search strategies from node u ($k = 3$).

- Find the node context with a hybrid strategy of
  - Breadth-first Sampling (BFS): **homophily**
  - Depth-first Sampling (DFS): **structural equivalence**

Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. KDD'16

Expand Node Contexts with Biased Random Walk



- Biased random walk with two parameters p and q
  - p: controls the probability of revisiting a node in the walk
  - q: controls the probability of exploring “outward” nodes
  - Find optimal p and q through cross-validation on labeled data
- Optimized through similar objective as LINE with first-order proximity

Comparison between LINE, DeepWalk, and Node2Vec

Algorithm	Neighbor Expansion	Proximity	Optimization	Validation Data
LINE	BFS	1^{st} or 2^{nd}	Negative Sampling	No
DeepWalk	Random	2^{nd}	Hierarchical Softmax	No
Node2Vec	BFS + DFS	1^{st}	Negative Sampling	Yes

Applications

- Node classification (Perozzi et al. 2014, Tang et al. 2015a, Grover et al. 2015)
- Node visualization (Tang et al. 2015a)
- Link prediction (Grover et al. 2015)
- Recommendation (Zhao et al. 2016)
- Text representation (Tang et al. 2015a, Tang et al. 2015b)
- …

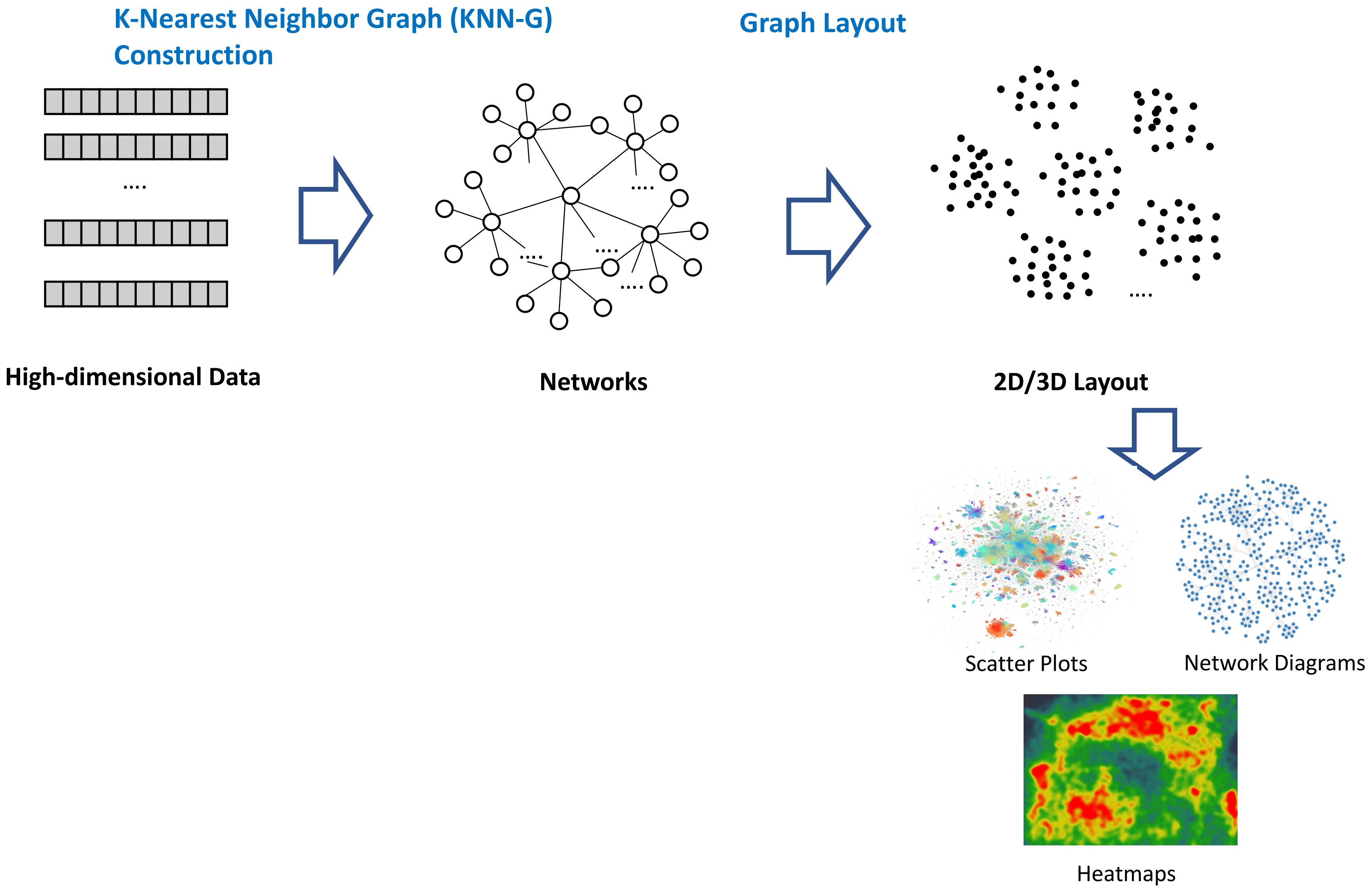
Many Extensions …

- Leverage global structural information (Cao et al. 2015)
- Non-linear methods based on autoencoders (Wang et al. 2016)
- Matrix-factorization based approaches (Qiu et al. 2018)
- Directed network embedding (Ou et al. 2016)
- Signed network embedding (Wang et al. 2017)
- Multi-view networks (Qu and Tang et al. 2017)
- Networks with node attributes (Yang et al. 2015)
- Heterogeneous networks (Chang et al. 2015)
- Task-specific network embedding (Chen et al. 2017)

Outline

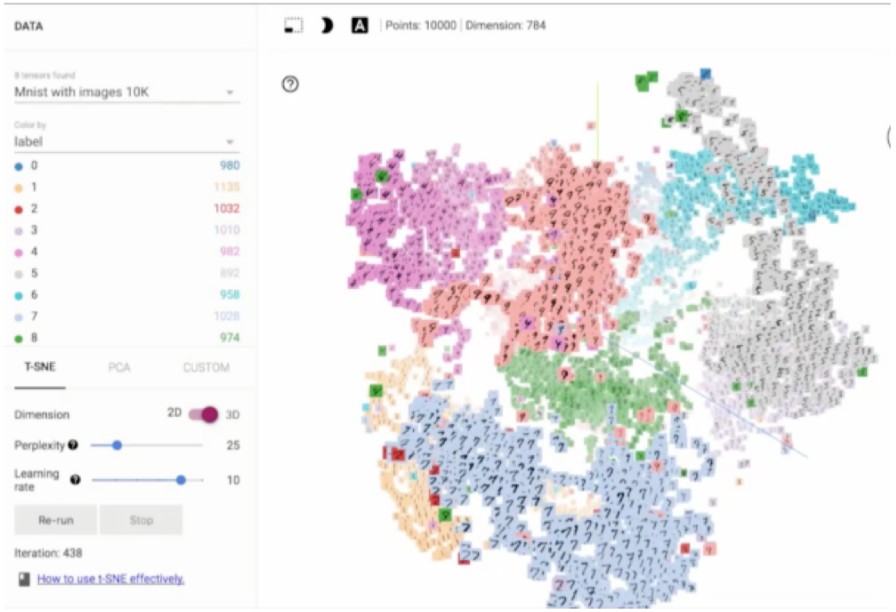
- Node Representation Methods
  - LINE, DeepWalk, node2vec
- **Graph and High-dimensional Data Visualization**
  - **LargeVis**
- Knowledge Graph Embedding
  - RotatE
- A High-performance Node Representation System

Extremely Low-dimensional Representations: 2D/3D for Visualizing Networks



t-SNE (Maarten and Hinton, 2008, 2014)

- State-of-the-art algorithm for high-dimensional data visualization
  - Deployed by Tensorflow
- Limitations
  - K-NNG construction: complexity grows **O(NlogN)** to the number of data points N
  - Graph layout: complexity is **O(NlogN)**
  - Very sensitive parameters



TensorBoard Visualizations by t-SNE

LargeVis (Tang et al., Best Paper Nomination at WWW 2016)

- Efficient approximation of K-NNG construction
  - **30** times faster than t-SNE (3 million data points)
  - Better time-accuracy tradeoff

- Efficient probabilistic model for graph layout
  - O(NlogN) -> O(N)
  - **7** times faster than t-SNE (3 million data points)
  - Better visualization layouts
  - Stable parameters across different data sets

Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. **Visualizing Large-scale and High-dimensional Data.** WWW'16

Learning the Layout of KNN Graph

- Preserve the similarities of the nodes in 2D/3D space
  - Represent each node with a 2D/3D vector
  - Keep similar data close while dissimilar data far apart
- Probability of observing a binary edge between nodes (i,j):

$$p(e_{ij}=1)=\frac{1}{1+\|\vec{y}_i-\vec{y}_j\|^2}$$

- Likelihood of observing a weighted edge between nodes (i,j):

$$p(e_{ij}=w_{ij})=p(e_{ij}=1)^{w_{ij}}$$

A Probabilistic Model for Graph Layout

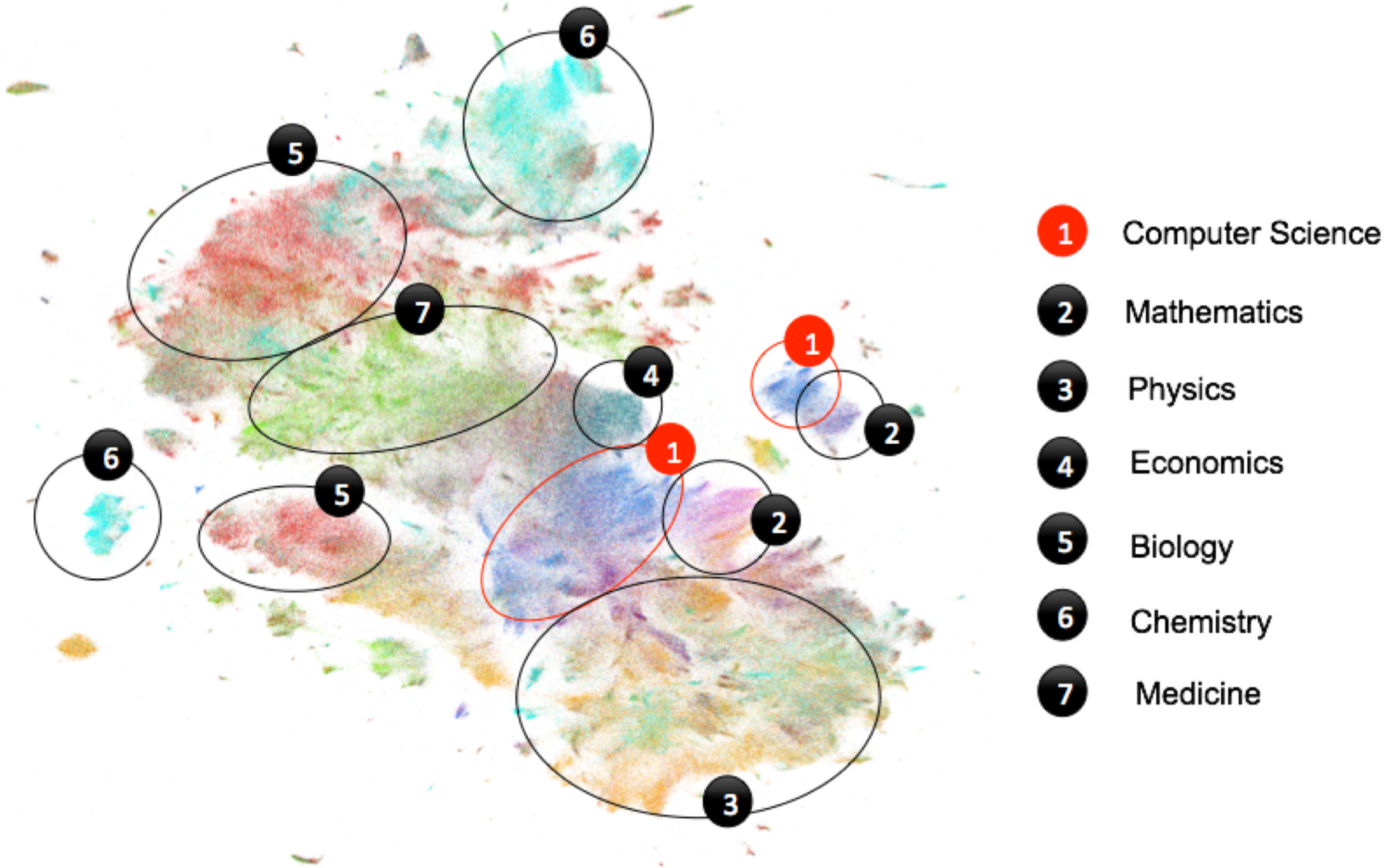
- Objective:
- $$O = \prod_{(i,j)\in E} p(e_{ij} = w_{ij}) \prod_{(i,j)\in \bar{E}} (1 - p(e_{ij} = w_{ij}))^{\gamma}$$

γ: an unified weight assigned to negative edge

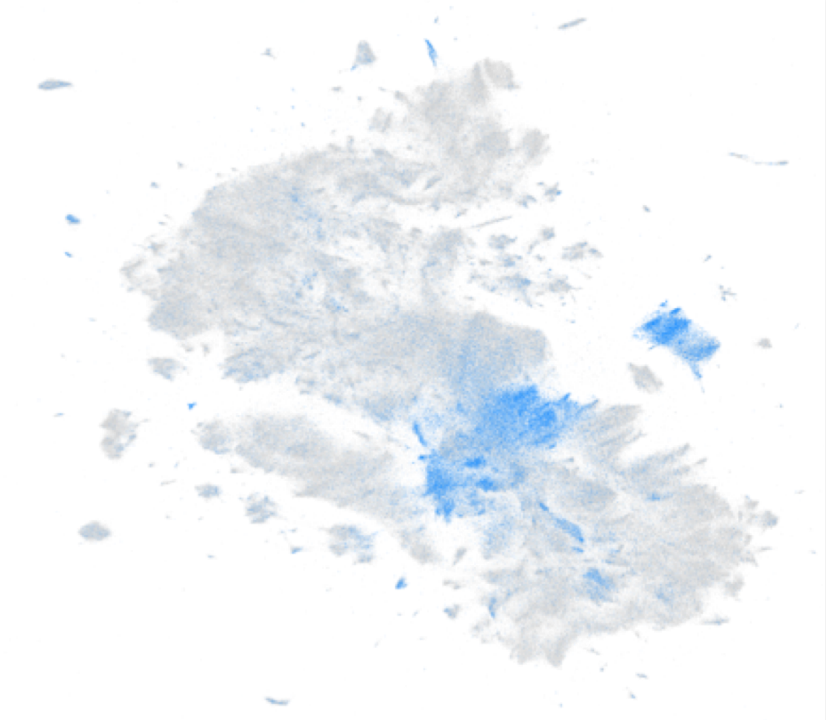
- Randomly sample some negative edges
- Optimized through asynchronous stochastic gradient descent
- Time complexity: linear to the number of data points

10M Scientific Papers on One Slide

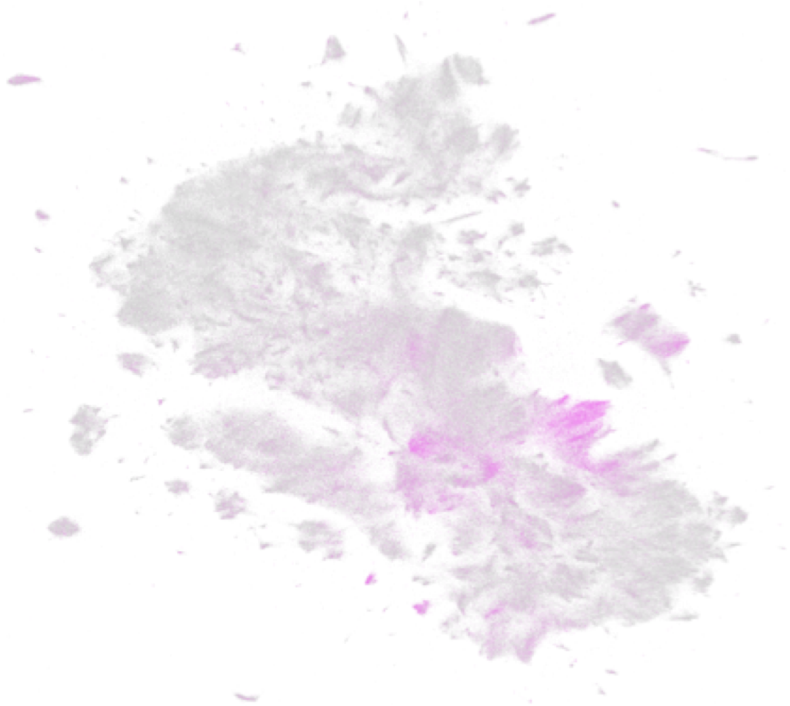
10M Scientific Papers on One Slide



Computer Science

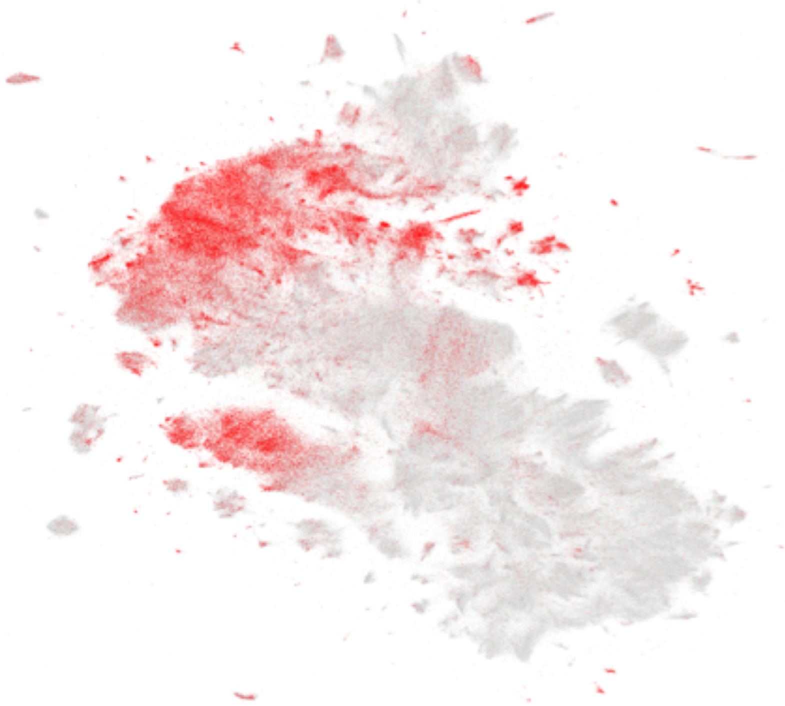


Mathematics

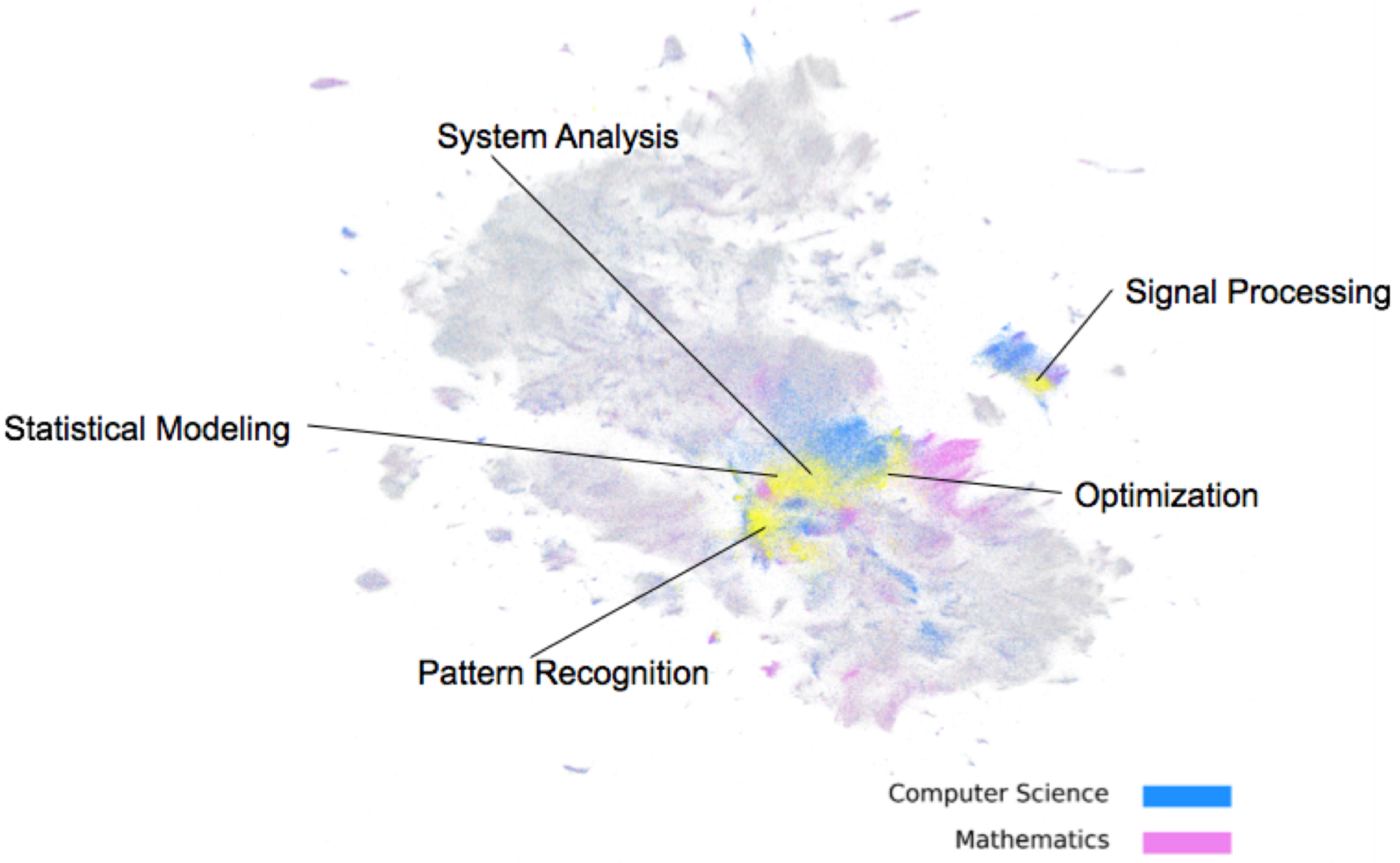


Physics

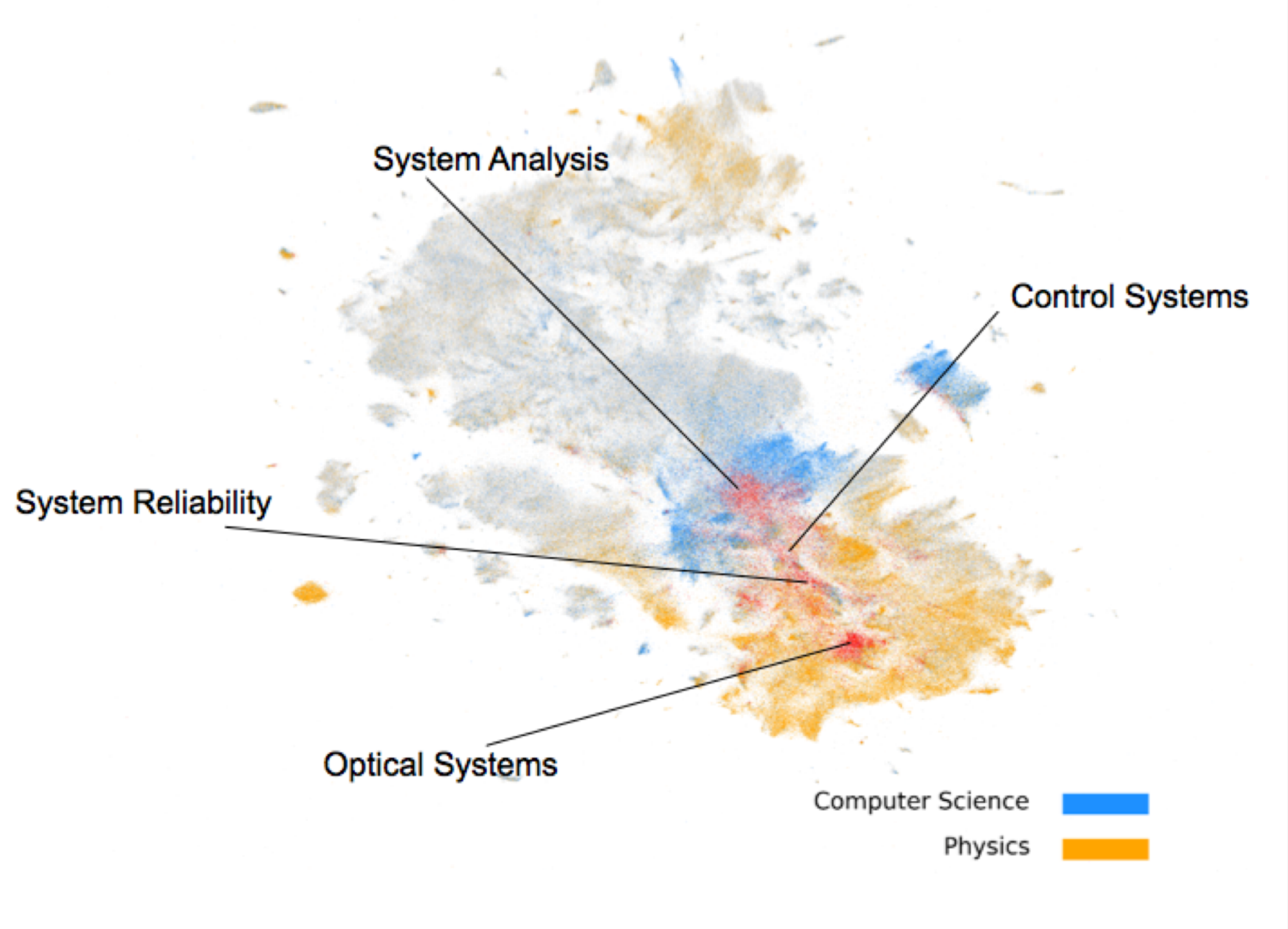
Biology

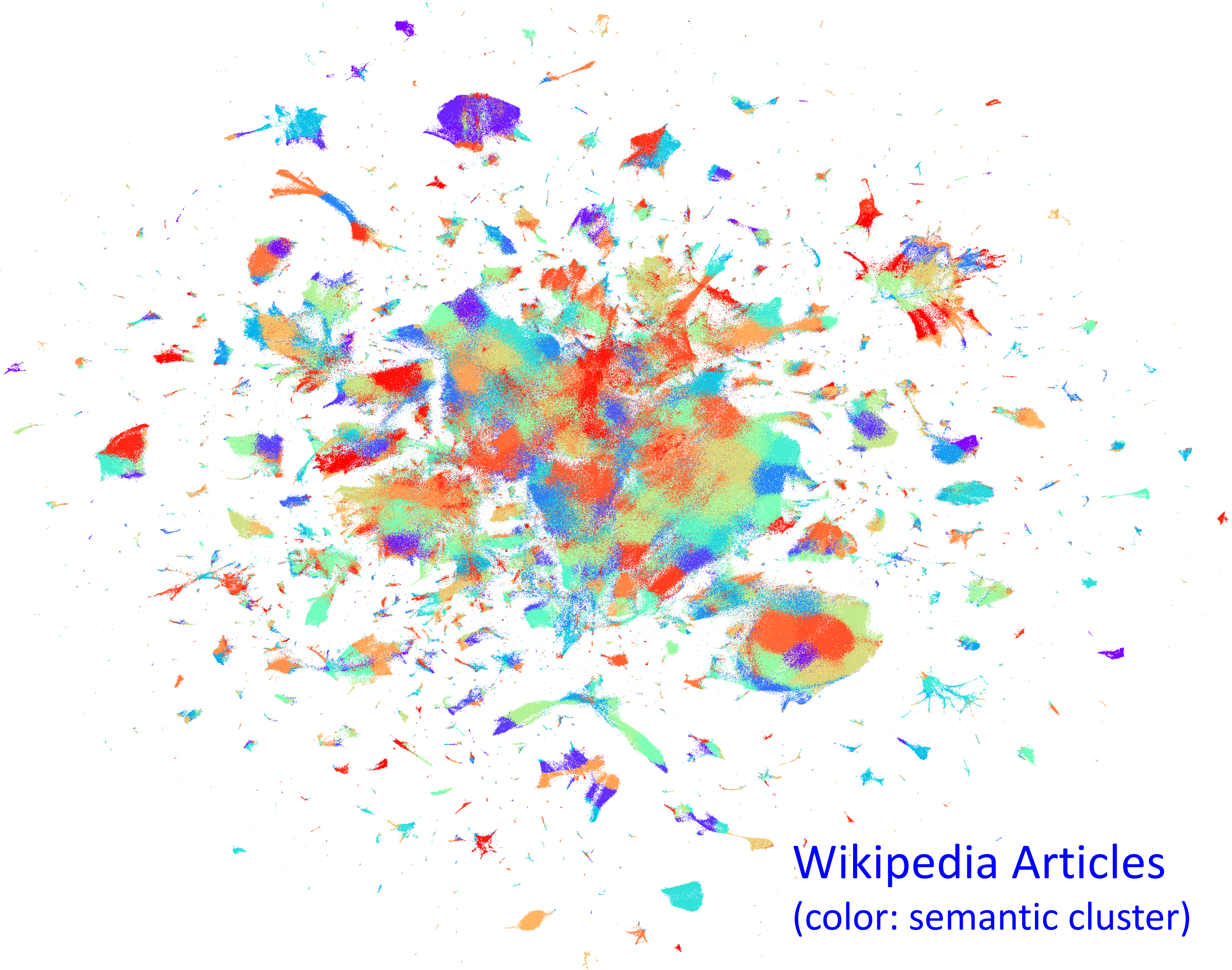


Computer Science vs. Mathematics

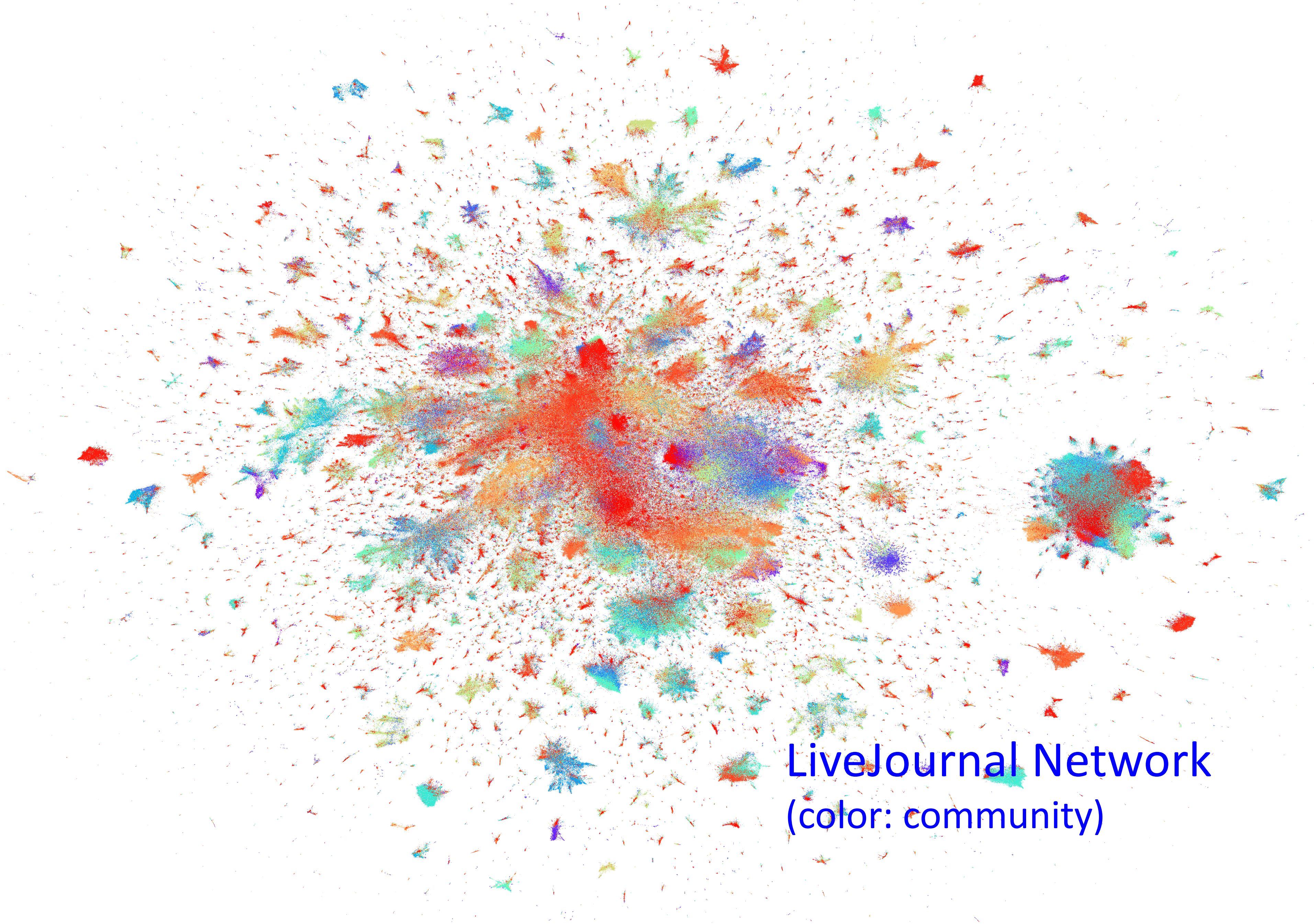


Computer Science vs. Physics

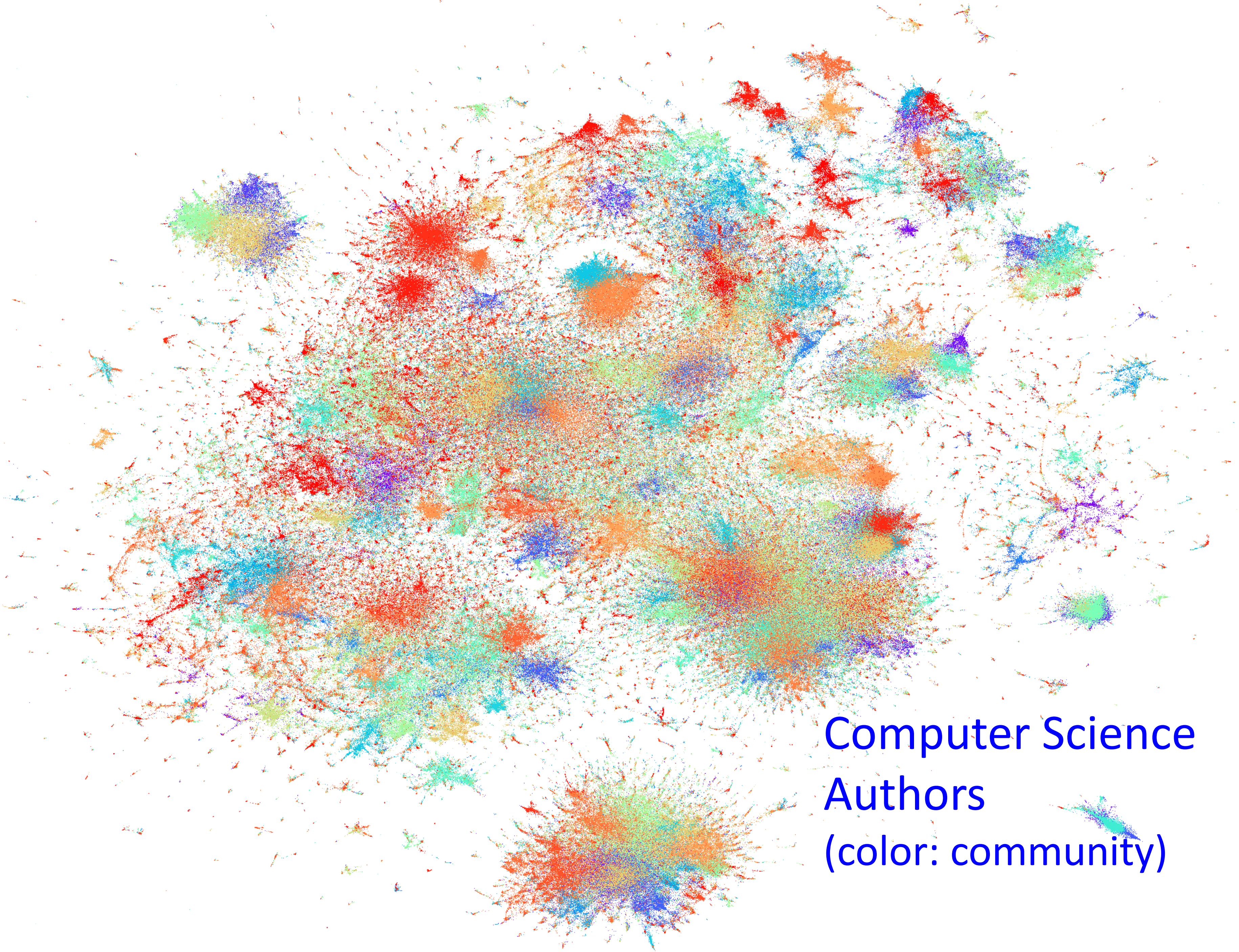




Wikipedia Articles
(color: semantic cluster)



LiveJournal Network
(color: community)

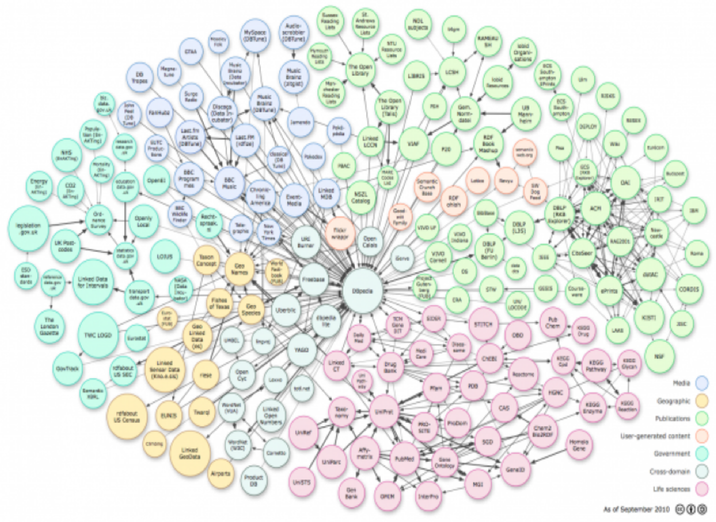


Computer Science Authors
(color: community)

Outline

- Node Representation Methods
  - LINE, DeepWalk, node2vec
- Graph and High-dimensional Data Visualization
  - LargeVis
- **Knowledge Graph Embedding**
  - **RotatE**
- A High-performance Node Representation System

Knowledge Graphs



- Knowledge graphs are **heterogeneous** graphs
  - Multiple types of relations
- A set of facts represented as triplets
  - (head entity, relation, tail entity)
- A variety of applications
  - Question answering
  - Search
  - Recommender Systems
  - Natural language understanding
  - …

Google Knowledge Graph

Microsoft SATORI

Browse the Knowledge Base!

NELL: Never-Ending Language Learning

Freebase

DBpedia

yago select knowledge

OpenIE (Reverb, OLLIE)

Related Work on Knowledge Graph Embedding

- Representing entities as **embeddings**
- Representing relations as **embeddings** or **matrices**

Model	Score Function	
SE (Bordes et al., 2011)	$-\|\boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t}\|$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \boldsymbol{W}_{r,\cdot} \in \mathbb{R}^{k \times k}$
TransE (Bordes et al., 2013)	$-\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
TransX	$-\|g_{r,1}(\mathbf{h}) + \mathbf{r} - g_{r,2}(\mathbf{t})\|$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
DistMult (Yang et al., 2014)	$\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\langle \mathbf{r}, \mathbf{h}, \overline{\mathbf{t}} \rangle)$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$
HolE (Nickel et al., 2016)	$\langle \mathbf{r}, \mathbf{h} \otimes \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ConvE (Dettmers et al., 2017)	$\langle \sigma(\mathrm{vec}(\sigma([\overline{\mathbf{r}}, \overline{\mathbf{h}}] * \boldsymbol{\Omega}))\boldsymbol{W}), \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
RotatE	$-\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|^1$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k, |r_i| = 1$

Task: Knowledge Graph Completion

- A fundamental task: **predicting missing links**
- The Key Idea: model and infer the **relation patterns** in knowledge graphs according to observed knowledge facts.
  - The relationship between relations
- Example:

Barack_Obama **BornIn** United_States

Barack_Obama **Nationality** American

Parents of Parents are Grandparents

Relation Patterns

- **Symmetric/Antisymmetric** Relations
  - Symmetric: e.g., Marriage
  - Antisymmetric: e.g., Filiation
- Formally:

r is **Symmetric**: $r(x, y) \Rightarrow r(y, x)$ if $\forall\ x, y$

r is **Antisymmetric**: $r(x, y) \Rightarrow \neg r(y, x)$ if $\forall\ x, y$

Relation Patterns

- **Inverse** Relations
  - Hypernym and hyponym
  - Husband and wife
- Formally:

r_1 **is inverse to relation** r_2**:** $r_2(x, y) \Rightarrow r_1(y, x)$ if $\forall\ x, y$

Relation Patterns

- **Composition** Relations
  - My mother's husband is my father
- Formally:

r_1 is a **composition** of relation r_2 and relation r_3**:** $r_2(x, y) \wedge r_3(y, z) \Rightarrow r_1(x, z)$ if $\forall\ x, y,$ z

Abilities in Inferring the Relation Patterns

- None of existing methods are able to model and infer all the three types of relation patterns

Model	Score Function	Symmetry	Antisymmetry	Inversion	Composition
SE	$-\|\boldsymbol{W}_{r,1}\mathbf{h} - \boldsymbol{W}_{r,2}\mathbf{t}\|$	✗	✗	✗	✗
TransE	$-\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|$	✗	✓	✓	✓
TransX	$-\|g_{r,1}(\mathbf{h})+\mathbf{r}-g_{r,2}(\mathbf{t})\|$	✓	✓	✗	✗
DistMult	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	✓	✗	✗	✗
ComplEx	$\mathrm{Re}(\langle \mathbf{h}, \mathbf{r}, \overline{\mathbf{t}} \rangle)$	✓	✓	✓	✗
RotatE	$-\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\|$	✓	✓	✓	✓

RotatE (Sun et al. 2019)

- A new knowledge graph embedding model RotatE
  - Each relation as a elementwise rotation from the source entity to the target entity in the complex vector space
- RotatE is able to model and infer all the three types of relation patterns
- An efficient and effective negative sampling algorithm for optimizing RotatE
- State-of-the-art results on all the benchmarks for link prediction on knowledge graphs

Zhiqing Sun, Zhihong Deng, Jian-Yun Nie, and **Jian Tang**. "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space." to appear in ICLR'19.

Relation as Elementwise Rotation in Complex Space

- Representing head and tail entities in complex vector space, i.e., $\mathbf{h}, \mathbf{t} \in \mathbb{C}^{\boldsymbol{k}}$
- Define each relation **r** as an element-wise rotation from the head entity **h** to the tail entity **t,** i.e.,

$$\mathbf{t} = \mathbf{h} \circ \boldsymbol{r}, \quad \text{where } |r_i|=1$$

- $\circ$ is the element-wise product. More specifically, we have $\mathrm{t_i} = \mathrm{h_i r_i}$, and

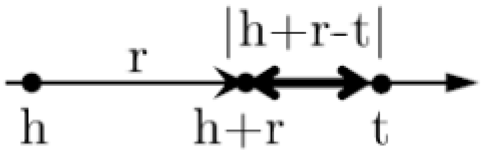
$$\mathrm{r_i} = e^{i\theta_{r,i}},$$

- where $\theta_{r,i}$ is the phase angle of **r** in the i-th dimension.

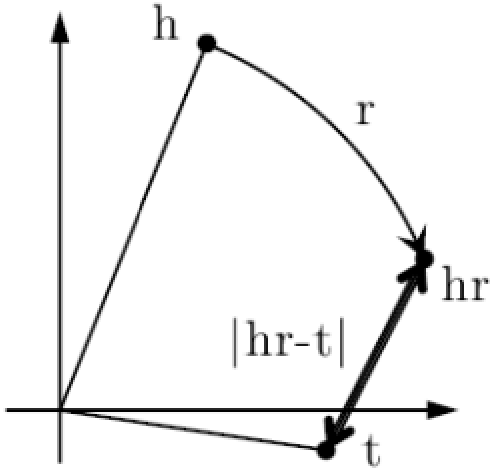
Geometric Interpretation

- Define the distance function of RotatE as

$$d_{\boldsymbol{r}}(\boldsymbol{h}, \boldsymbol{t}) = ||\mathbf{h} \circ \mathbf{r} - \mathbf{t}||$$



(a) TransE models $\mathbf{r}$ as translation in real line.



(b) RotatE models $\mathbf{r}$ as rotation in complex plane.

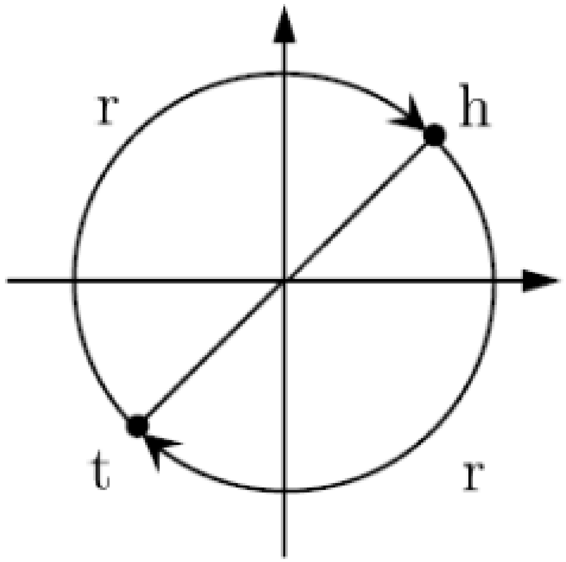
Modeling the Relation Patterns with RotatE

- A relation **r** is **symmetric** if and only if $r_i = \pm 1$, i.e.,

$$\theta_{r,i} = 0 \text{ or } \pi$$

- An example on the space of $\mathbb{C}$

$$r_i = -1 \text{ or } \theta_{r,i} = \pi$$



Modeling the Relation Patterns with RotatE

- A relation r is **antisymmetric** if and only if $\mathbf{r} \circ \mathbf{r} \neq \mathbf{1}$

- Two relations r_1 and r_2 are **inverse** if and only if $\mathbf{r}_2 = \bar{\mathbf{r}}_1$, i.e.,

$$\theta_{2,i} = -\theta_{1,i}$$

- A relation $\boldsymbol{r_3} = e^{i\boldsymbol{\theta_3}}$ is a **composition** of two relations $\boldsymbol{r_1} = e^{i\boldsymbol{\theta_1}}$ and $\boldsymbol{r_2} = e^{i\boldsymbol{\theta_2}}$ if only if $\boldsymbol{r_3} = \boldsymbol{r_1} \circ \boldsymbol{r_2}$, i.e.,

$$\boldsymbol{\theta_3} = \boldsymbol{\theta_1} + \boldsymbol{\theta_2}$$

Optimization

- Negative sampling loss

$$L = -\log \sigma\big(\gamma - d_r(\boldsymbol{h}, \boldsymbol{t})\big) - \sum_{i=1}^{k} \frac{1}{k} \log \sigma(d_r(\boldsymbol{h}_i', \boldsymbol{t}_i') - \gamma)$$

- γ is a fixed margin, σ is the sigmoid function, and $(\boldsymbol{h}_i', \boldsymbol{r}, \boldsymbol{t}_i')$ is the i-th negative triplet.

Self-adversarial Negative Sampling

- Traditionally, the negative samples are drawn in an uniform way
  - Inefficient as training goes on since many samples are obviously false
  - Does not provide useful information
- A self-adversarial negative sampling
  - Sample negative triplets according to the current embedding model
  - Starts from easier samples to more and more difficult samples
  - Curriculum Learning

$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(\mathbf{h}'_j, \mathbf{t}'_j)}{\sum_i \exp \alpha f_r(\mathbf{h}'_i, \mathbf{t}'_i)}$$

- α is the temperature of sampling. $f_r(h'_j, t'_j)$ measures the salience of the triplet

The Final Objective

- Instead of sampling, treating the sampling probabilities as weights.

$$L = -\log \sigma(\gamma - d_r(\mathbf{h}, \mathbf{t})) - \sum_{i=1}^{n} p(h'_i, r, t'_i) \log \sigma(d_r(\mathbf{h}'_i, \mathbf{t}'_i) - \gamma)$$

Experiments: Data Sets

- **FB15K**: a subset of Freebase. The main relation types are **symmetry/antisymmetry** and **inversion** patterns.
- **WN18**: a subset of WordNet. The main relation types are **symmetry/antisymmetry** and **inversion** patterns.
- **FB15K-237**: a subset of FB15K, where inversion relations are deleted. The main relation types are **symmetry/antisymmetry** and **composition** patterns.
- **WN18RR**: a subset of WN18, where inversion relations are deleted. The main relation types are **symmetry/antisymmetry** and **composition** patterns.

Dataset	#entity	#relation	#training	#validation	#test
FB15k	14,951	1,345	483,142	50,000	59,071
WN18	40,943	18	141,442	5,000	5,000
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134

Results on FB15k and WN18

- RotatE performs the best
- pRotatE performs similarly to RotatE

	FB15k					WN18				
	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE [♥]	-	.463	.297	.578	.749	-	.495	.113	.888	.943
DistMult [♦]	42	.798	-	-	**.893**	655	.797	-	-	.946
HolE	-	.524	.402	.613	.739	-	.938	.930	.945	.949
ComplEx	-	.692	.599	.759	.840	-	.941	.936	.945	.947
ConvE	51	.657	.558	.723	.831	374	.943	.935	.946	.956
pRotatE	43	**.799**	**.750**	.829	.884	**254**	.947	.942	.950	.957
RotatE	**40**	.797	.746	**.830**	.884	309	**.949**	**.944**	**.952**	**.959**

Results on FB15k-237 and WN18RR

- RotatE performs the best
- RotatE performs significantly better than pRotatE
  - A lot of composition patterns on the two data sets
  - Modulus information are important for modeling the composition patterns

	FB15k-237					WN18RR				
	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE [♥]	357	.294	-	-	.465	3384	.226	-	-	.501
DistMult	254	.241	.155	.263	.419	5110	.43	.39	.44	.49
ComplEx	339	.247	.158	.275	.428	5261	.44	.41	.46	.51
ConvE	244	.325	.237	.356	.501	4187	.43	.40	.44	.52
pRotatE	178	.328	.230	.365	.524	**2923**	.462	.417	.479	.552
RotatE	**177**	**.338**	**.241**	**.375**	**.533**	3340	**.476**	**.428**	**.492**	**.571**

Results on Countries (Bouchard et al. 2015)

- A carefully designed dataset to explicitly test the capabilities for modeling the composition patterns
  - Three subtasks S1, S2, S3
  - From easy to difficult

	Countries (AUC-PR)			
	DistMult	ComplEx	ConvE	RotatE
S1	$\mathbf{1.00 \pm 0.00}$	0.97 ± 0.02	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
S2	0.72 ± 0.12	0.57 ± 0.10	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$
S3	0.52 ± 0.07	0.43 ± 0.07	0.86 ± 0.05	$\mathbf{0.95 \pm 0.00}$

Summary

- Modeling relation patterns is critical for knowledge base completion
  - Symmetric/Antisymmetric, Inverse, and composition
- RotatE: define each relation as a **elementwise rotation** from the head entity to the tail entity in the complex vector space
  - Capable of modeling and inferring all the three types of relation patterns
- A self-negative sampling techniques for training knowledge graph embeddings
- State-of-the-art results on all existing benchmark data sets

Software

LINE: **(C++)**	https://github.com/tangjianpku/LINE (593 stars, released since 2015.3)
LargeVis : **(C++&Python)**	https://github.com/lferry007/LargeVis (459 stars, released since 2016.7)
RotatE : **(Pytorch)**	https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding (just released!!)

Outline

- Node Representation Methods
  - LINE, DeepWalk, node2vec
- Graph and High-dimensional Data Visualization
  - LargeVis
- Knowledge Graph Embedding
  - RotatE
- **A High-performance Node Representation System**

A High-Performance CPU-GPU Hybrid System for Node Embedding (Zhu et al. 2019)

- A specific system designed for node embeddings through algorithm and system co-design
  - CPUs: online random walk generation
  - GPUs: training node embeddings
  - Efficient and effective collaboration strategies between CPUs and GPUs
- 50 times faster than existing systems
- Take only **one minute** for a network with one million node

Zhaocheng Zhu, Shizhen Xu, Meng Qu, and **Jian Tang**. "**A High-Performance CPU-GPU Hybrid System for Node Embedding** ". To appear in WWW'19.

Summary

- Node Representation Methods
  - LINE, DeepWalk, node2vec
- Graph and High-dimensional Data Visualization
  - LargeVis
- Knowledge Graph Embedding
  - RotatE
- A High-performance Node Representation System

Thanks!

Contact: jian.tang@hec.ca