Graph topology inference benchmarks for machine learning

Carlos Lassance[1], Vincent Gripon[1] and Gonzalo Mateos[2]

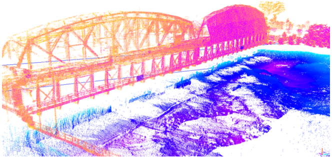
[1] IMT Atlantique, [2] University of Rochester

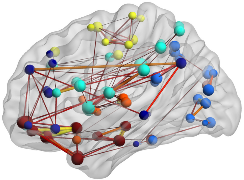
MLSP - September 2020

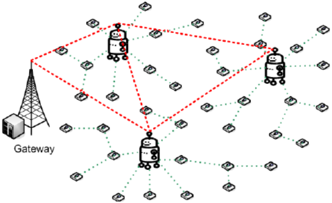
Benchmarks available at:
https://github.com/cadurosar/benchmark_graphinference

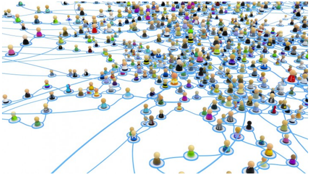
Motivation

Graphs are ubiquitous in machine learning





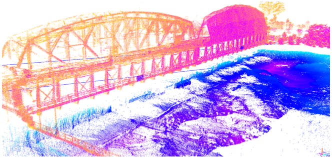


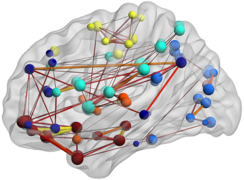


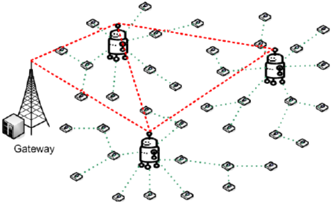
What is the best graph for my considered task?

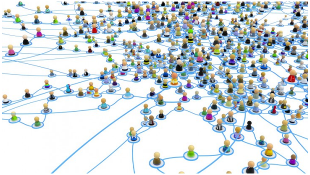
Motivation

Graphs are ubiquitous in machine learning









What is the best graph for my considered task?

Outline

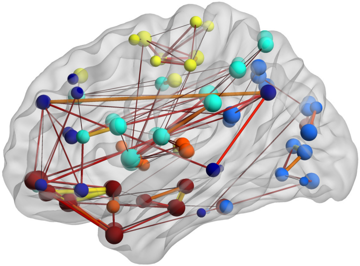
1. Graph inference
2. Proposed benchmark framework
3. Baseline, results and findings
4. Conclusion

Outline

1. Graph inference
2. Proposed benchmark framework
3. Baseline, results and findings
4. Conclusion

Graphs

- Graphs express relationships between items;
- Ubiquitous in machine learning;
- **Problem:** Not always available;
- *Solution:* Infer a graph from the data.



Graph Inference

- There are numerous ways to infer a graph from data:
  - Stationarity [Pasdeloup et al 2017, Segarra et al 2018];
  - Probabilistic [Egilmez et al 2017];
  - Smoothness [Kalofolias et al 2019];
  - Sparsity [Shekizzar and Ortega 2020].

Key question: what method offers the best performance?

Example of graph inference:

Graph Inference

- There are numerous ways to infer a graph from data:
  - Stationarity [Pasdeloup et al 2017, Segarra et al 2018];
  - Probabilistic [Egilmez et al 2017];
  - Smoothness [Kalofolias et al 2019];
  - Sparsity [Shekizzar and Ortega 2020].

Key question: what method offers the best performance?

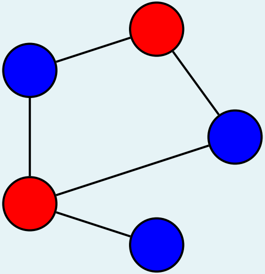
Example of graph inference:

Graph Inference

- There are numerous ways to infer a graph from data:
  - Stationarity [Pasdeloup et al 2017, Segarra et al 2018];
  - Probabilistic [Egilmez et al 2017];
  - Smoothness [Kalofolias et al 2019];
  - Sparsity [Shekizzar and Ortega 2020].

Key question: what method offers the best performance?

Example of graph inference:



Graph Inference

Limitations of existing works

- Techniques are developed using specific priors;
- Those priors might or might not be aligned with various tasks;
- Benchmarking graph inference requires to be able to showcase these different characteristics;
- Often rely on synthetic data and graphs aligned with their priors.

Our contribution

- We introduce a comprehensive set of benchmarks meant to:
  - Represent a diverse set of cases;
  - Use real data and real tasks;
  - Be easy-to-use and fair for comparison purposes;
  - Reflect the pros and cons of proposed methods.

Graph Inference

Limitations of existing works

- Techniques are developed using specific priors;
- Those priors might or might not be aligned with various tasks;
- Benchmarking graph inference requires to be able to showcase these different characteristics;
- Often rely on synthetic data and graphs aligned with their priors.

Our contribution

- We introduce a comprehensive set of benchmarks meant to:
  - Represent a diverse set of cases;
  - Use real data and real tasks;
  - Be easy-to-use and fair for comparison purposes;
  - Reflect the pros and cons of proposed methods.

Outline

Proposed benchmark

Main contribution

Tasks

1. Unsupervised Clustering of Vertices;
2. Semi-Supervised Classification of Vertices;
3. Denoising of Graph Signals.

Datasets for tasks 1 and 2

Graphs encode relationship between observations:

- Images - flowers102
- Audio - ESC-50
- Text - cora

Dataset for task 3

Graph of relationship between features

- Vehicule traffic volume denoising - Toronto.

Proposed benchmark

Main contribution

Tasks

1. Unsupervised Clustering of Vertices;
2. Semi-Supervised Classification of Vertices;
3. Denoising of Graph Signals.

Datasets for tasks 1 and 2

Graphs encode relationship between observations:

- Images - flowers102
- Audio - ESC-50
- Text - cora

Dataset for task 3

Graph of relationship between features

- Vehicule traffic volume denoising - Toronto.

Proposed benchmark

Main contribution

Tasks

1. Unsupervised Clustering of Vertices;
2. Semi-Supervised Classification of Vertices;
3. Denoising of Graph Signals.

Datasets for tasks 1 and 2

Graphs encode relationship between observations:

- Images - flowers102
- Audio - ESC-50
- Text - cora

Dataset for task 3

Graph of relationship between features

- Vehicule traffic volume denoising - Toronto.

Objective

Evaluate how aligned is the inferred graph structure to the unsupervised clustering.

Evaluation

AMI of the Spectral Clustering using the inferred graph.

Task 2

Semi-Supervised Classification of Vertices

Objective

Evaluate the performance of the inferred graph in identifying the class of each vertex.

Evaluation

Two types of evaluation:

- Label only: Test set accuracy using label propagation on the graph;
- Label and Features: Test set accuracy using SGC[Wu et al 2019].

Averaged over 100 runs using 5% of labeled nodes.

Task 1 and 2 Datasets

Image dataset - flowers102 [Nilsback and Zisserman 2008]

- Flower identification;
- From flowers102 [Nilsback and Zisserman 2008]
- InceptionV2 [Szegedy et al 2016] features;
- Very challenging:
  - High amount of classes (102)
  - High signal to items ratio (2)

Task 1 and 2 Datasets

Audio dataset - ESC-50 [Piczak 2015]

- Environment classification via audio;
- Features extracted with [Kumar et al 2018];
- Easier than the image dataset:
  - Medium amount of classes (50)
  - Small signal to items ratio (0.512)

Task 1 and 2 Datasets

Text dataset - Cora [Sen et al 2008]

- Scientific article classification;
- Has a citation graph that we do not use;
- Medium difficulty dataset:
    - Small amount of classes (7);
    - Small signal to items ratio (0.53);
    - Bag of Words -> binary features.

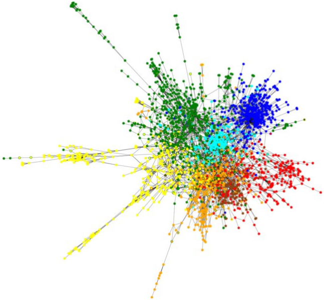


Figure extracted from [Monti et al 2017]

Task 3

Denoising of Graph Signals

Objective

Performance of the inferred graph on graph signal denoising.

Evaluation

Signal-to-noise ratio of the signal denoised by a Simoncelli graph filter.

$$f_l = \begin{cases} 1 & \text{if } \lambda_l \leq \frac{\tau}{2} \\ \cos\left(\frac{\pi}{2}\frac{\log(\lambda_l)}{\log(2)}\right) & \text{if } \frac{\tau}{2} < \lambda_l \leq \tau \\ 0 & \text{if } \lambda_l > \tau \end{cases},$$

Dataset

Toronto traffic data with added noise [Irion and Saito 2016].

Objective

Performance of the inferred graph on graph signal denoising.

Evaluation

Signal-to-noise ratio of the signal denoised by a Simoncelli graph filter.

$$f_l = \begin{cases} 1 & \text{if } \lambda_l \leq \frac{\tau}{2} \\ \cos\left(\frac{\pi}{2}\frac{\log(\lambda_l)}{\log(2)}\right) & \text{if } \frac{\tau}{2} < \lambda_l \leq \tau \\ 0 & \text{if } \lambda_l > \tau \end{cases},$$

Dataset

Toronto traffic data with added noise [Irion and Saito 2016].

Task 3

Denoising of Graph Signals

Objective

Performance of the inferred graph on graph signal denoising.

Evaluation

Signal-to-noise ratio of the signal denoised by a Simoncelli graph filter.

$$f_l = \begin{cases} 1 & \text{if } \lambda_l \leq \frac{\tau}{2} \\ \cos\left(\frac{\pi}{2}\frac{\log(\lambda_l)}{\log(2)}\right) & \text{if } \frac{\tau}{2} < \lambda_l \leq \tau \\ 0 & \text{if } \lambda_l > \tau \end{cases},$$

Dataset

Toronto traffic data with added noise [Irion and Saito 2016].

Outline

Proposed baselines

Naive baselines

Baselines based on four steps:

1. Similarity: Cosine, RBF and Covariance;
2. Sparsity: Different values of k-NN;
3. Symmetrize the graph;
4. Normalization: Yes or no.

Prior-based baselines

Baselines based on a prior of a signal property:

- Smoothness prior: Kalofolias method [Kalofolias et al 2019]
- Sparsity prior: NNK method [Shekizzar and Ortega 2020]

Results

Task 1: Unsupervised Clustering of Vertices

Method	Inference/Dataset	ESC-50	cora	flowers102
C-means		0.59	0.10	0.36
Spectral clustering	Naive	**0.66**	**0.34**	**0.45**
	NNK	**0.66**	**0.34**	0.44
	Kalofolias	0.65	0.27	0.44

Results

Task 2: Semi-Supervised Classification of Vertices - labels only

Method	Inference/Dataset	ESC-50	cora	flowers102
Logistic Regression		52.92% $\pm$1.9	46.84% $\pm$1.6	33.51% $\pm$1.7
Label Propagation	Naive	59.05% $\pm$1.8	**58.86%** $\pm$2.9	36.73% $\pm$1.6
	NNK	57.44% $\pm$2.2	58.66% $\pm$2.9	33.57% $\pm$1.6
	Kalofolias	**59.16%** $\pm$1.8	58.60% $\pm$3.4	**37.01%** $\pm$1.7

Results

Task 2: Semi-Supervised Classification of Vertices - labels and features

Method	Inference/Dataset	ESC-50	cora	flowers102
Logistic Regression		52.92% ±1.9	46.84% ±1.6	33.51% ±1.7
SGC	Naive	60.48% ±2.0	**67.19%** ±1.5	**37.73%** ±1.5
	NNK	**61.38%** ±2.0	66.58% ±1.5	36.81% ±1.5
	Kalofolias	59.36% ±2.0	66.28% ±1.5	37.5% ±1.5

Results

Task 3: Denoising of Graph Signals

	Road graph	Kalofolias	RBF NNK	RBF k-NN
Best SNR	10.32	**10.41**	9.99	9.80

Initial findings

1. **Graph-based methods:** Outperform the non-graph baselines;
2. **Similarity choice:** When possible use the cosine similarity;
3. **Sparsity**: Method and dataset dependant. NNK and Kalofolias are more stable;
4. **Normalization:** As expected, normalized graphs perform better;
5. **Naive baselines vs. optimization approaches:** No clear winner, NNK and Kalofolias are less hyperparameter dependant;
6. **Hard task:** sparse graphs in semi-supervised random splits.

Outline

Conclusion

Summary

- Graphs -> natural way to encode relational data;
- Not always available -> **graph inference** is needed;
- Evaluating graph inference is hard -> we introduce a benchmark;
- Easily accessible online:
  - `https://github.com/cadurosar/benchmark_graphinference`;
- Tests with baselines -> encouraging findings.

Future work

- Extend to additional graph-related tasks;
- Include more task-agnostic methods;
- Improve task-specific evaluation.

Conclusion

Summary

- Graphs -> natural way to encode relational data;
- Not always available -> **graph inference** is needed;
- Evaluating graph inference is hard -> we introduce a benchmark;
- Easily accessible online:
  - `https://github.com/cadurosar/benchmark_graphinference`;
- Tests with baselines -> encouraging findings.

Future work

- Extend to additional graph-related tasks;
- Include more task-agnostic methods;
- Improve task-specific evaluation.

Thank you for your attention

Benchmarks available at:

https://github.com/cadurosar/benchmark_graphinference

References

- Nilsback and Zisserman 2008 “Collective classification in network data” AI Magazine;
- Sen et al 2008 “Automated flower classification over a large number of classes” ICVGIP;
- Piczak 2015 “ESC: Dataset for Environmental Sound Classification” ACM on Multimedia;
- Irion and Saito 2016 “Efficient approximation and denoising of graph signals using the multiscale basis dictionaries” IEEE TSIPN;
- Szegedy et al 2016 “Rethinking the inception architecture for computer vision” CVPR;
- Pasdeloup et al 2017 “Characterization and inference of graph diffusion processes from observations of stationary signals” IEEE TSIPN;
- Egilmez et al 2017 “Graph learning from data under Laplacian and structural constraints” IEEE JSTSP;
- Monti et al 2017 “Geometric deep learning on graphs and manifolds using mixture model CNNs” CVPR;
- Kumar et al 2018 “Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes” ICASSP;
- Kalofolias et al 2019 “Large scale graph learning from smooth signals” ICLR;
- Shekizzar and Ortega 2020 “Graph construction from data using non negative kernel regression (NNK graphs)” ICASSP;