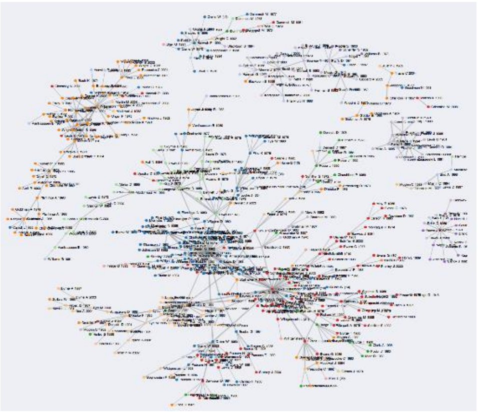
Large-scale Network Embedding &Visualization

Jian Tang

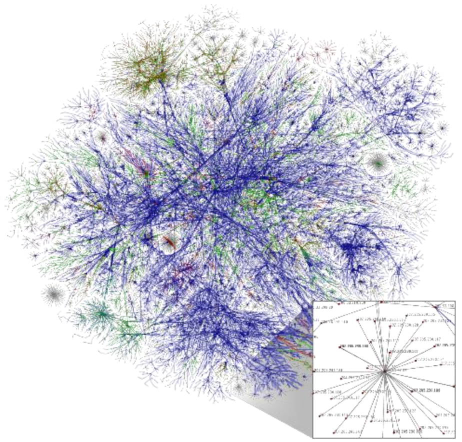
Microsoft Research

Ubiquitous Large-scale Information Networks

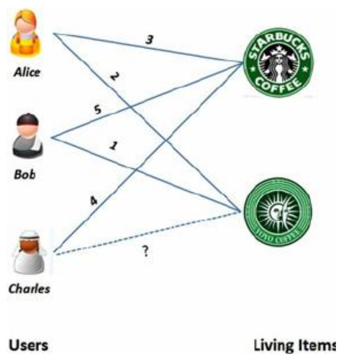
Social network



Citation network



World Wide Web



User-item Network

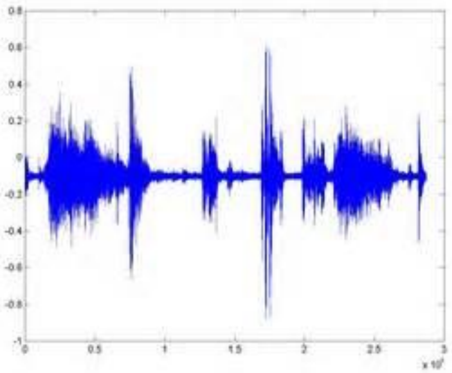
…

- Real-world networks are very ***large***, e.g.,
  - Facebook social network: ~ 1 billion users
  - WWW: ~50 billion webpages
  - User-item network: >300M users, ~200M items(Amazon)
- Very challenging in analyzing large-scale networks
  - Sparse
  - High-dimension

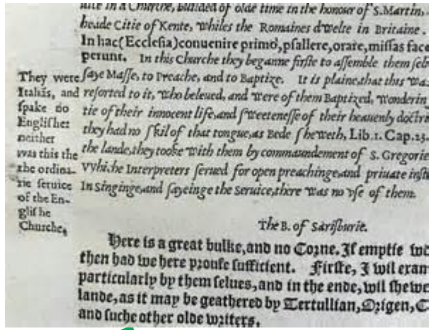
Various Applications

- Recommendation
  - User-item Network in Amazon, Netflix etc.
  - Academic Network: recommend similar authors, papers and venues
- Ranking
  - Query-Page Network: page ranking
- Classification/Clustering
  - Query/page classification or clustering
- …
- All require a good **representation** of network nodes
  - Compute pairwise ***similarities*** for ranking, recommendation
  - As ***features*** for classification or clustering

Representation Learning/ Deep Learning for Different Modalities



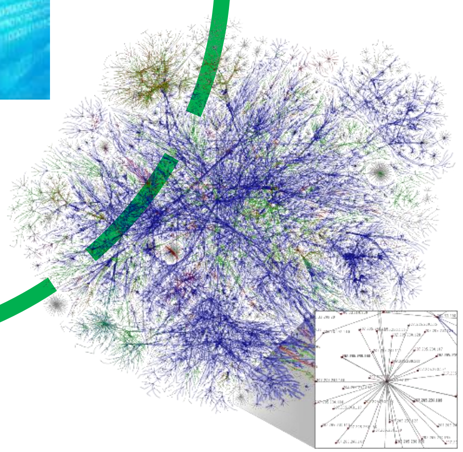
Speech



Natural language

Deep Learning

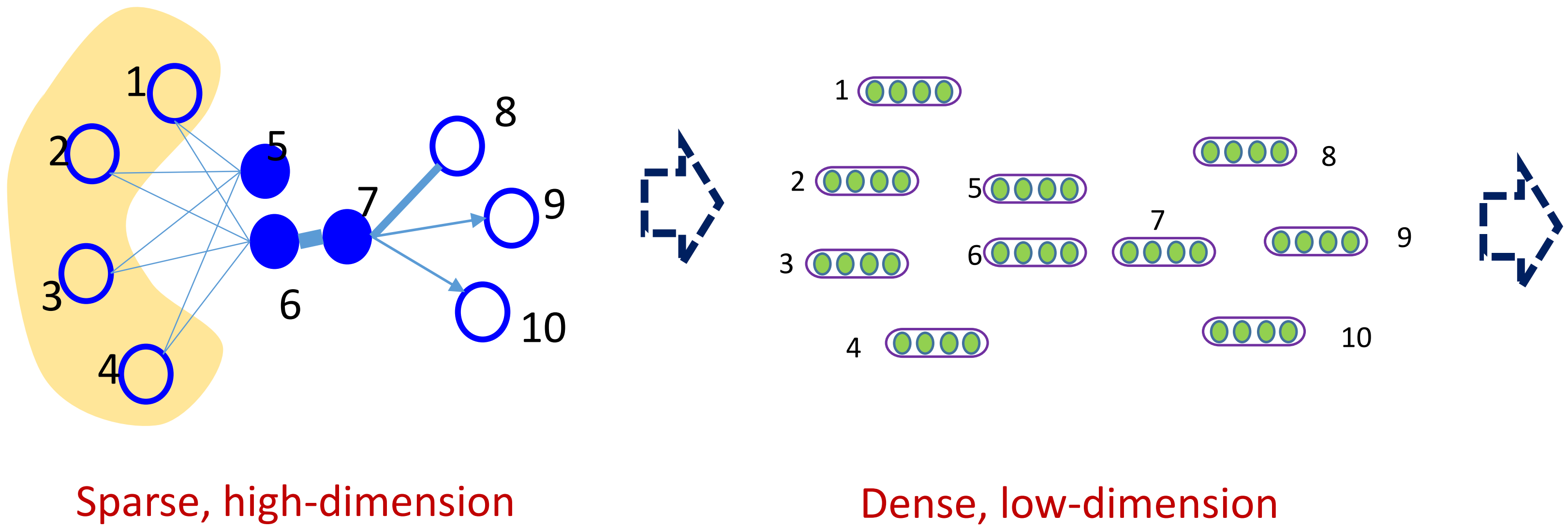
Image



?

Network

Learning Node Representations of Networks

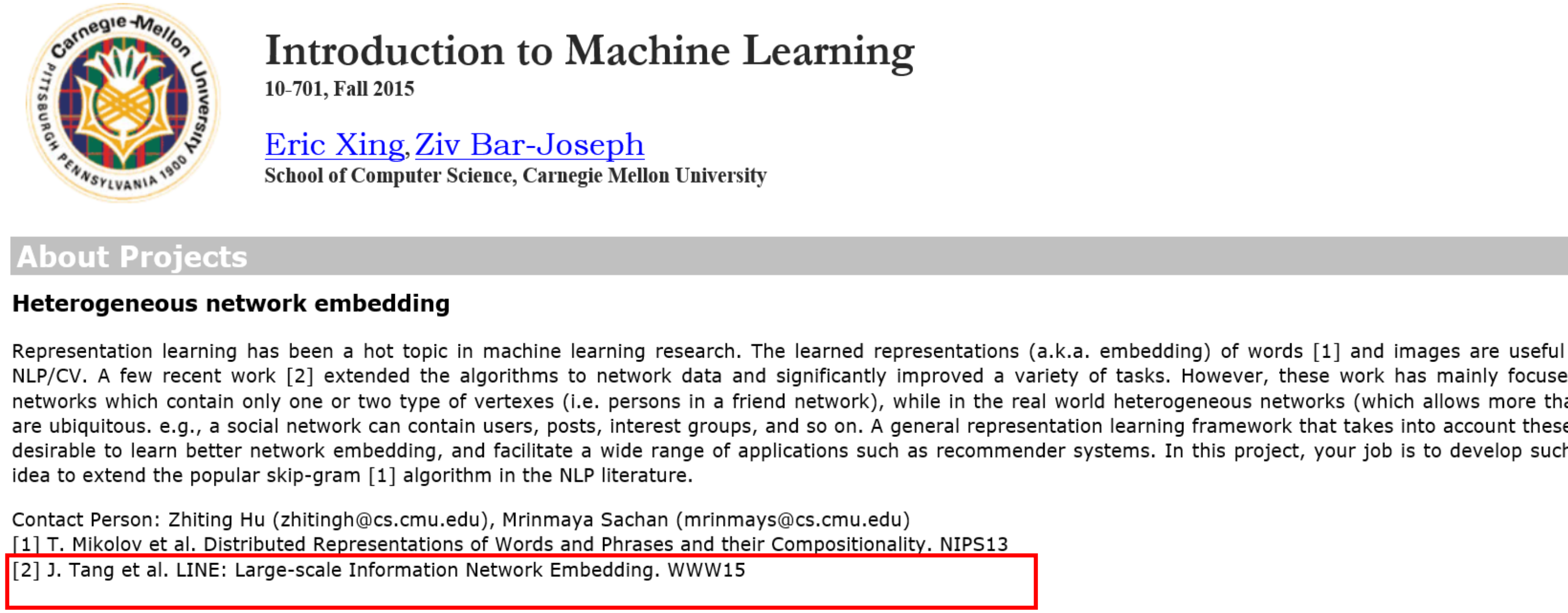


- Network visualization
- Node classification
- Link prediction
- Recommendation
- …

Large-scale Network Representation and Visualization

- LINE: Large-scale Network Embedding/Representation (WWW’15)
- Social network, academic network representation
- A benchmark model in this area

Large-scale Network Representation and Visualization



Introduction to Machine Learning

10-701, Fall 2015

Eric Xing, Ziv Bar-Joseph

School of Computer Science, Carnegie Mellon University

About Projects

Heterogeneous network embedding

Representation learning has been a hot topic in machine learning research. The learned representations (a.k.a. embedding) of words [1] and images are useful NLP/CV. A few recent work [2] extended the algorithms to network data and significantly improved a variety of tasks. However, these work has mainly focuse networks which contain only one or two type of vertexes (i.e. persons in a friend network), while in the real world heterogeneous networks (which allows more tha are ubiquitous. e.g., a social network can contain users, posts, interest groups, and so on. A general representation learning framework that takes into account these desirable to learn better network embedding, and facilitate a wide range of applications such as recommender systems. In this project, your job is to develop such idea to extend the popular skip-gram [1] algorithm in the NLP literature.

Contact Person: Zhiting Hu (zhitingh@cs.cmu.edu), Mrinmaya Sachan (mrinmays@cs.cmu.edu)

[1] T. Mikolov et al. Distributed Representations of Words and Phrases and their Compositionality. NIPS13

[2] J. Tang et al. LINE: Large-scale Information Network Embedding. WWW15

- LINE: Large-scale Network Embedding/Representation (WWW’15)
- Social network, academic network representation
- A benchmark model in this area

Large-scale Network Representation and Visualization

- Network Representation for text representation (KDD’15)
- State-of-the-art results for word and document embedding
- LINE > Word2VEC, LINE> ParagraphVEC

- LINE: Large-scale Network Embedding/Representation (WWW’15)
- Social network, academic network representation
- A benchmark model in this area

Large-scale Network Representation and Visua

Text representation, e.g., word and document representation, …

Deep learning has been attracting increasing attention …

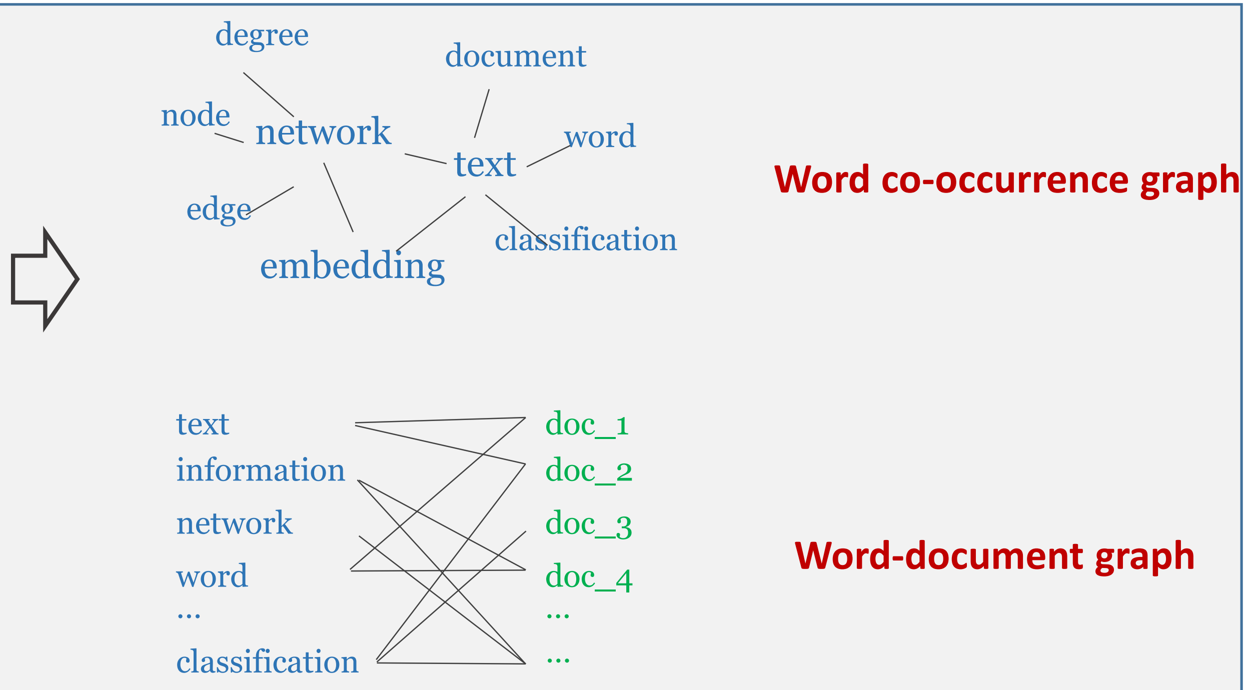
A future direction of deep learning is to integrate unlabeled data …

…

The Skip-gram model is quite effective and efficient …

Information networks encode the relationships between the data objects …

Free text



- Network Representation for text representation (KDD'15)
- State-of-the-art results for word and document embedding
- LINE > Word2VEC, LINE> ParagraphVEC

- LINE: Large-scale Network Embedding/Representation (WWW'15)
- Social network, academic network representation
- A benchmark model in this area

Large-scale Network Representation and Visualization

- LargeVis: Big Network/Data Visualization (WWW'16)
- Best paper nominee 5/727

- Network Representation for text representation (KDD'15)
- State-of-the-art results for word and document embedding
- LINE > Word2VEC, LINE> ParagraphVEC

- LINE: Large-scale Network Embedding/Representation (WWW'15)
- Social network, academic network representation
- A benchmark model in this area

Large-scale Network Representation and Visualization

Scientific Literature

- LargeVis: Big Network/Data Visualiza
- Best paper nominee 5/727

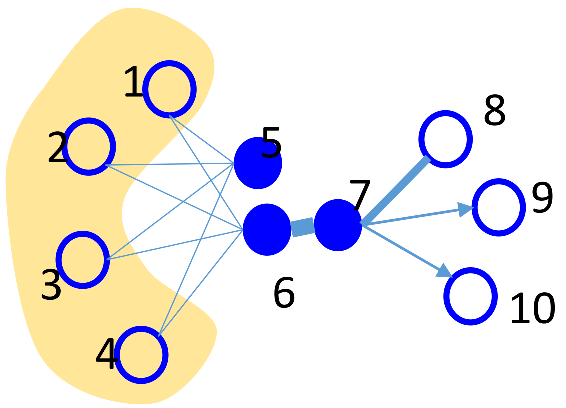
- Network Representation for text representation (KDD’
- State-of-the-art results for word and document embed
- LINE > Word2VEC, LINE> ParagraphVEC

- LINE: Large-scale Netw
- Social network, acade
- A benchmark model i

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 embedding
  - Usually for undirected graphs
- DeepWalk (Perozzi et al. 2014)
  - Lack a clear objective function
  - Designed for networks with binary edges

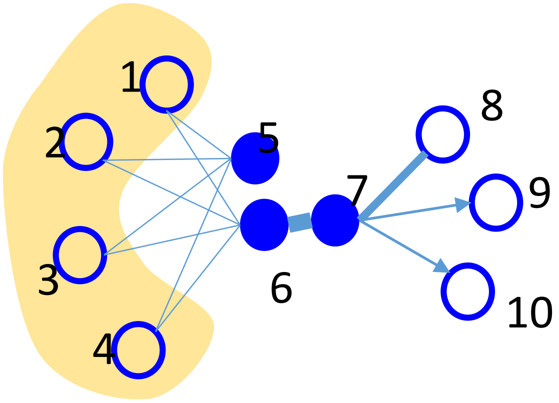
First-order Proximity



Vertex **6** and **7** have a large first-order proximity

- The local pairwise proximity between the vertices
  - Determined by the observed links
- However, many links between the vertices are missing
  - Not sufficient for preserving the entire network structure

Second-order Proximity



Vertex **5** and **6** have a large second-order proximity

$$\hat{p}_5 = (1,1,1,1,0,0,0,0,0,0)$$

$$\hat{p}_6 = (1,1,1,1,0,0,5,0,0,0)$$

- The proximity between the *neighborhood structures* of the vertices
- Mathematically, the second-order proximity between each pair of vertices (u,v) is determined by:

$$\hat{p}_u = (w_{u1}, w_{u2}, \dots, w_{u|V|})$$

$$\hat{p}_v = (w_{v1}, w_{v2}, \dots, w_{v|V|})$$

"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

Preserving the First-order Proximity

- Given an ***undirected*** edge (v_i, v_j), the joint probability of v_i, v_j

$$p_1(v_i, v_j) = \frac{\exp(\vec{u}_j^T \cdot \vec{u}_i)}{\sum_{m,n=1}^{|V|} \exp(\vec{u}_m^T \cdot \vec{u}_n)}$$

$\vec{u}_i$: Embedding of vertex v_i

$$\hat{p}_1(v_i, v_j) = \frac{w_{ij}}{\sum_{(i',j')} w_{i'j'}}$$

- Objective:

$$O_1 = d(\hat{p}_1(\cdot,\cdot), p_1(\cdot,\cdot))$$

KL-divergence

$$\propto - \sum_{(i,j) \in E} w_{ij} \log p_1(v_i, v_j)$$

Preserving the Second-order Proximity

- Given a ***directed*** edge (v_i, v_j), the conditional probability of v_j given v_i is:

$$p_2(v_j|v_i) = \frac{\exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}_k'^T \cdot \vec{u}_i)}$$

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

$\vec{u}_i$: Embedding of vertex *i* when *i* is a source node;
$\vec{u}_i'$: Embedding of vertex *i* when *i* is a target node.

- Objective:

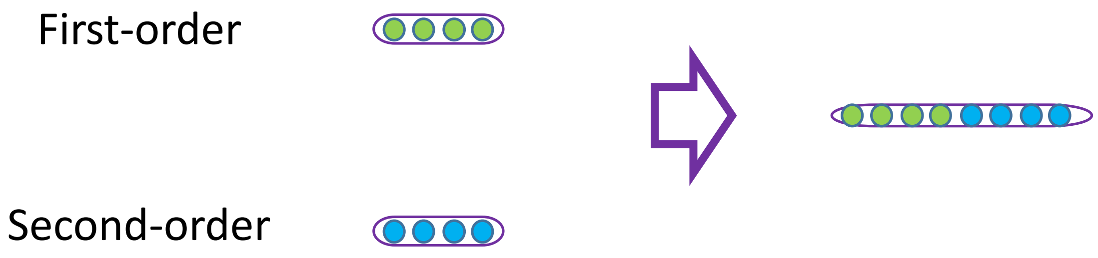
$$O_2 = \sum_{i \in V} \lambda_i d(\hat{p}_2(\cdot|v_i), p_2(\cdot|v_i))$$

$$\propto -\sum_{(i,j) \in E} w_{ij} \log p_2(v_j|v_i)$$

λ_i: Prestige of vertex in the network
$\lambda_i = \sum_j w_{ij}$

Preserving both Proximity

- Concatenate the embeddings individually learned by the two proximity



Optimization

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

$$\frac{\partial O_2}{\partial \vec{u}_i} = w_{ij} \cdot \frac{\partial \log p_2(v_j|v_i)}{\partial \vec{u}_i}$$

Multiplied by the weight of the edge w_{ij}

- Problematic when the weights of the edges diverge
  - The scale of the gradients with different edges diverges
- Solution: **edge sampling**
  - Sample the edges according to their weights and treat the edges as binary
- Complexity: $O(dK|E|)$
  - Linear to the dimension *d*, the number of negative samples *K*, and the number of edges $|E|$

Social Network Embedding

Node Classification

- Node classification
  - Community as the ground truth

Algorithm	10%	20%	30%	40%	50%	60%	70%	80%	90%
GF	53.23	53.68	53.98	54.14	54.32	54.38	54.43	54.50	54.48
DeepWalk	60.38	60.77	60.90	61.05	61.13	61.18	61.19	61.29	61.22
DeepWalk(256dim)	60.41	61.09	61.35	61.52	61.69	61.76	61.80	61.91	61.83
LINE(1^{st})	63.27	63.69	63.82	63.92	63.96	64.03	64.06	64.17	64.10
LINE(2^{nd})	62.83	63.24	63.34	63.44	63.55	63.55	63.59	63.66	63.69
LINE(1^{st}+2nd)	**63.20****	**63.97****	**64.25****	**64.39****	**64.53****	**64.55****	**64.61****	**64.75****	**64.74****

LINE(1^{st}+2^{nd})>LINE(1^{st})>LINE(2^{nd})>DeepWalk>GF

Unsupervised Text Embedding via Network Embedding

Modeling Word Co-occurrences with **Networks**

Text representation, e.g., word and document representation, …

Deep learning has been attracting increasing attention …

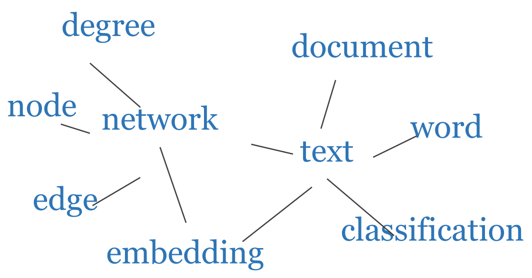
A future direction of deep learning is to integrate unlabeled data …

…

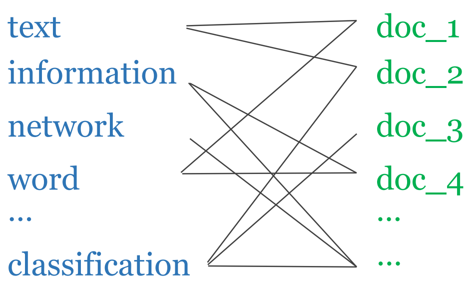
The Skip-gram model is quite effective and efficient …

Information networks encode the relationships between the data objects …

Unstructured text



Word co-occurrence network



Word-document network

Advantages of Networks over Free Texts

- Capture global structure of word co-occurrences
- Size of word co-occurrence networks does not grow linearly with data size
  - Only the weights of edges change

Text representation, e.g., word and document representation, …

Deep learning has been attracting increasing attention …

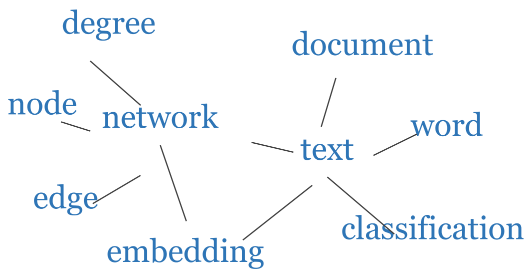
A future direction of deep learning is to integrate unlabeled data …

…

The Skip-gram model is quite effective and efficient …

Information networks encode the relationships between the data objects …

Unstructured text



Word co-occurrence network

Word Analogy

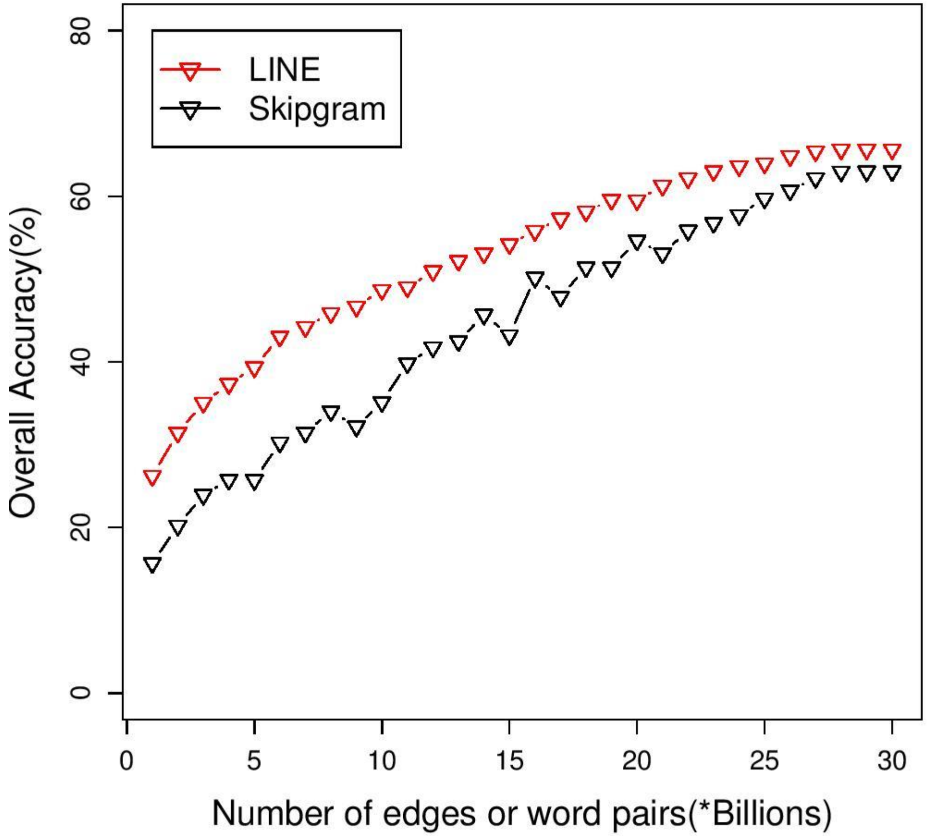
- Entire Wikipedia articles => word co-occurrence network (~2M words, 1B edges)

Algorithm	Semantic(%)	Syntactic(%)	Overall	Running time
GF	61.38	44.08	51.93	2.96h
Skip-gram	69.14	57.94	63.02	2.82h
LINE(1^{st})	58.08	49.42	53.35	**2.44h**
LINE(2^{nd})	**73.79**	**59.72**	**66.10**	2.55h

Effectiveness: **LINE(2^{nd}) >LINE(1^{st})**
LINE(2^{nd}) >Skipgram!!

Efficiency: **LINE(1st)>LINE(2^{nd})>Skipgram**

Performance w.r.t Number of Samples



Examples of nearest words

Word	Proximity Type	Top Similar Words
good	1st	luck, bad, faith, assume, nice
	2nd	decent, bad, excellent, lousy, reasonable
Information	1st	provide, provides, detailed, facts, verifiable
	2nd	information, ifnormaiton, informations, nonspammy, animecons
graph	1st	graphs, algebraic, finite, symmetric, topology
	2nd	graphs, subgraph, matroid, hypergraph, undirected
learn	1st	teach, learned, inform, educate, how
	2nd	learned, teach, relearn, learnt, understand

Text classification on long documents

- Word co-occurrence network **(w-w)** , word-document network **(w-d)** to learn the word embedding
- Document embedding as the average word embeddings in the document
- 20 newsgroup, Wikipedia article, IMDB

		20NG		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Unsupervised Embedding	Skipgram	70.62	68.99	75.80	75.77	85.34	85.34
	PV	75.13	73.48	76.68	76.75	86.76	86.76
	LINE(w-w)	72.78	70.95	77.72	77.72	86.16	86.16
	LINE(w-d)	**79.73**	**78.40**	**80.14**	**80.13**	**89.14**	**89.14**
	LINE (w-w +w-d)	78.74	77.39	79.91	79.94	89.07	89.07

LINE(w-w) > Skipgram

LINE(w-d) > ParagraphVEC

LINE(w-d) > LINE(w-w)

Text classification on short documents

- Word co-occurrence network **(w-w)** , word-document network **(w-d)** to learn the word embedding
- Document embedding as the average word embeddings in the document
- DBLP paper title (DBLP), movie review (MR), Tweets (Twitter)

		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Unsupervised Embedding	SkipGram	73.08	68.92	67.05	67.05	73.02	73.00
	PV	67.19	62.46	67.78	67.78	71.29	71.18
	LINE(w-w)	73.98	69.92	71.07	71.06	73.19	73.18
	LINE(w-d)	71.50	67.23	69.25	69.24	73.19	73.19
	LINE (w-w +w-d)	**74.22**	**70.12**	**71.13**	**71.12**	**73.84**	**73.84**

LINE(w-w) > Skipgram

LINE(w-d) > ParagraphVEC

LINE(w-w) > LINE(w-d)

Supervised/Semi-Supervised Text Embedding via Network Embedding

Unsupervised v.s. Supervised Text Embedding

- Unsupervised text embedding
  - Skip-gram (Mikolov et al. 2013)
  - Paragraph vector (Le et al. 2014)
  - LINE
- Supervised text embedding
  - Recurrent neural networks (Mikolov et al. 2010)
  - Recursive neural networks (Socher et al. 2012)
  - Convolutional neural network (Kim et al. 2014)

- Cons
  - Fully unsupervised, not tuned for specific tasks
- Pros
  - Simple model, scalable
  - Leverage a large amount of unlabeled data, embeddings are general for different tasks
  - Insensitive parameters

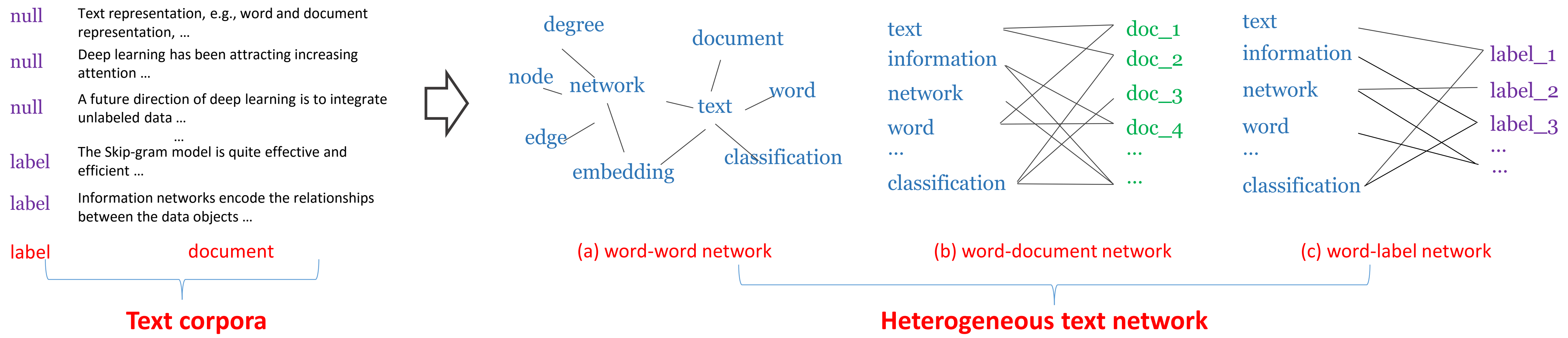
Skip-gram

- Pros
  - State-of-the art performance on specific tasks
- Cons
  - Computationally expensive
  - Require a large number of labeled data, hard to leverage unlabeled data
  - Very sensitive parameters, difficult to tune

Convolutional neural network

Predictive Text Embedding (PTE) with Heterogeneous Text Networks

- Adapt the advantages of unsupervised text embedding approaches but naturally utilize the ***labeled*** data for specific tasks
- How to represent unsupervised and supervised information?
  - Heterogeneous text network
- Different levels of word co-occurrences*: local context-level, document-level, label-level*



Heterogeneous Text Network Embedding

- Heterogeneous text network: three bipartite networks
  - Word-word (word-context), word-document, word-label network
  - Jointly embed the three bipartite networks
- Objective

$$O_{pte} = O_{ww} + O_{wd} + O_{wl}$$

- where

$$O_{ww} = -\sum_{(i,j)\in E_{ww}} w_{ij} \log p(v_i|v_j)$$

Objective for **word-word** network

$$O_{wd} = -\sum_{(i,j)\in E_{wd}} w_{ij} \log p(v_i|d_j)$$

Objective for **word-document** network

$$O_{wl} = -\sum_{(i,j)\in E_{wl}} w_{ij} \log p(v_i|l_j)$$

Objective for **word-label** network

Optimization

- Two different ways of optimization
  - Depends on when the labels kick in
- Joint training
  - Jointly train the three networks
- Pre-training + Fine-tuning
  - Jointly train the word-word and word-document networks
  - Fine-tuning the word embeddings with the word-label network

Results on **Long** Documents

With pretrained word embeddings

Pretrain with word-word, word-document network and then **fine-tune** with word-label network

Jointly train the word-word, word-document, word-label networks

		20newsgroup		Wikipedia		IMDB	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Unsupervised	LINE(G_{wd})	79.73	78.40	80.14	80.13	89.14	89.14
Predictive embedding	CNN	78.85	78.29	79.72	79.77	86.15	86.15
	CNN(pretrain)	80.15	79.43	79.25	79.32	89.00	89.00
	PTE(G_{wl})	82.70	81.97	79.00	79.02	85.98	85.98
	PTE($G_{ww} + G_{wl}$)	83.90	83.11	81.65	81.62	89.14	89.14
	PTE($G_{wd} + G_{wl}$)	**84.39**	**83.64**	82.29	82.27	89.76	89.76
	PTE(pretrain)	82.86	82.12	79.18	79.21	86.28	86.28
	PTE(joint)	84.20	83.39	**82.51**	**82.49**	**89.80**	**89.80**

PTE(joint) > PTE(pretrain)

PTE(joint) > PTE(G_{wl})

PTE(joint) > CNN/CNN(pretrain)

Results on **Short** Documents

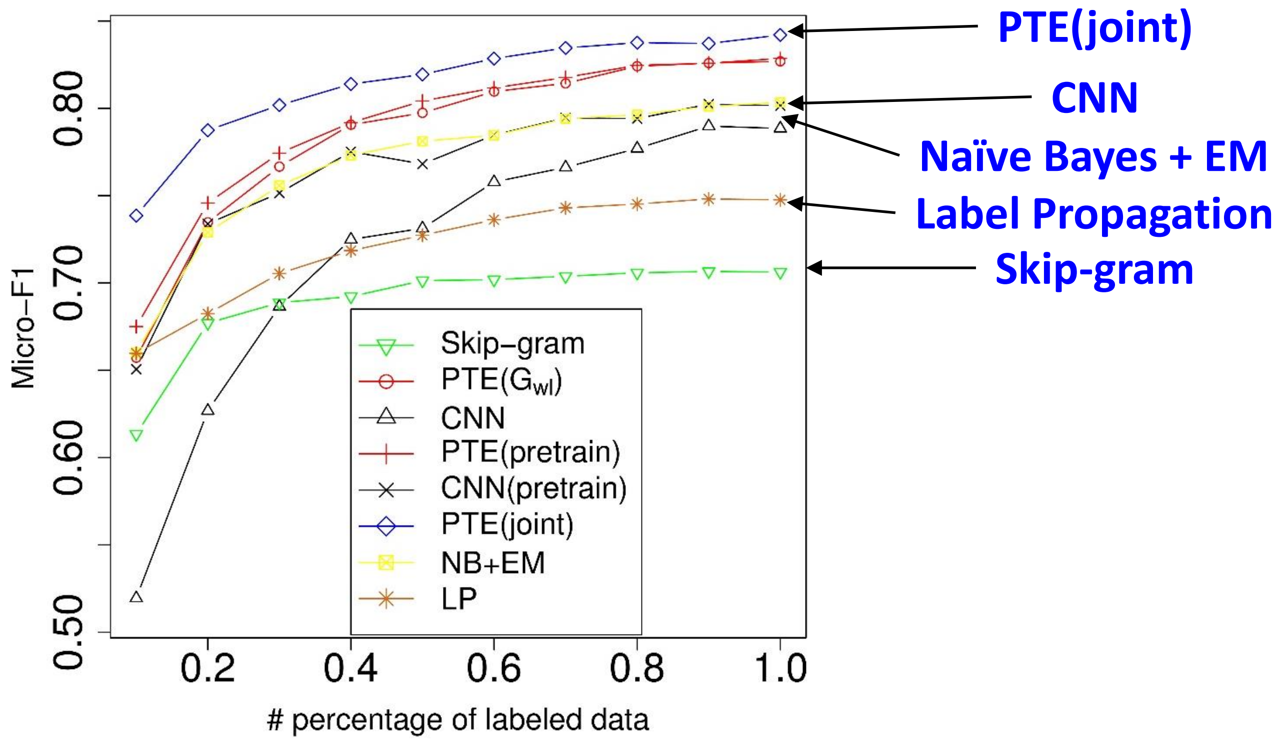
		DBLP		MR		Twitter	
Type	Algorithm	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Unsupervised embedding	LINE ($G_{ww} + G_{wd}$)	74.22	70.12	71.13	71.12	73.84	73.84
Predictive embedding	CNN	76.16	73.08	72.71	72.69	**75.97**	**75.96**
	CNN(pretrain)	75.39	72.28	68.96	68.87	75.92	75.92
	PTE(G_{wl})	76.45	72.74	73.44	73.42	73.92	73.91
	PTE($G_{ww} + G_{wl}$)	76.80	73.28	72.93	72.92	74.93	74.92
	PTE($G_{wd} + G_{wl}$)	**77.46**	**74.03**	73.13	73.11	75.61	75.61
	PTE(pretrain)	76.53	72.94	73.27	73.24	73.79	73.79
	PTE(joint)	77.15	73.61	**73.58**	**73.57**	75.21	75.21

PTE(joint) > PTE(pretrain)

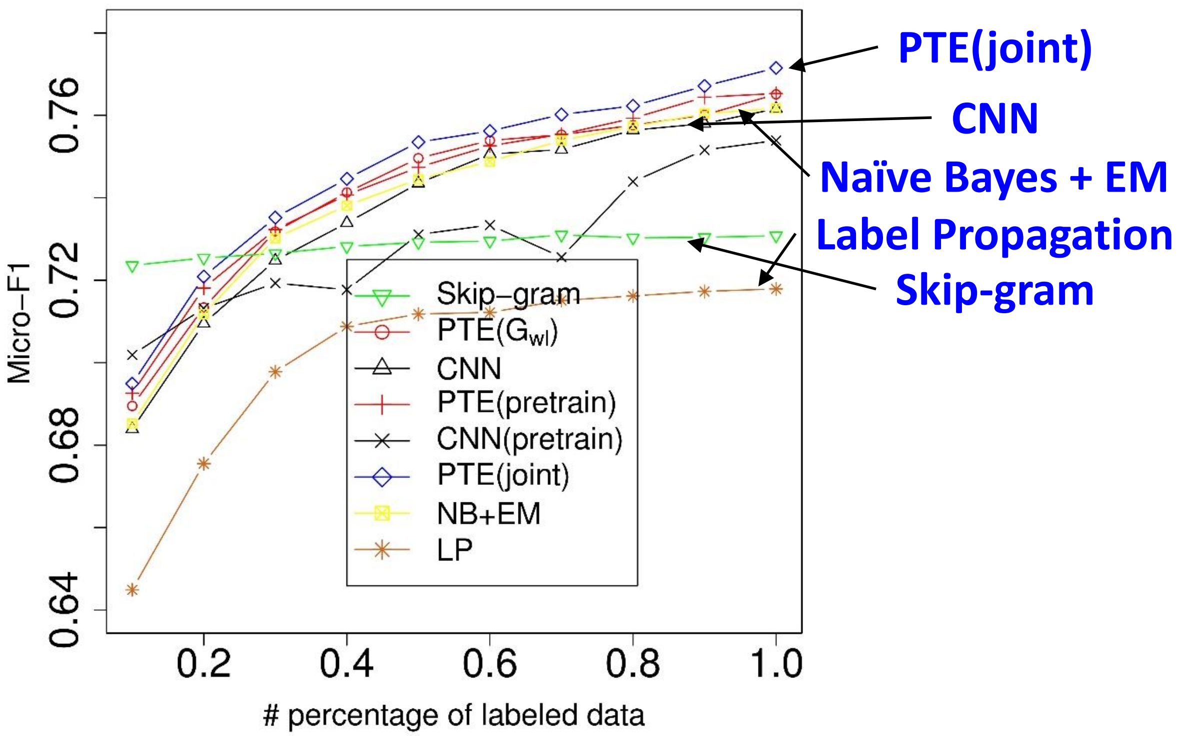
PTE(joint) > PTE(G_{wl})

PTE(joint) ≈ CNN/CNN(pretrain)

Performance w.r.t # Labeled Data



(a) 20newsgroup

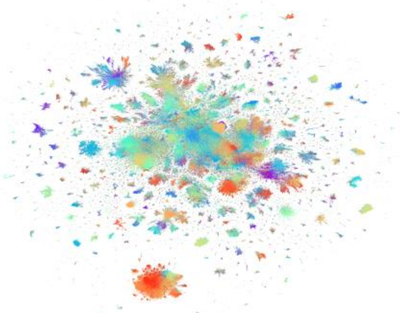


(b) DBLP

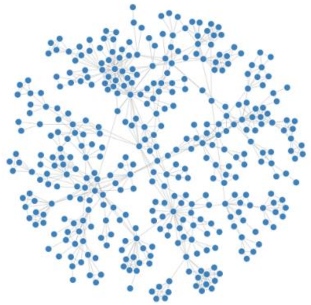
Large-scale Network/Data Visualization

Challenging to Visualize Big Data

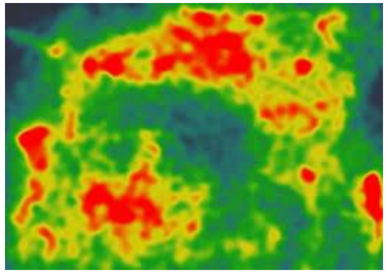
- *Intuitive* ways for data understanding and exploration
- Classical visualization techniques
  - Scatter plots, network diagrams, heatmaps,...
  - Requires 2D/3D layouts of data
- Real-world data are often Big
  - E.g., images, text, speech and networks
  - *Large-scale* (> millions) and *high-dimensional* (> hundreds)



Scatter Plots



Network Diagrams



Heatmaps

Goal: project big data into 2D/3D space

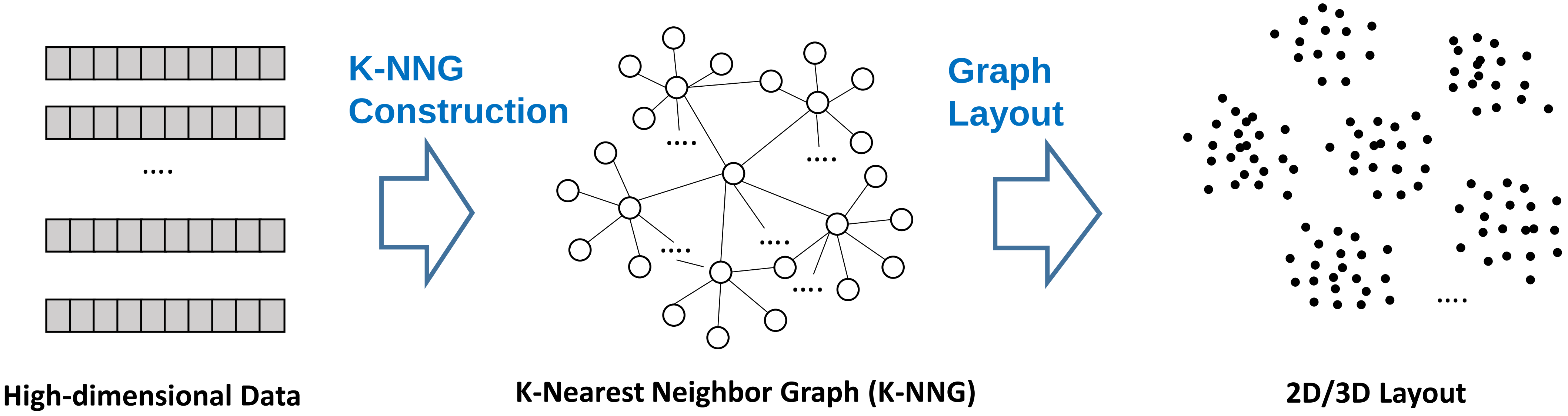
The State-of-the-art

- L. Maaten and Hinton G. “**Visualizing Data using t-SNE**”. 2011

[PDF] Visualizing Data using t-SNE - Department of Computer S…
www.cs.toronto.edu/~hinton/absps/**tsne**.pdf
by L van der Maaten - Cited by 1332 - Related articles
Visualizing Data using t-SNE. Laurens van der Maaten. L.VANDERMAATEN@
MICC.UNIMAAS.NL. MICC-IKAT. Maastricht University. P.O. Box 616, 6200 MD ...

- Not scalable to large scale data and graphs

Pipeline of t-SNE



- Limitations of t-SNE:
  - K-NNG construction: complexity grows *exponentially* to the data dimension
  - Graph layout: complexity is *O(NlogN),* where N is the number of data points
  - Very *sensitive* parameters

Our Approach: **LargeVis**

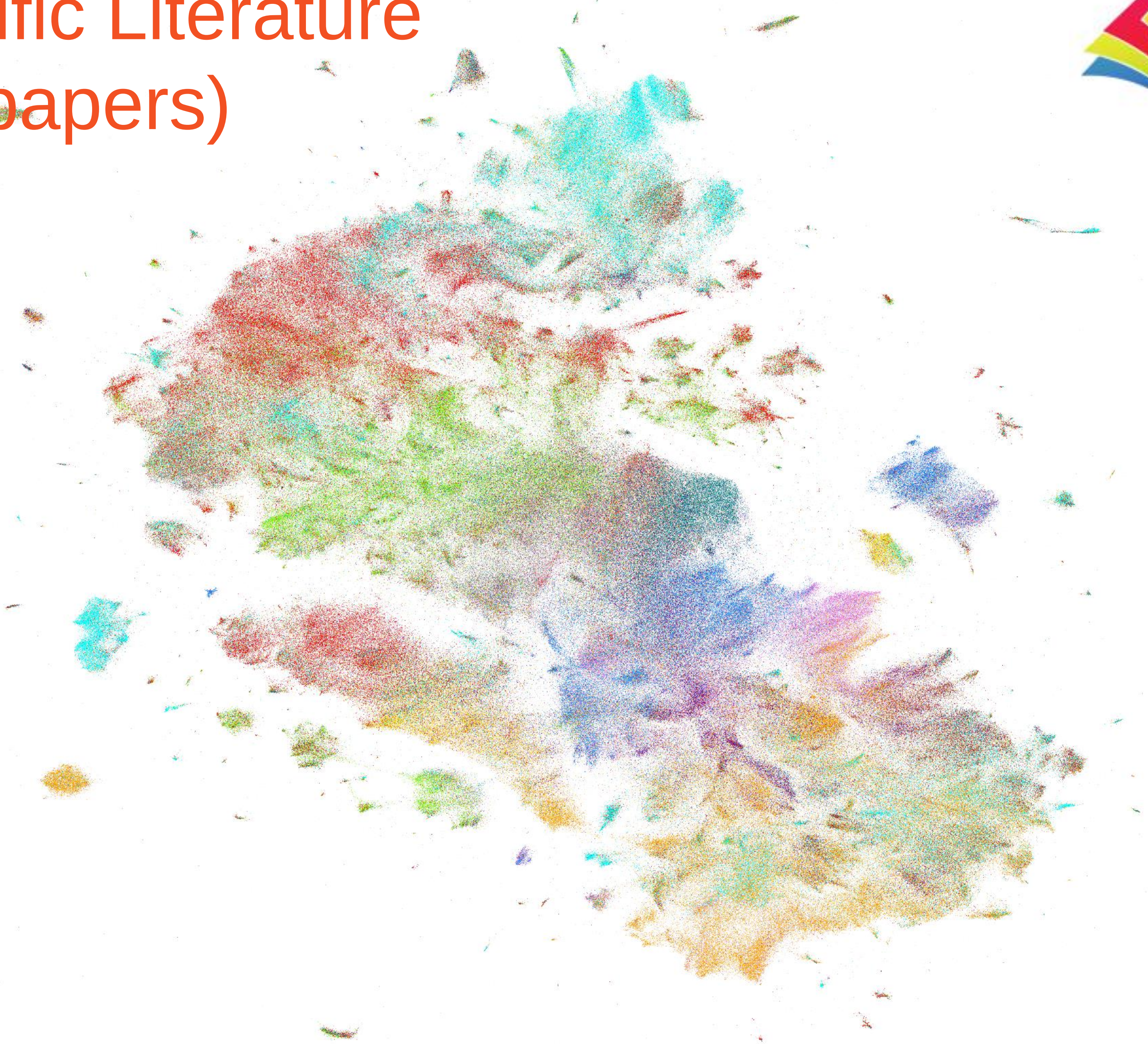
- An efficient approach for approximate K-NNG construction
  - ***Thirty*** times faster than t-SNE on 3 million data points
  - Better time-accuracy tradeoff

- An efficient probabilistic model for graph layout
  - O(NlogN) -> O(N)
  - ***Seven*** times faster than t-SNE on 3 million data points
  - More ***effective*** visualization layouts than t-SNE
  - ***Stable*** parameters across different data sets

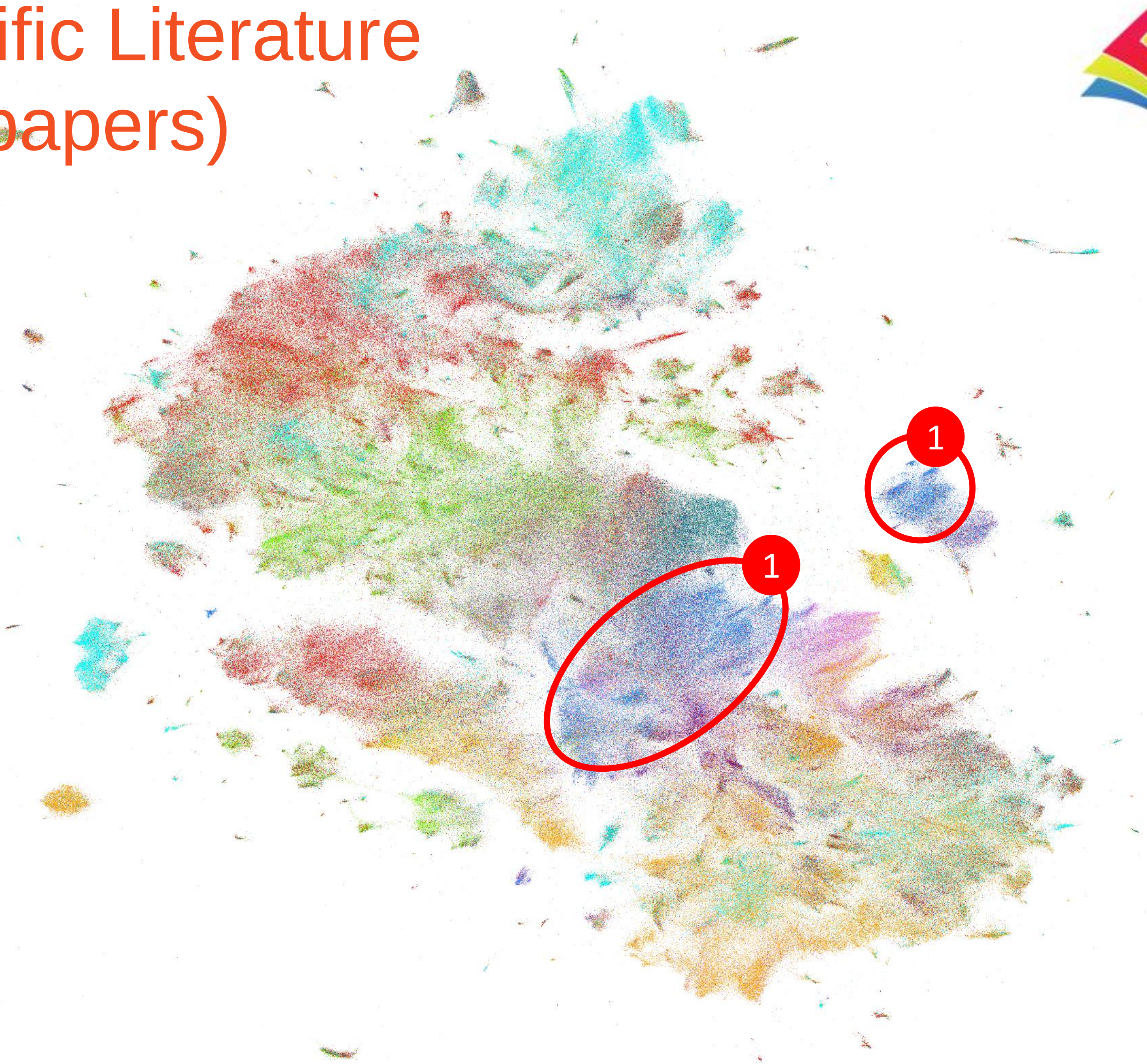
Tang et al. **"Visualizing Large-scale and High-dimensional Data."**
In WWW'16 **(Best paper nominee 5/727)**

Scientific Literature (10M papers)

Microsoft® Academic Search Beta



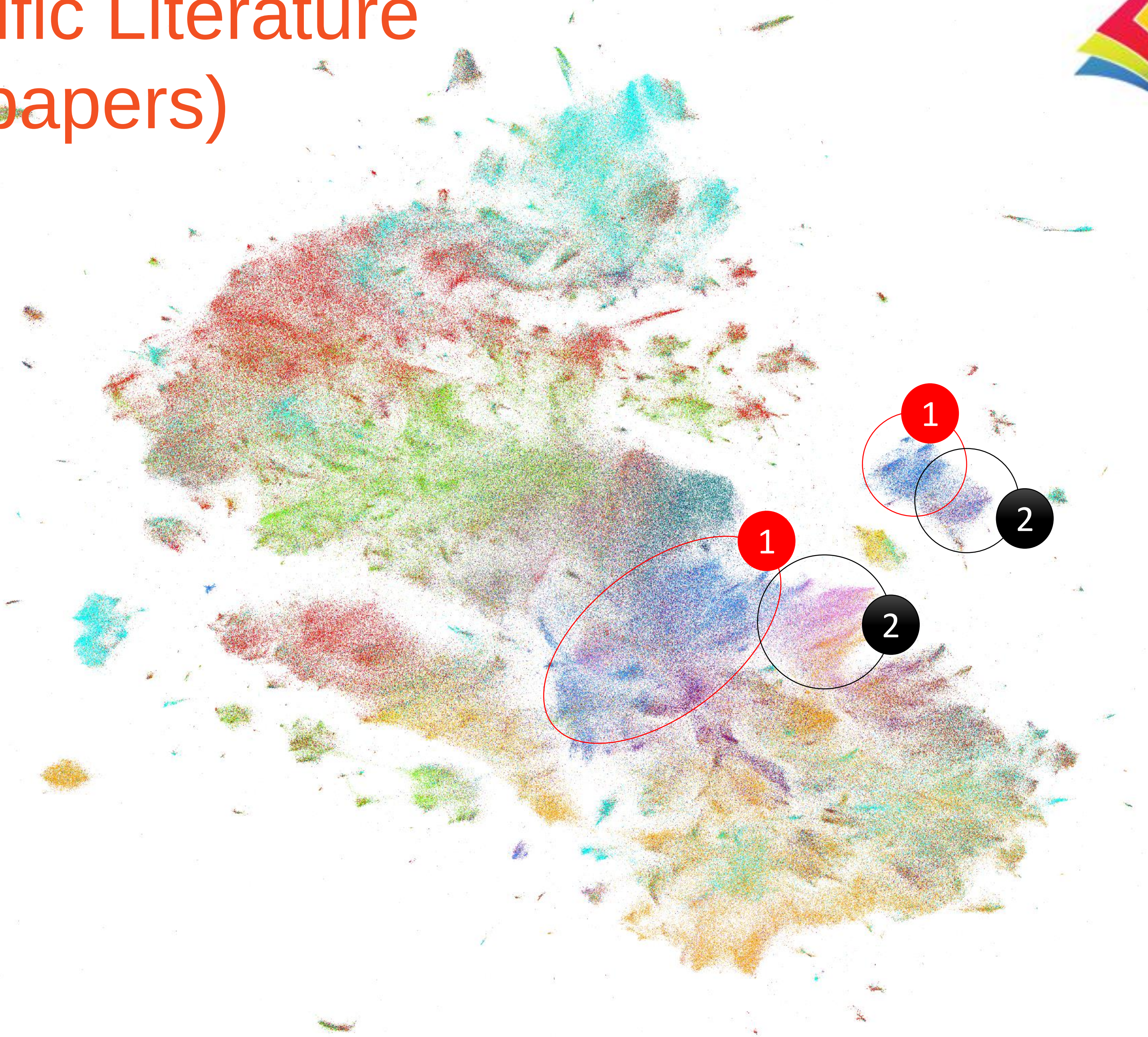
Scientific Literature (10M papers)



1 Computer Science

Scientific Literature (10M papers)

Microsoft Academic Search Beta

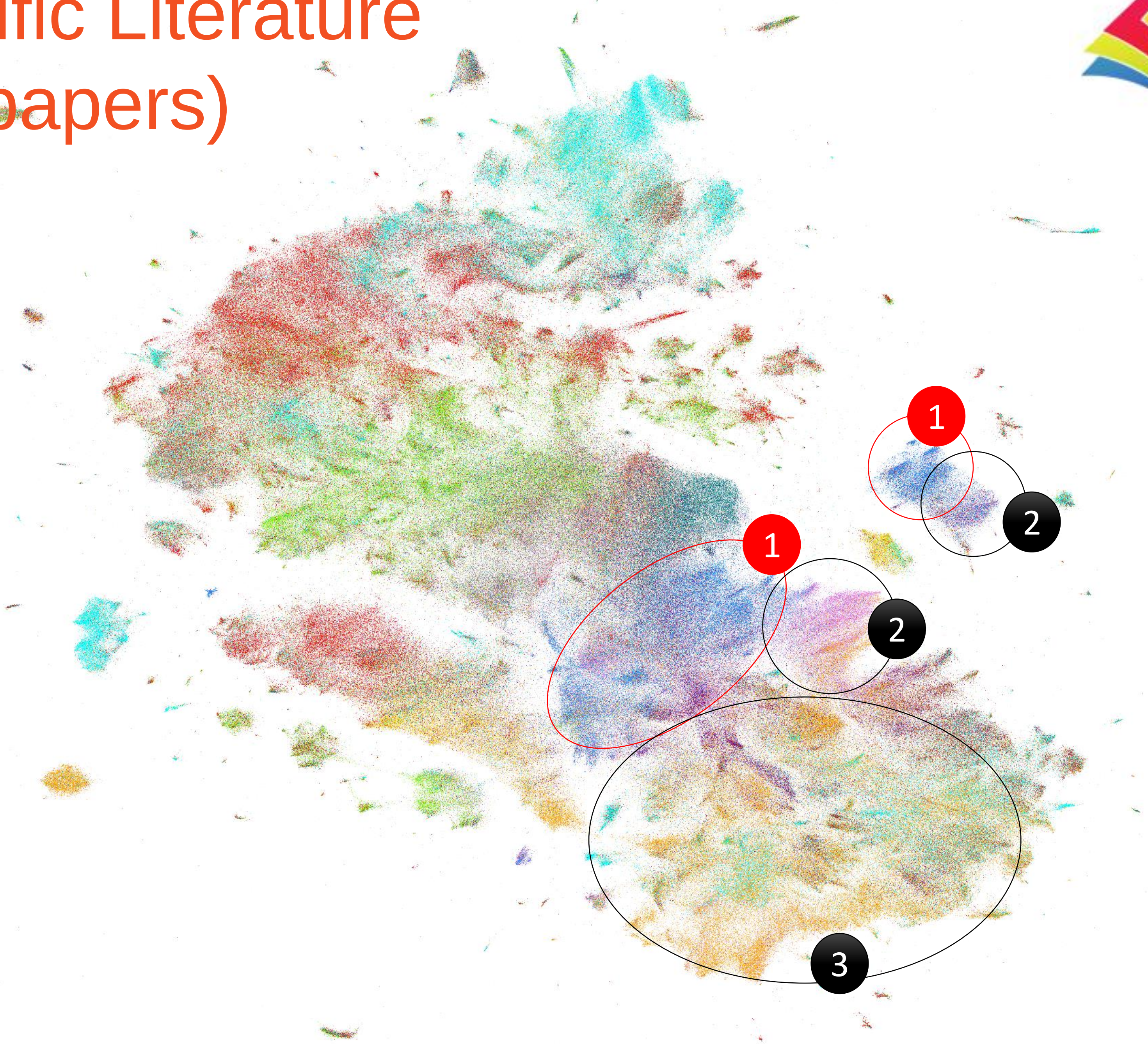


1 Computer Science

2 Mathematics

Scientific Literature (10M papers)

Microsoft® Academic Search Beta

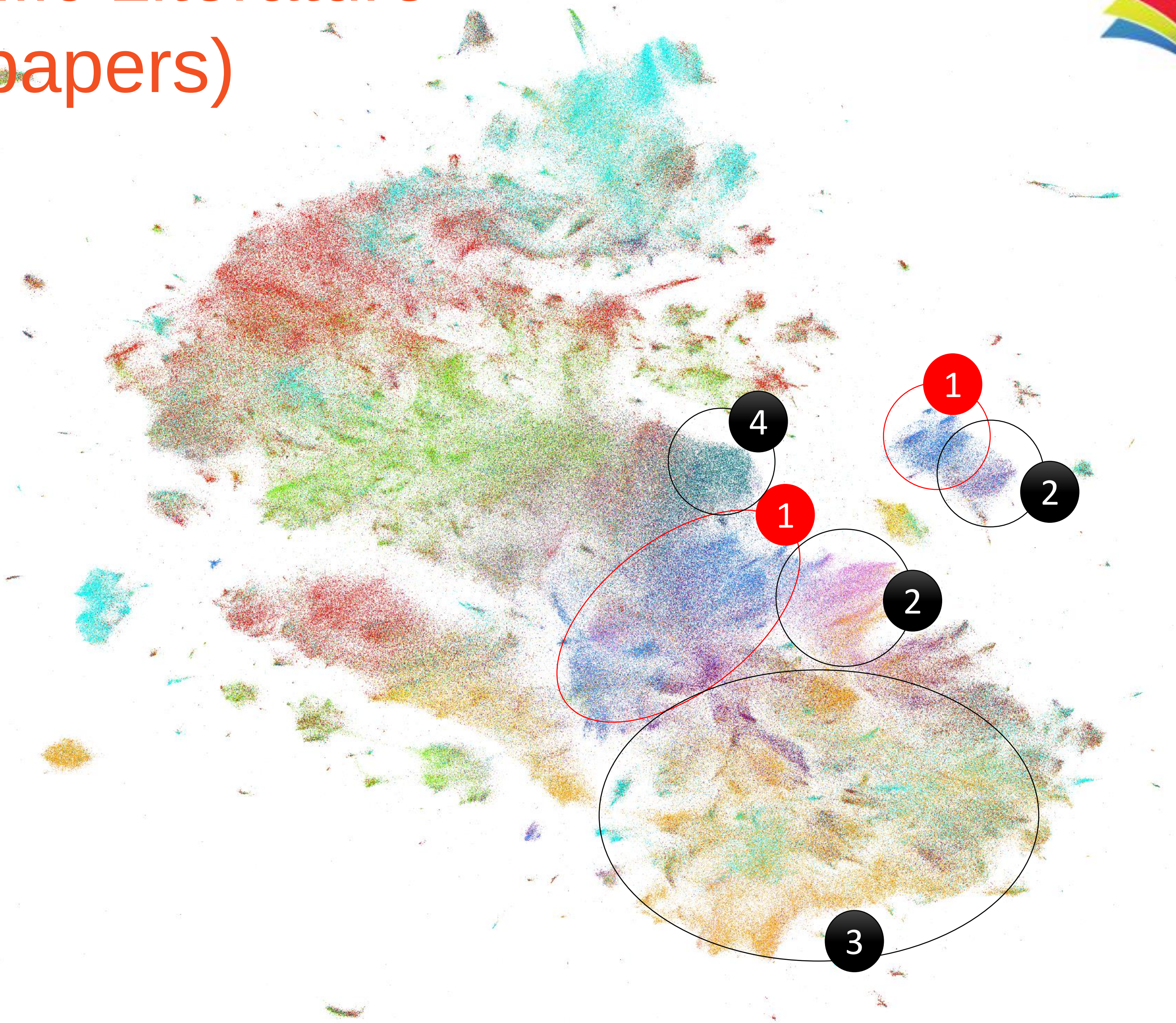


1 Computer Science

2 Mathematics

3 Physics

Scientific Literature (10M papers)



1 Computer Science

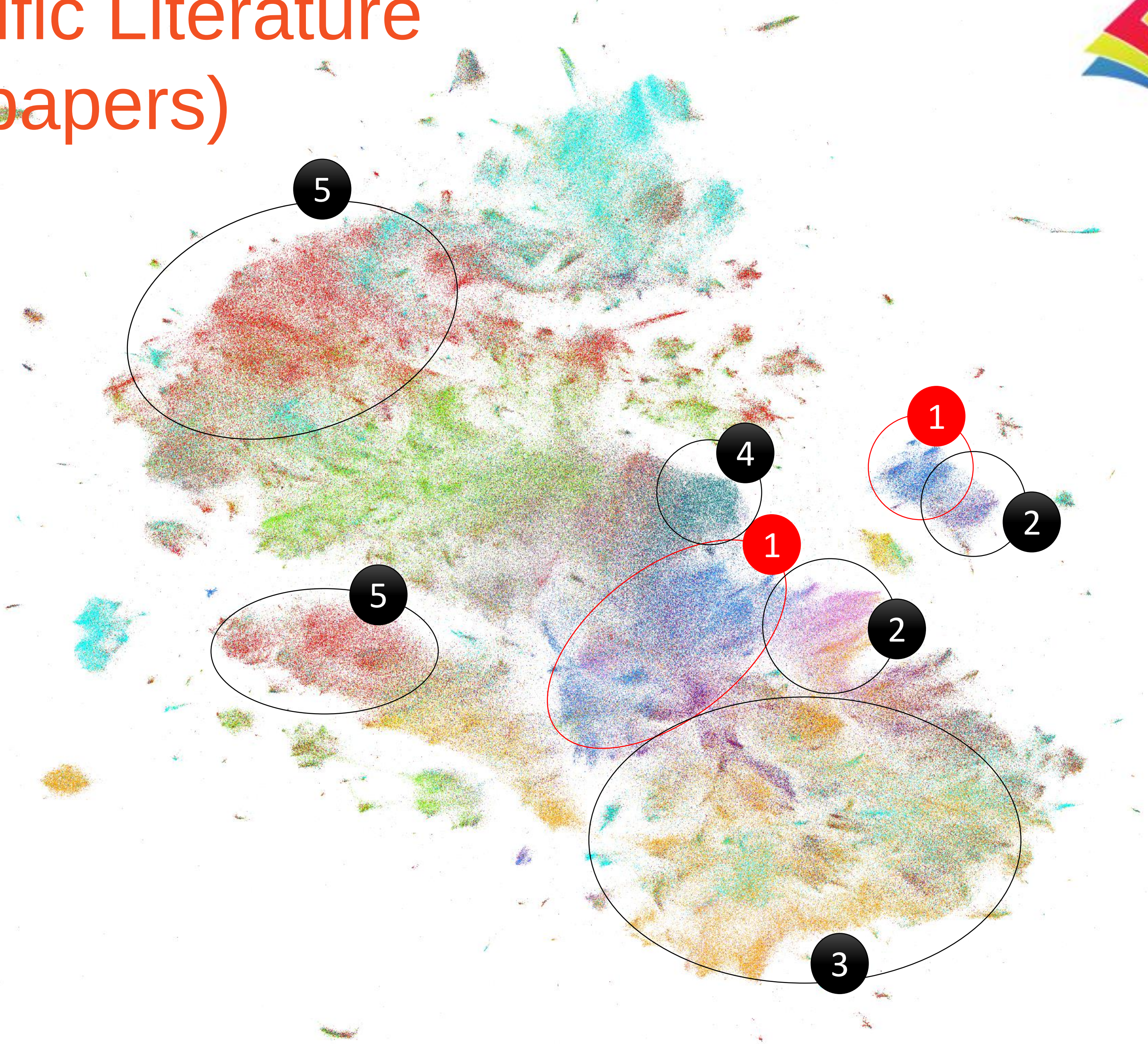
2 Mathematics

3 Physics

4 Economics

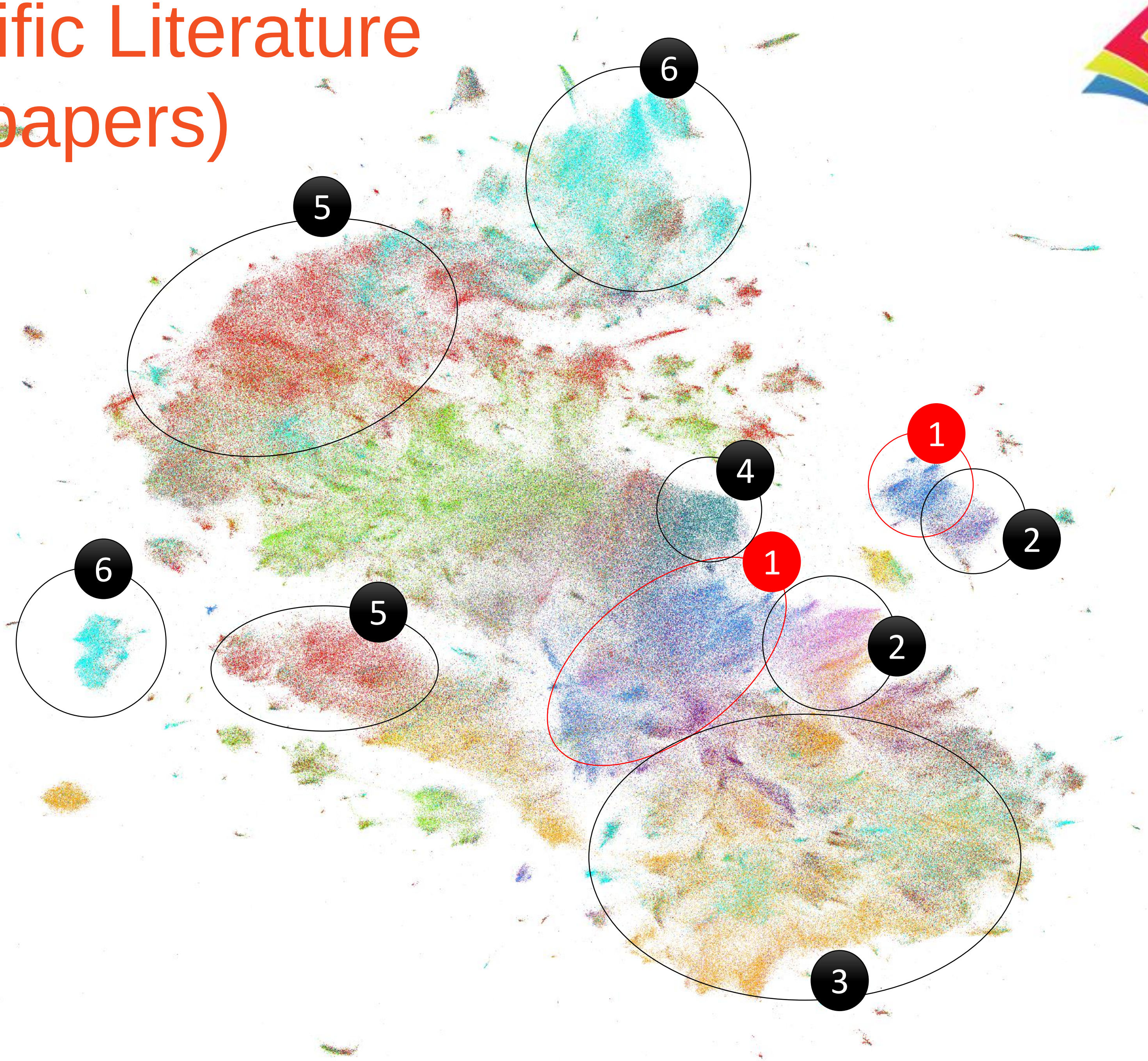
Scientific Literature (10M papers)

Microsoft® Academic Search Beta



1. Computer Science
2. Mathematics
3. Physics
4. Economics
5. Biology

Scientific Literature (10M papers)



1 Computer Science

2 Mathematics

3 Physics

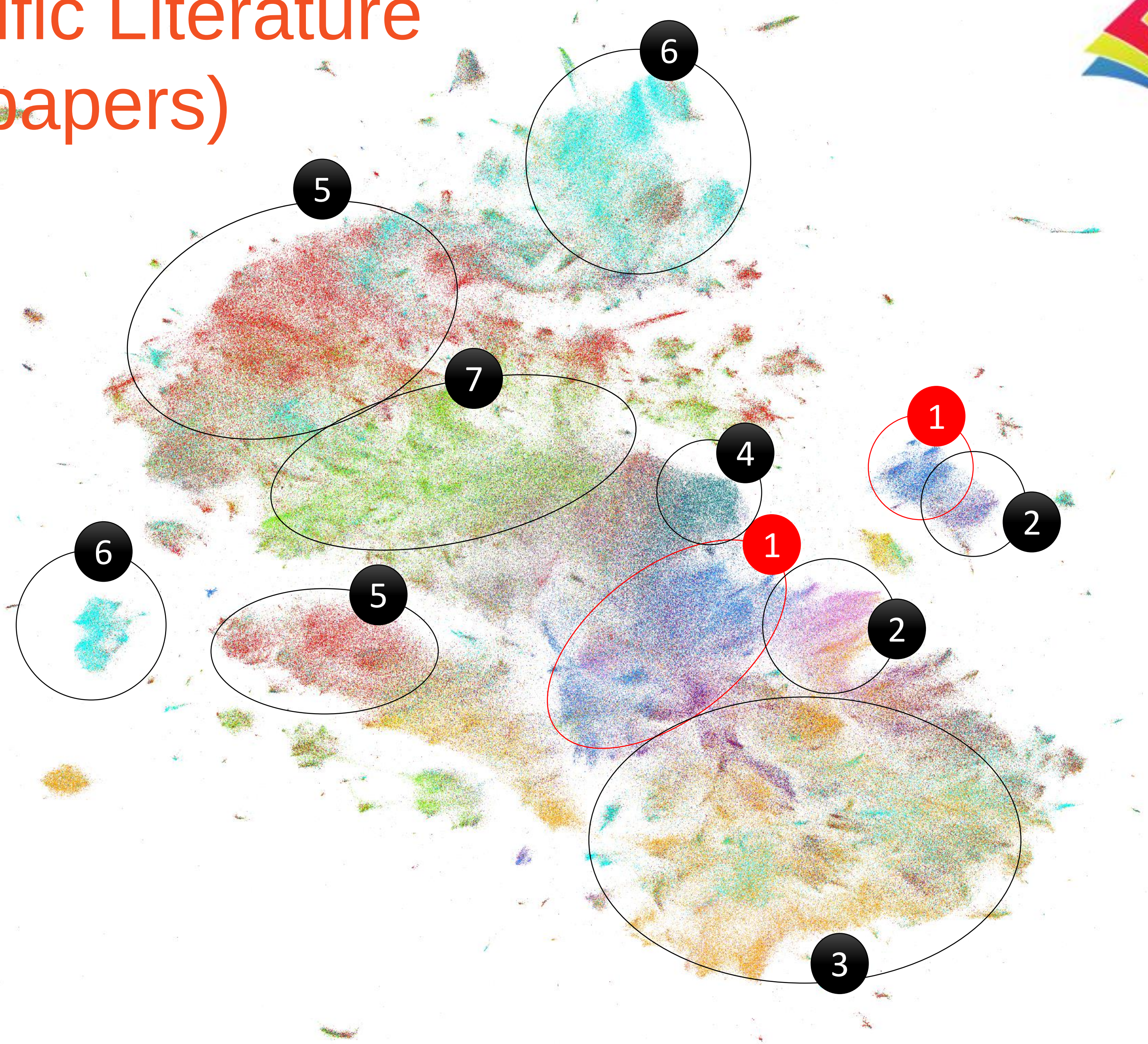
4 Economics

5 Biology

6 Chemistry

Scientific Literature (10M papers)

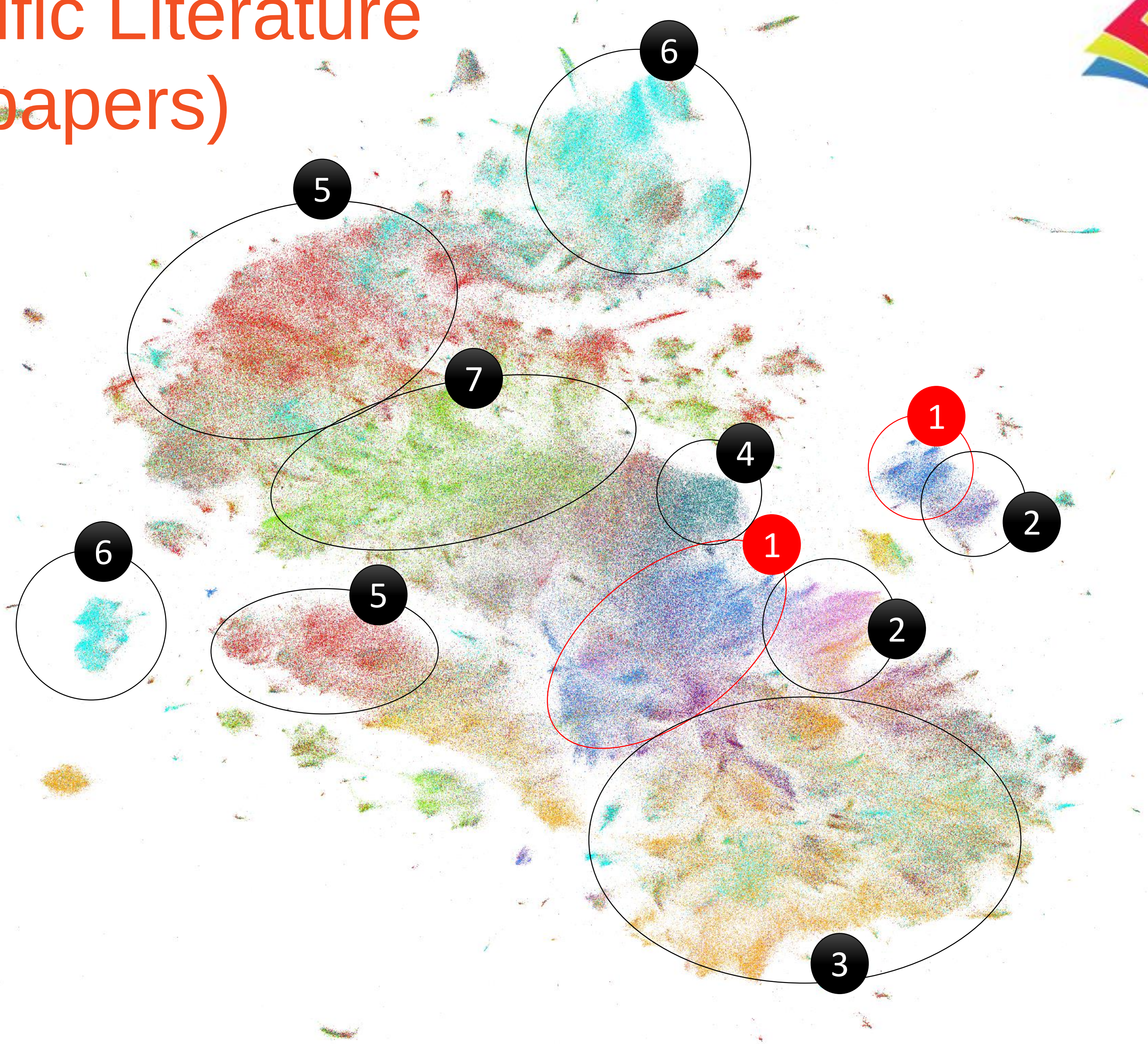
Microsoft® Academic Search Beta



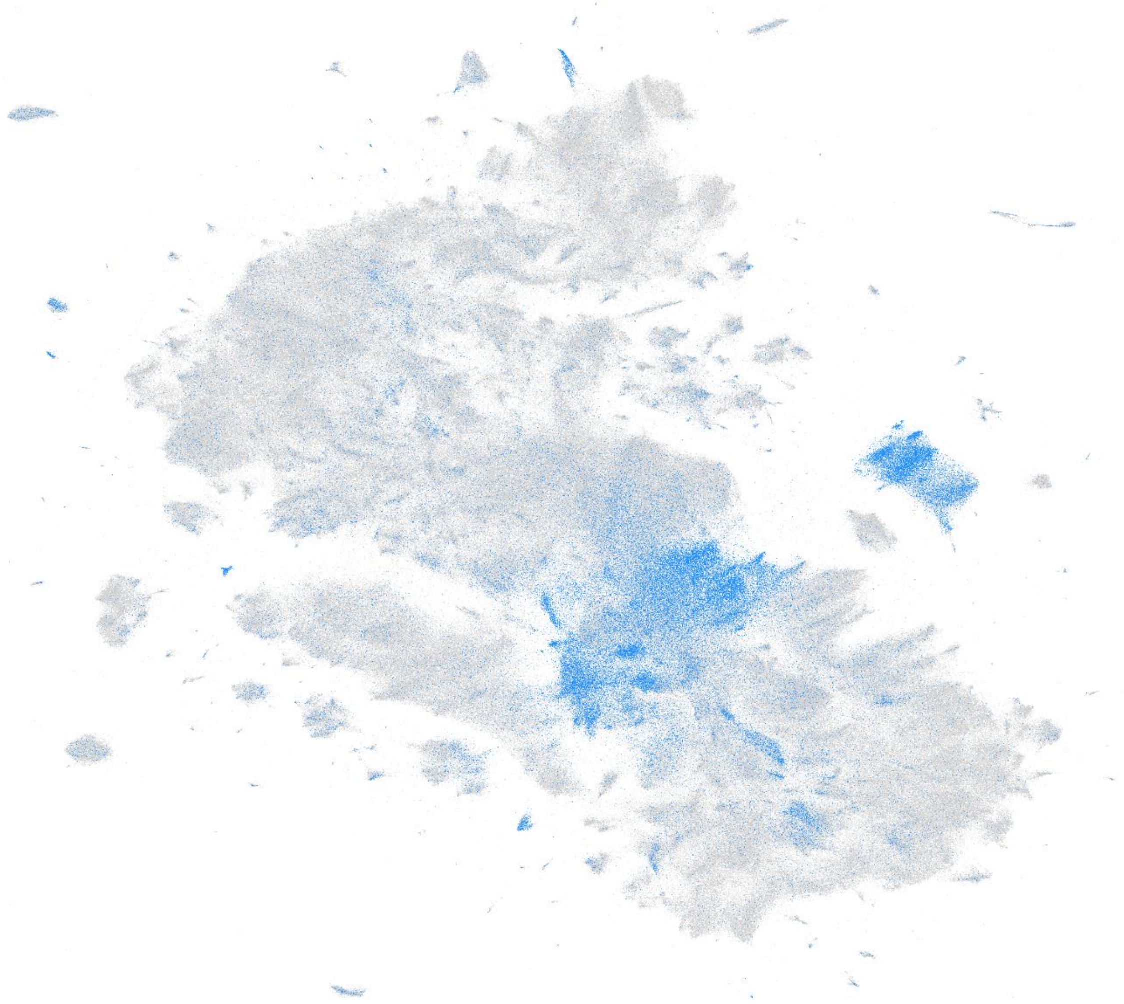
1 Computer Science
2 Mathematics
3 Physics
4 Economics
5 Biology
6 Chemistry
7 Medicine

Scientific Literature (10M papers)

Microsoft Academic Search Beta



1 Computer Science
2 Mathematics
3 Physics
4 Economics
5 Biology
6 Chemistry
7 Medicine



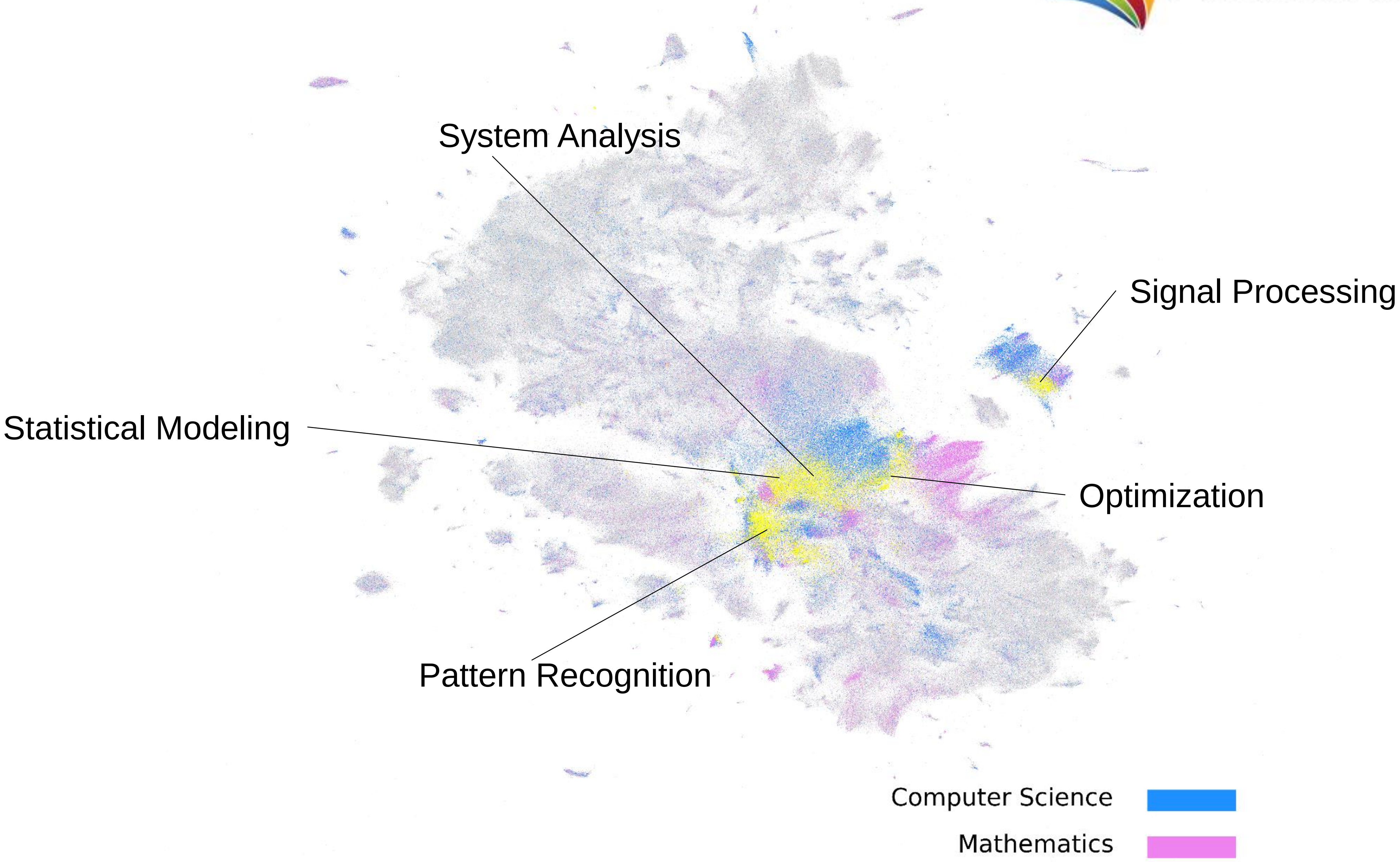
Computer Science

Mathematics

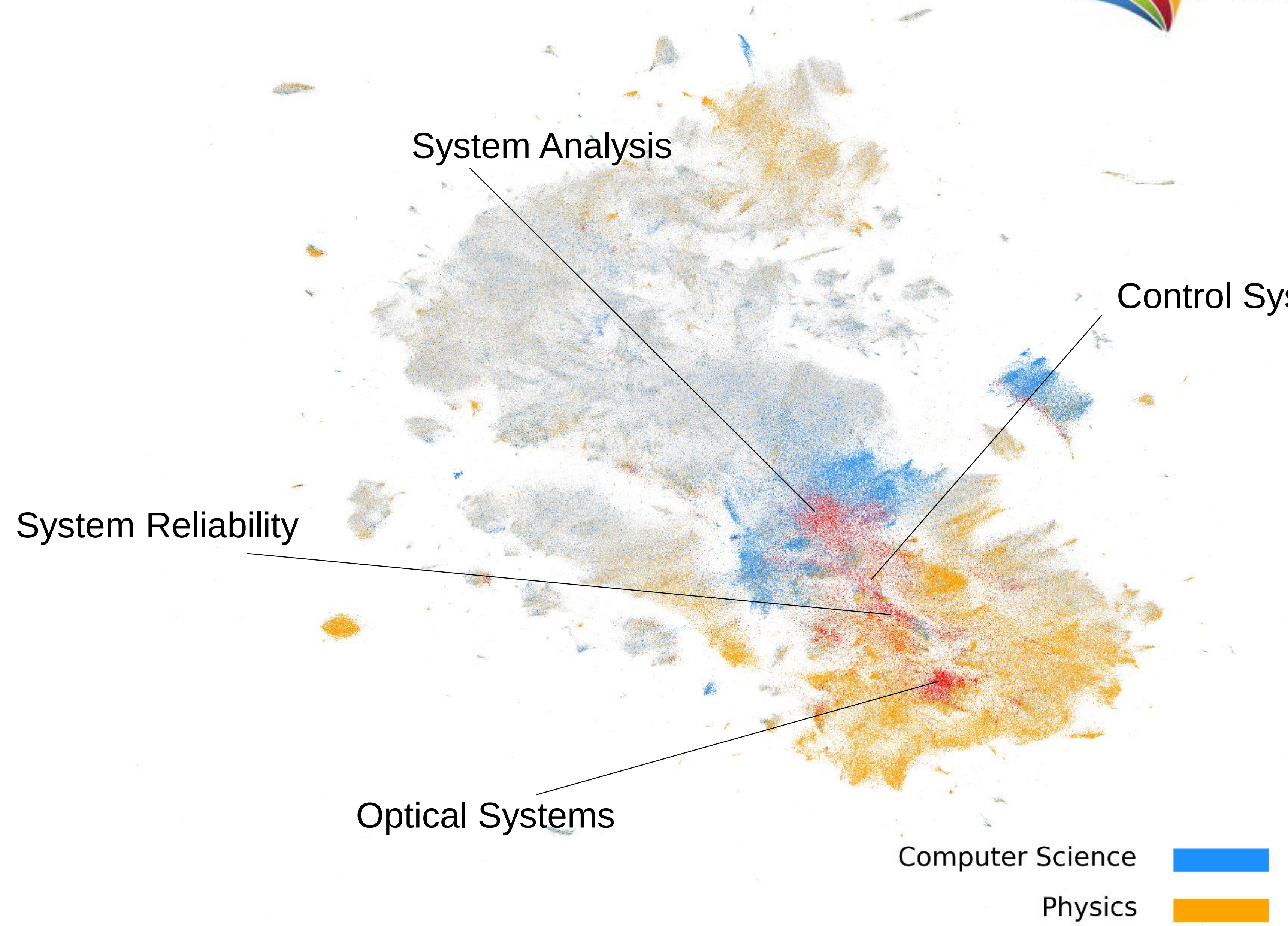
Physics

Biology

Computer Science vs. Mathematics



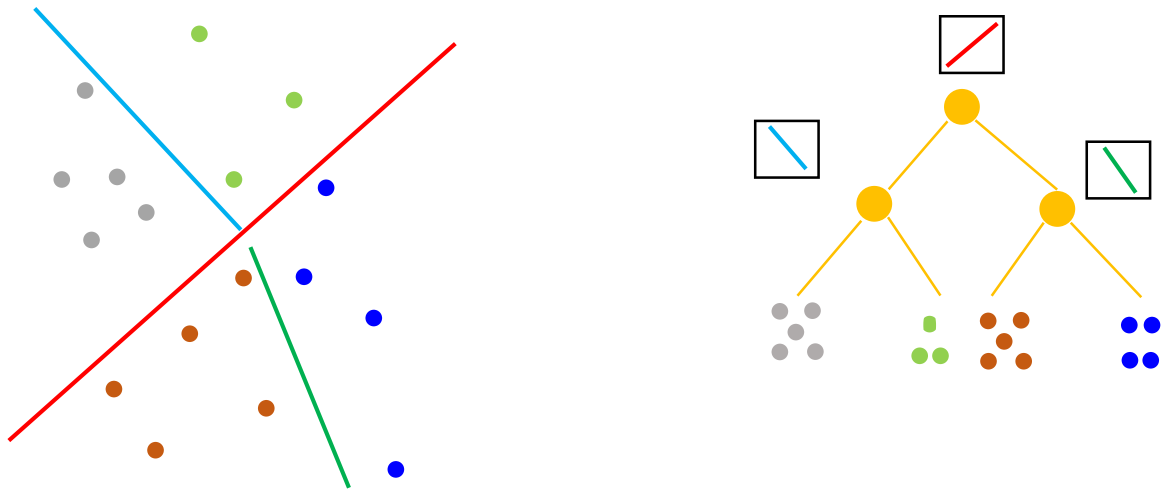
Computer Science vs. Physics



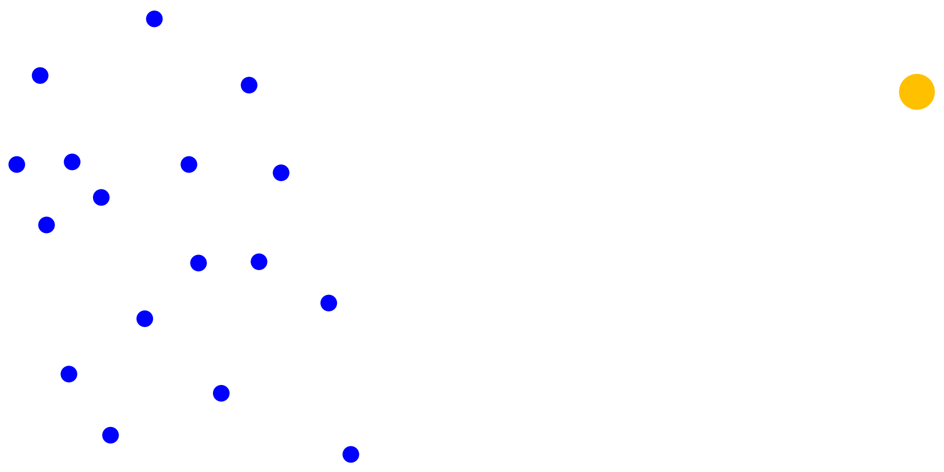
Network Construction

Random Projection Trees

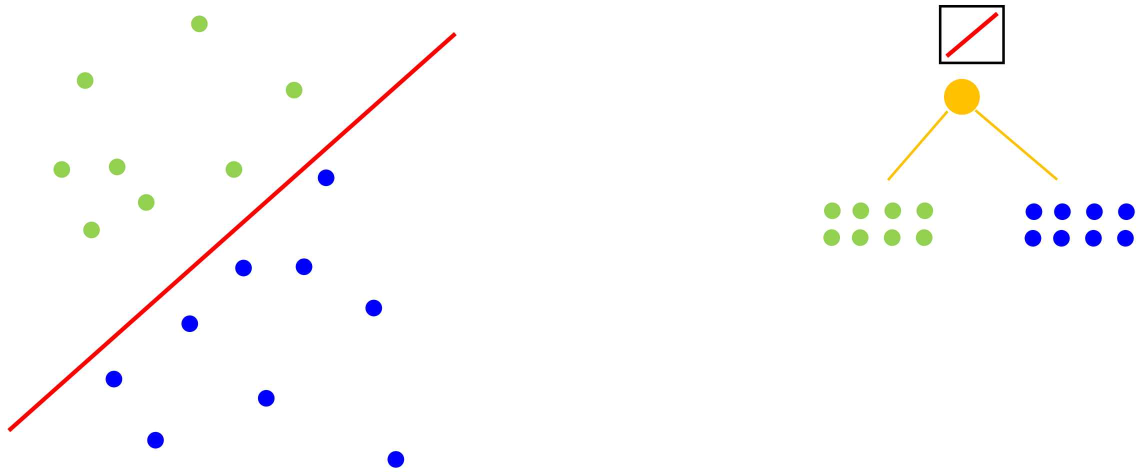
- Partition the whole space into different regions with multiple hyperplanes



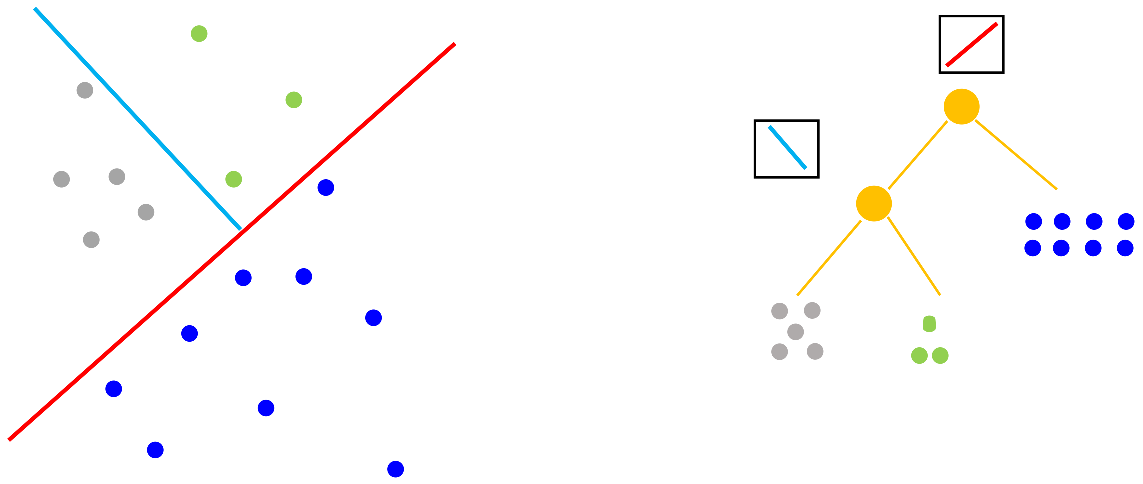
Random Projection Trees



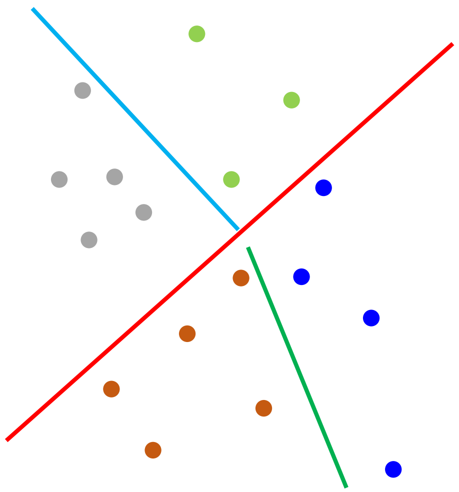
Random Projection Trees

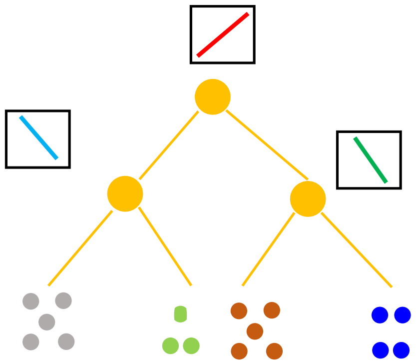


Random Projection Trees



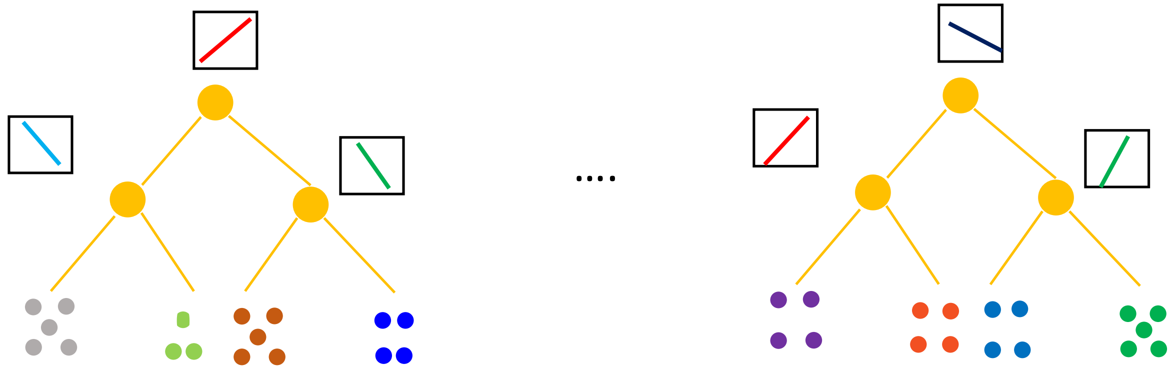
Random Projection Trees





K-NNG Construction

- Search nearest neighbors through traversing random projection trees
  - Only data points in the leaf are considered as nearest neighbors
- Multiple trees are usually used to improve the accuracy
  - e.g., hundreds



Reduce the Number of Trees

- Construct a less accurate K-NNG with *a few* trees
- Iteratively refine the K-NNG through "***neighbor exploring***"
  - "A neighbor of my neighbor is also likely to be my neighbor"
  - *Second-order* neighbors are also treated as candidates of *first-order* neighbors

Results of K-NNG Construction

- X axis: accuracy of K-NNG
  - With different values of parameters
- Y axis: running time (minutes)
- tSNE: 16 hours (95% accuracy)
- LargeVis: 25 minutes
  - ***>30*** times faster than t-SNE

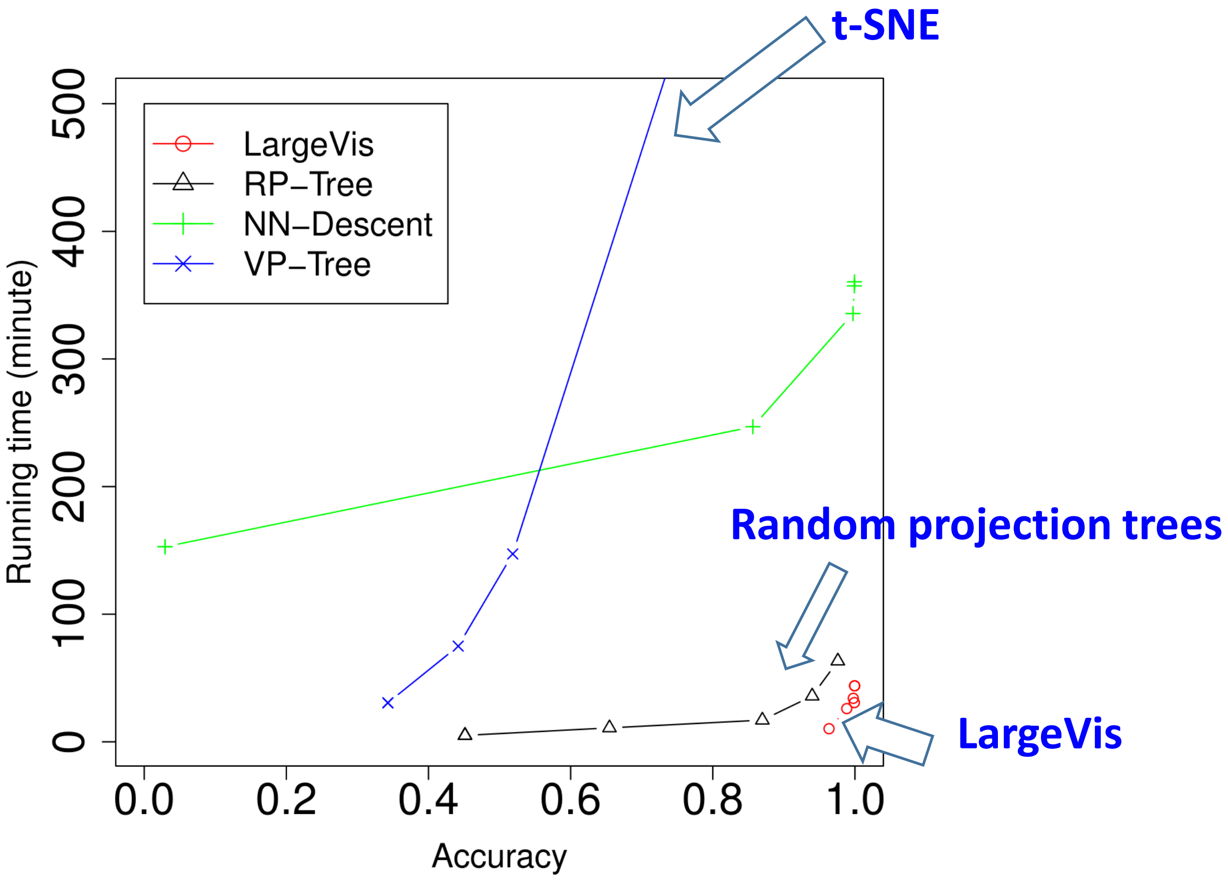


Fig.: Results on 3 Million Data with 100 Dimension

Network Visualization/Layout

A Probabilistic Model for Network Layout

- Preserve the similarities of the vertices in 2D/3D space
  - Represent each vertex *i* with a 2D/3D vector $\vec{y}_i$
  - Keep *similar* data close while *dissimilar* data far apart
- Probability of observing a *binary* edge between vertices *(i,j)*:

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

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

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

A Probabilistic Model for Network Layout

- Objective:

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

Weight of the negative edges

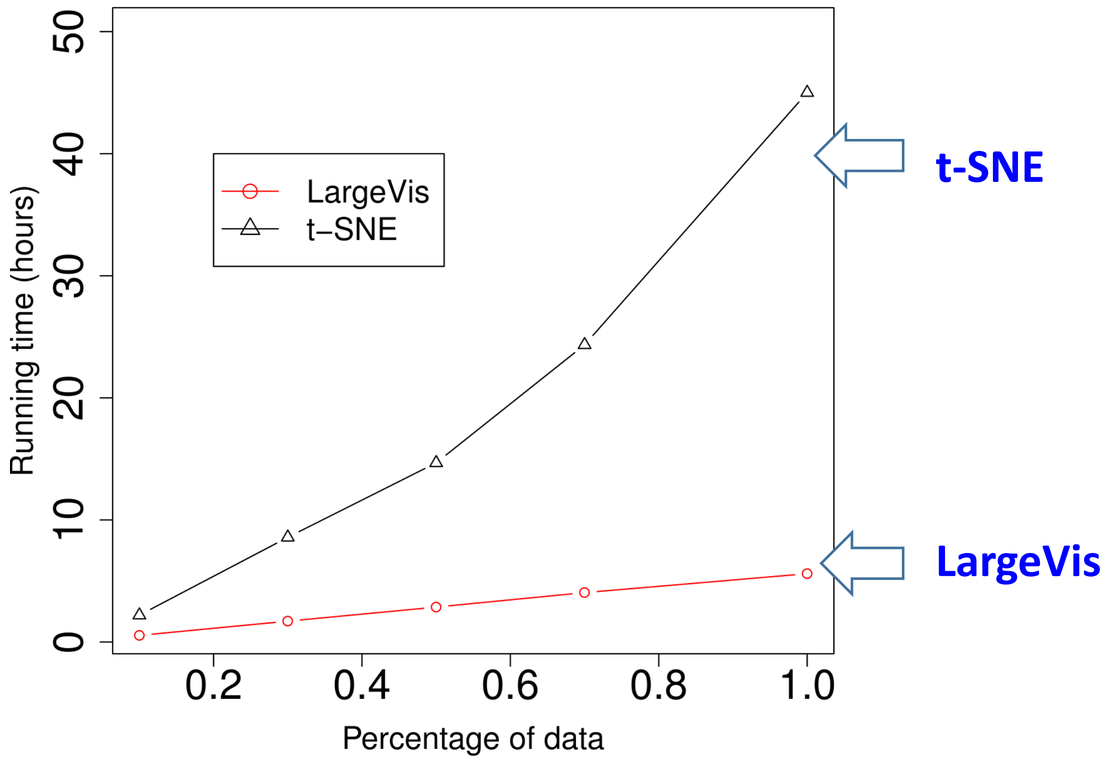
Positive edges

Negative edges

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

Efficiency of Network Layout

- Time complexity
  - t-SNE: O(NlogN)
  - LargeVis: O(N)
- On 3 million data points
  - t-SNE: 45 hours
  - LargeVis: 5.6 hours
  - *Seven* times faster



Visualization Quality

- Metric: *classification accuracy* with KNN on 2D space
- Configuration:
  - LargeVis with *default* parameters
  - t-SNE with *default* and *optimal* parameters (tuned per data set)
- LargeVis ≈ tSNE with optimal parameters
- LargeVis >> tSNE with default parameters
- Parameters of LargeVis are very *stable*

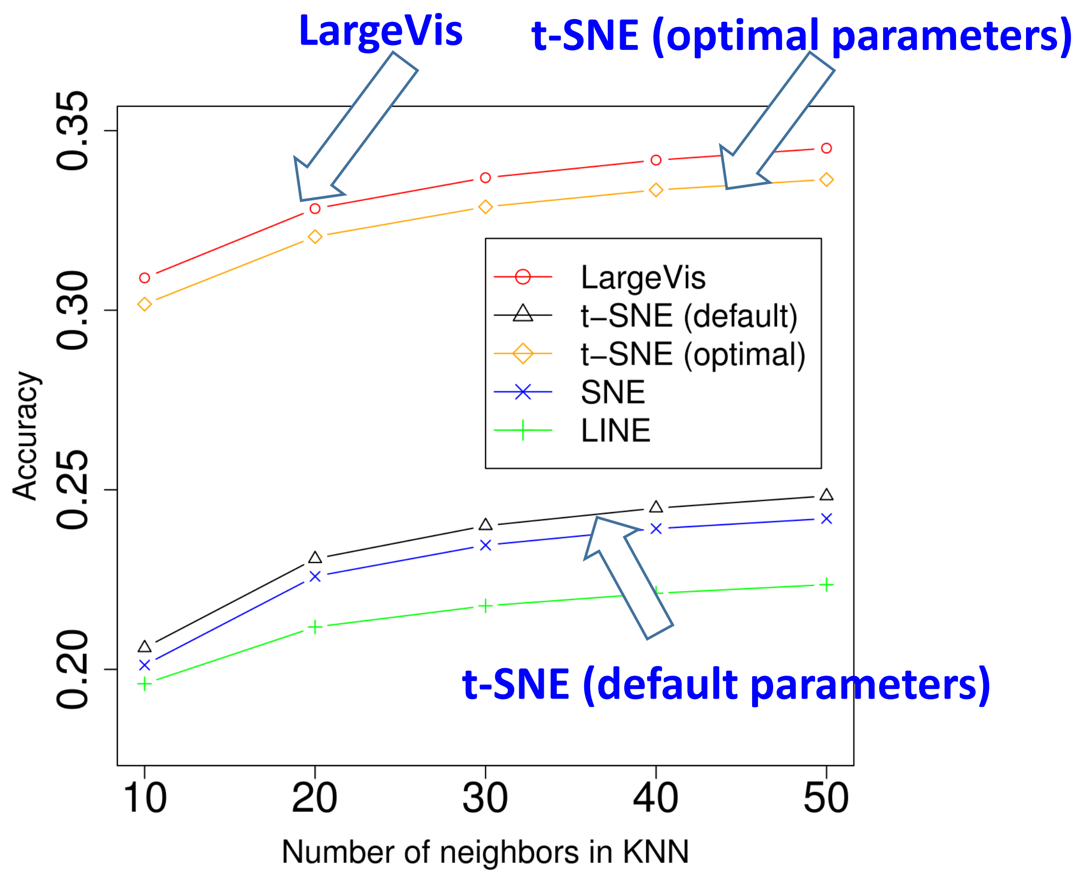


Fig.: Results on 3 Million Data with 100 Dimension

Large-scale Network Representation, Visualization and **Reasoning**

- **Reasoning and Prediction on Graphs**
- **More applications, e.g., recommendation**

- Big Graph/Data Visualization (WWW'16)
- Best paper nominee 5/727

- Graph Representation for text representation (KDD'15)
- State-of-the-art results for word and document embedding

- LINE: Large-scale Graph Embedding/Representation (WWW'15)
- Social graph, academic graph representation
- A benchmark model in this area