Aalto University
School of Science
and Technology

A Comparative Analysis of Graph Signal Recovery Methods for Big Data Networks

Alexandru Cristian Mara

Department of Computer Science
Aalto University, School of Science
alexandru.mara@aalto.fi

Version 1.0, August 9, 2017

Overview

Research Questions

- Main research question:
  - Which is the most accurate, scalable and robust graph signal recovery (GSR) algorithm?
- Secondary research questions:
  - How and to what extent does the graph structure such as edge weights, clustering coefficient or average degree affect GSR algorithms?
  - How does noise affect the recovery process of different methods?
  - Does the sampling set selection strategy affect the recovery algorithms?
  - Are there any significant differences between the behaviour of the methods on real and synthetic datasets?

Thesis Contributions

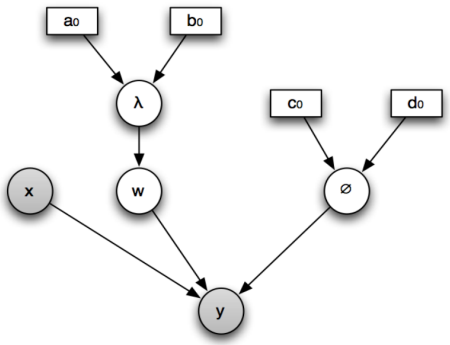
- The main contributions of this thesis are:
  - First review and in depth comparison of the most prominent GSR methods on both real and synthetic data.
  - Guidelines to select the most adequate recovery method based on the graph structure are provided.
  - Scalable message passing implementations in GRAPHX of all algorithms are made publicly available.

Big Data

- In most applications we have:
  - Huge amounts of data
  - Heterogeneous nature (audio, text, video, . . .)
  - Mostly unlabelled
- Labelling:
  - Done by human experts with domain knowledge
  - Very expensive
  - Only for small fractions of datasets
- Main goal in ML:
  - Automatically label entire datasets starting from small subsets of initially known labels

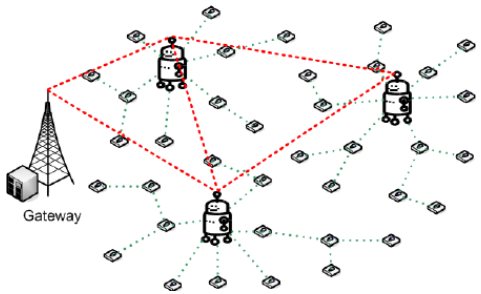
Big Data Over Networks

- Most Datasets present some intrinsic graph structure
  - Nodes represent data points
  - Edges connect similar or related nodes
- The network structure can arise from:
  - Statistical correlations
  - Physical proximity
  - A combination of both



Statistical graph

Flight network



Sensor network

Graph-based models for big data

- A dataset $\mathcal{D}$ can be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$
- Each node $i \in \mathcal{V}$ represents an individual data point $z_i \in \mathcal{D}$
- Labels $x[\cdot]$ present additional characteristics of the nodes
- Edges $\{i, j\} \in \mathcal{E}$ represent similarity between data points
- Weights $W_{i,j}$ quantify the strength of edge connections

- Advantages of graph representation:
  - Can cope well with heterogeneous data:
    Data points $z_i \in \mathcal{D}$ can represent any type of information
    Only a week notion of similarity is needed
  - Many efficient graph signal processing tools available
  - Scalability:
    Efficient message passing formulations
    Graphs take computation to data

Graph Semi-Supervised Learning

- Assume a Facebook graph $\mathcal{G}$ where nodes $i \in \mathcal{V}$ are users and edges $\{i, j\} \in \mathcal{E}$ represent friendship relations.
- Users are associated with labels $x[i]$ indicating their political view:
  - $x[i] = 0$ if user i is conservative
  - $x[i] = 1$ if user i is liberal
- Initially we know the labels x_i (political views) of few data points (users) $i \in \mathcal{M}$
- The objective is to discover the political views of all users $i \in \mathcal{V}$ using the graph structure and the set $\mathcal{M}$ of initially known labels

Graph Signal Recovery Problem

- Labels $x[\cdot]$ and $\hat{x}[\cdot]$ denote the real and recovered graph signals
- In order to recover the whole signal the smoothness hypothesis of SSL is needed:
  - Well connected nodes i, j have similar signals $x[i], x[j]$
  - The graph signal varies little over well connected subsets of nodes (clusters)
- Many smoothness measures $\mathcal{R}(\hat{x}[\cdot])$ can be defined
- The empirical error $E(\hat{x}[\cdot])$ between the initial labels (x_i) and predicted $(\hat{x}[i])$ can be computed via different norms
- The label recovery problem becomes a minimisation task:

$$\hat{x} \in \underset{\hat{x}[\cdot] \in \mathbb{R}^{\mathcal{V}}}{\arg\min}\, E(\hat{x}[\cdot]) + \lambda \mathcal{R}(\hat{x}[\cdot])$$

Average Consensus (Avg_cons)

- Used as a baseline to compare the rest of the methods
- No smoothness measure $\mathcal{R}(\hat{x}[\cdot])$ is used
- Empirical error is defined as:

$$E(\hat{x}[\cdot]) := \sum_{i \in \mathcal{M}} ||\hat{x}[i] - x_i||_2^2$$

- The resulting minimization problem is:

$$\hat{x} \in \arg\min_{\hat{x}[\cdot] \in \mathbb{R}^{\mathcal{V}}} \sum_{i \in \mathcal{M}} ||\hat{x}[i] - x_i||_2^2$$

Community-based Labelling (LP_cd)

- Recovery based on Label Propagation for community detection (LP_cd)
- Graph is partitioned in a set of clusters $\mathcal{F} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$ through LP_cd
- For each cluster $\mathcal{C}_l \in \mathcal{F}$ a consensus signal value $\hat{x}_l^{cons}$ is selected for all nodes $i \in \mathcal{C}_l$ as:

$$\hat{x}_l^{cons} = mode(x_i) \text{ for all } i \in \mathcal{C}_l \cap \mathcal{M}$$

Label Propagation (LP_sr)

- Label Propagation for signal recovery by Zhou et. al.
- Solved via power iteration method
- Quadratic form used as smoothness measure $\mathcal{R}(\hat{x}[\cdot])$:

$$\mathcal{R}(\hat{x}[\cdot]) := \sum_{\{i,j\}\in\mathcal{E}} W_{i,j}||\hat{x}[i] - \hat{x}[j]||_2^2$$

- Empirical error is forced to be $E(\hat{x}[\cdot]) = 0$ i.e.:

$$\hat{x}[i] = x_i \text{ for all } i \in \mathcal{M}$$

- The resulting minimization problem is:

$$\hat{x} \in \arg\min_{\hat{x}[\cdot]\in\mathbb{R}^{\mathcal{V}}} \sum_{\{i,j\}\in\mathcal{E}} W_{i,j}||\hat{x}[i] - \hat{x}[j]||_2^2 \text{ s.t. } \hat{x}[i] = x_i \text{ for all } i \in \mathcal{M}$$

Sparse Label Propagation (SLP)

- Sparse label propagation by Jung et. al.
- Solved via the Pock-Chambolle primal-dual method
- Total variation used as smoothness measure $\mathcal{R}(\hat{x}[\cdot])$:

$$\mathcal{R}(\hat{x}[\cdot]) := \sum_{\{i,j\}\in\mathcal{E}} W_{i,j}||\hat{x}[i] - \hat{x}[j]||_1$$

- Empirical error is forced to be $E(\hat{x}[\cdot]) = 0$ i.e.:

$$\hat{x}[i] = x_i \text{ for all } i \in \mathcal{M}$$

- The resulting minimization problem is:

$$\hat{x} \in \underset{\hat{x}[\cdot]\in\mathbb{R}^{\mathcal{V}}}{\arg\min} \sum_{\{i,j\}\in\mathcal{E}} W_{i,j}||\hat{x}[i] - \hat{x}[j]||_1 \text{ s.t. } \hat{x}[i] = x_i \text{ for all } i \in \mathcal{M}$$

Netowrk Lasso (nLasso)

- Adaptation of nLasso by Hallac et. al. for signal recovery
- Solved via ADMM
- Total variation used as smoothness measure $\mathcal{R}(x[\cdot])$:

$$\mathcal{R}(\hat{x}[\cdot]) := \sum_{\{i,j\}\in\mathcal{E}} W_{i,j}||\hat{x}[i] - \hat{x}[j]||_2$$

- Empirical error defined as:

$$E(\hat{x}[\cdot]) := \sum_{i\in\mathcal{M}} ||\hat{x}[i] - x_i||_1$$

- The resulting minimization problem is:

$$\hat{x} \in \arg\min_{\hat{x}[\cdot]\in\mathbb{R}^{\mathcal{V}}} \sum_{i\in\mathcal{M}} ||\hat{x}[i] - x_i||_1 + \lambda \sum_{\{i,j\}\in\mathcal{E}} W_{i,j}||\hat{x}[i] - \hat{x}[j]||_2$$

Datasets

- Chain graphs:
  - Synthetic chain graph with a constant square wave pattern
  - Electricity consumption dataset
- Grid graphs:
  - Synthetic grid graph with a check board pattern
  - Flower image data (foreground/background segmentation)
- Power law graphs:
  - Synthetic LFR graph (high and low clust. coef.)
  - Amazon product co-purchase
  - 3D road network dataset (disconnected graph)

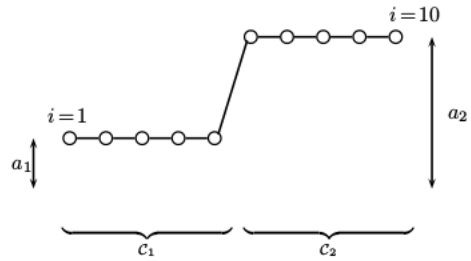


Fig. 1. A clustered graph signal (cf. (5)) on a chain graph which is partitioned into two clusters.

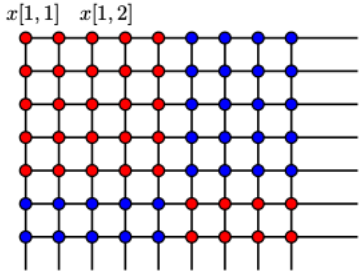
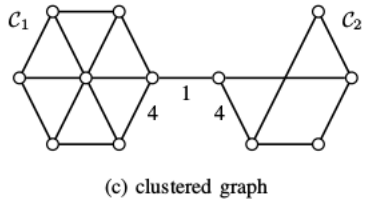


Fig. 2. A clustered graph signal defined over a grid graph.



(c) clustered graph

Fig. 3. A power law graph organized in clusters with inter-cluster weights lower than intra-cluster weights.

Datasets

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Signals	Weights	Missing $x[\cdot]$
Synth chain	$1M$	$1M$	$\{1,5\}$	$\{1,2\}$	No
Electrcity	$2M$	$2M$	$\mathbb{R}^+$	$\{1\}$	1,25%
Synth grid	$1M$	$1.9M$	$\{1,5\}$	$\{1,2\}$	No
Image	$27K$	$538K$	$\{-1,1\}$	$\mathbb{R}^+$	No
Synth LFR-L	$100K$	$945K$	$\mathbb{R}^+$	$\mathbb{R}^+$	No
Synth LFR-H	$100K$	$949K$	$\mathbb{R}^+$	$\mathbb{R}^+$	No
Amazon	$524K$	$1.7M$	$\frac{1}{2}\{0,2,3..,10\}$	$\{1\}$	40%
3D road map	$397K$	$377K$	$\mathbb{R}$	$\mathbb{R}^+$	No

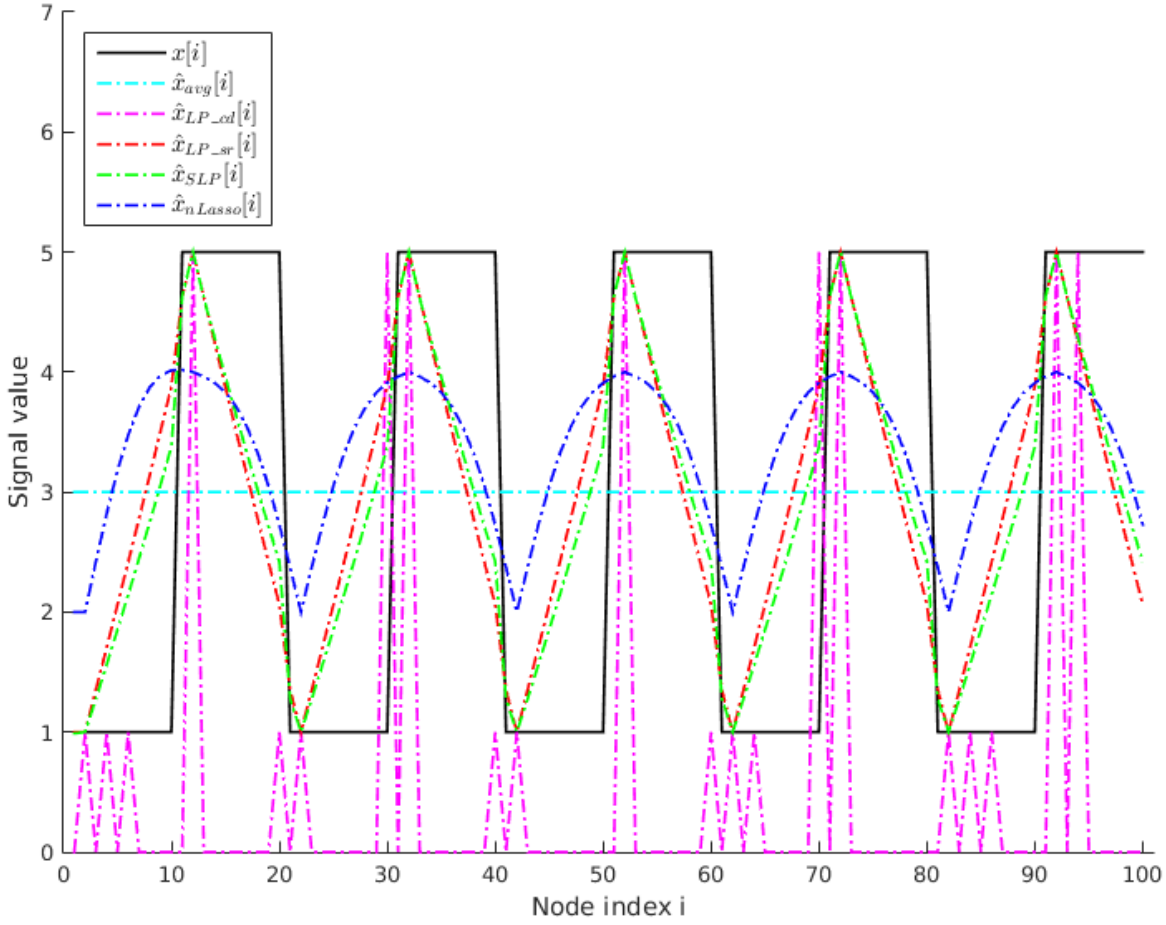
Table : Data graphs and main features.

Evaluation Measures

- Accuracy
  - Measured in terms of NMSE i.e. $\varepsilon := \|\hat{x}[\cdot] - x[\cdot]\|_2^2 / \|x[\cdot]\|_2^2$
  - We compare the accuracy on real and synthetic data
- Sampling sets
  - We test the algorithm on varying sizes of the sampling set, 10%-80% of total nodes
  - We compare random sample selection to cluster based
- Scalability
  - Measured in terms of execution time (sec.)
  - We study the scalability on different cluster worker configurations (1-8 workers)
- Weight effect
  - We compare the algorithms on weighted and unweighted graphs
- Noise robustness
  - We compare the algorithms with and without random Gaussian noise added to the graph signals

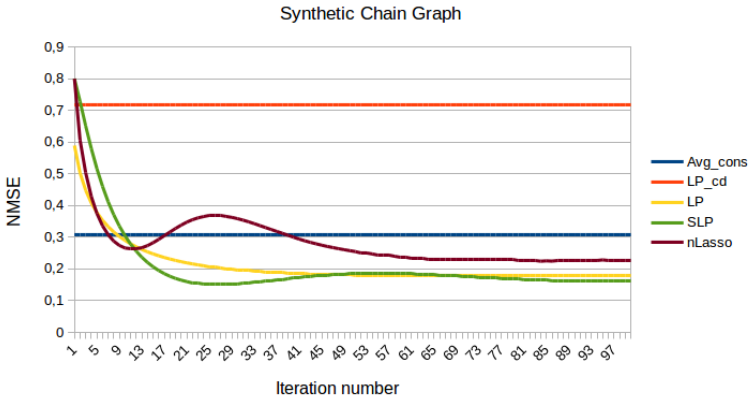
Chain Graphs

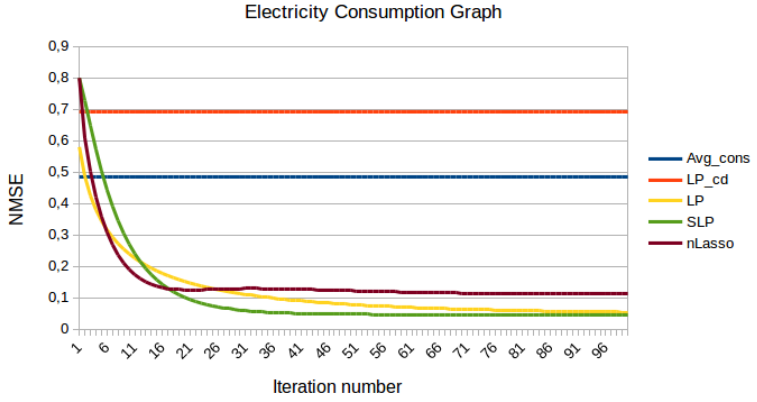
- Recovered labels:

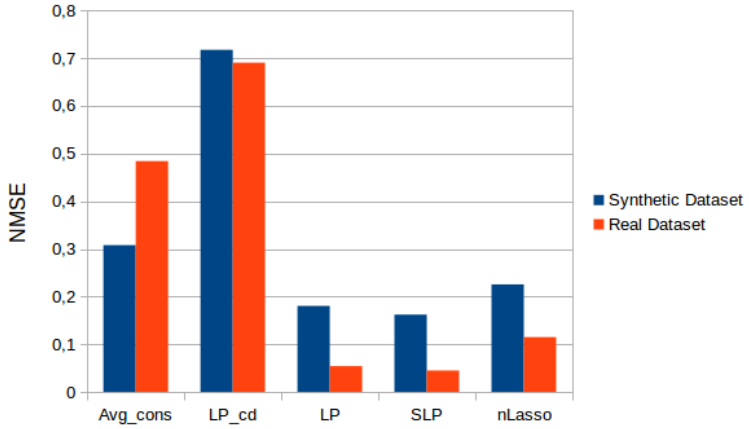


Chain Graphs

- Accuracy:

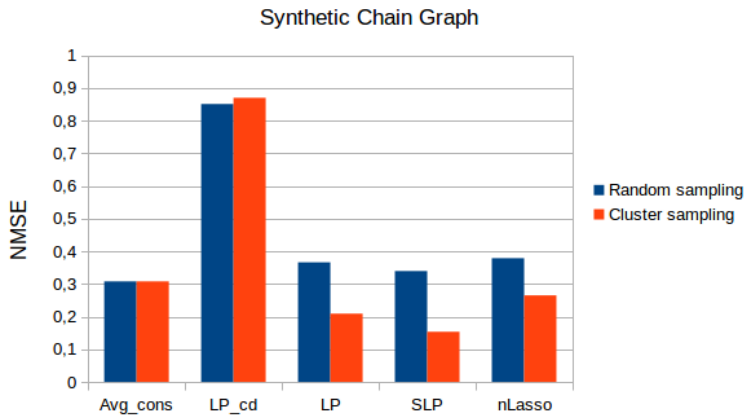


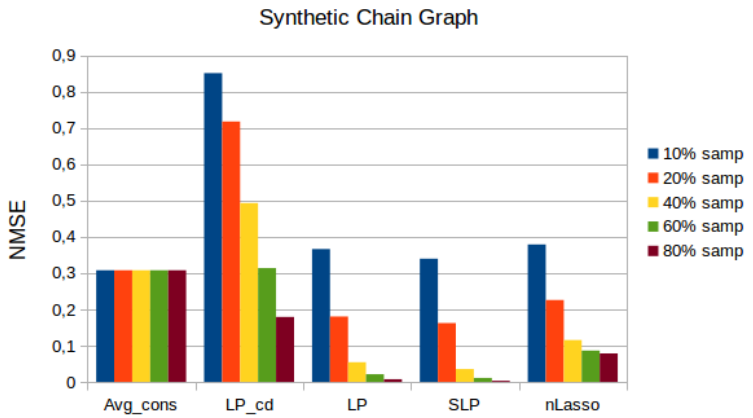




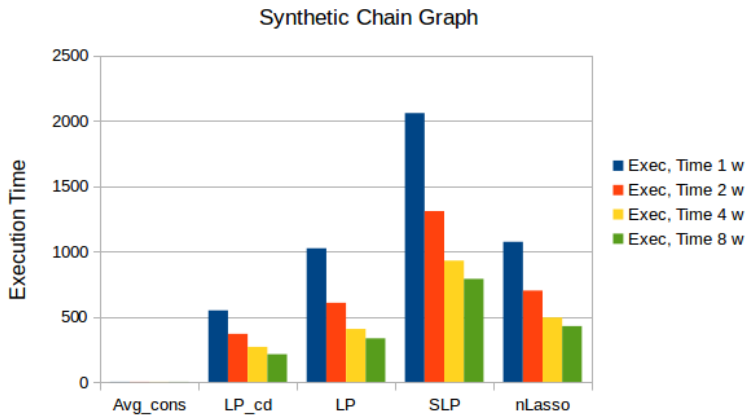
Chain Graphs

- Sampling sets:



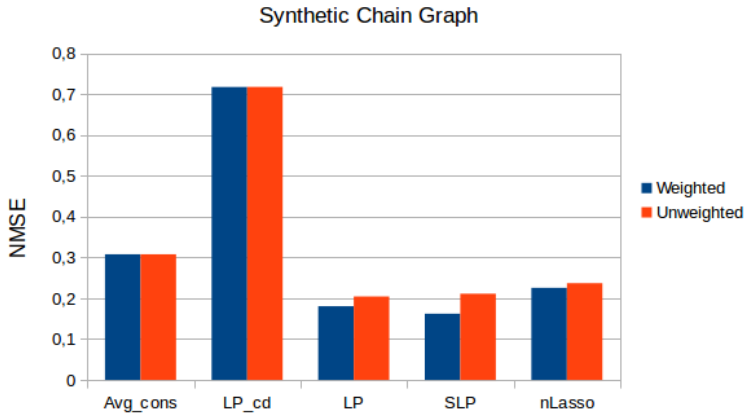


- Scalability:

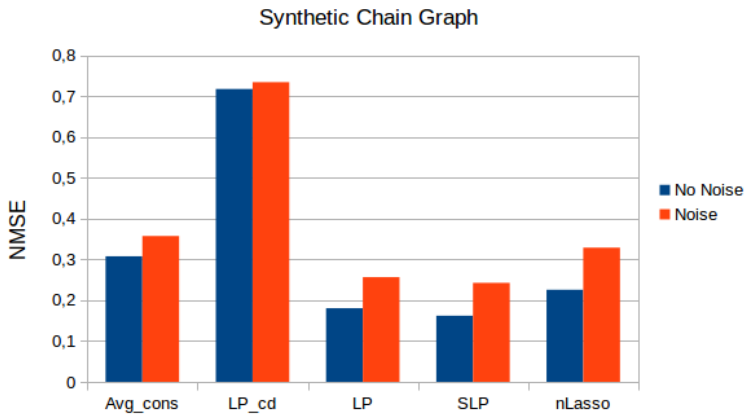


Chain Graphs

- Weight effect:



- Noise:



Grid Graphs

- Real labels and sampling set:

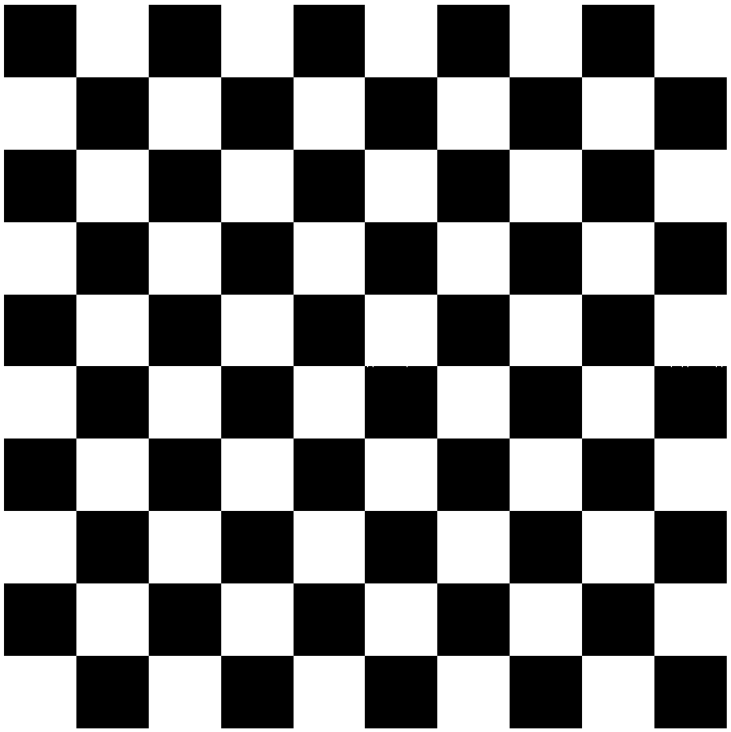


Figure : Real labels

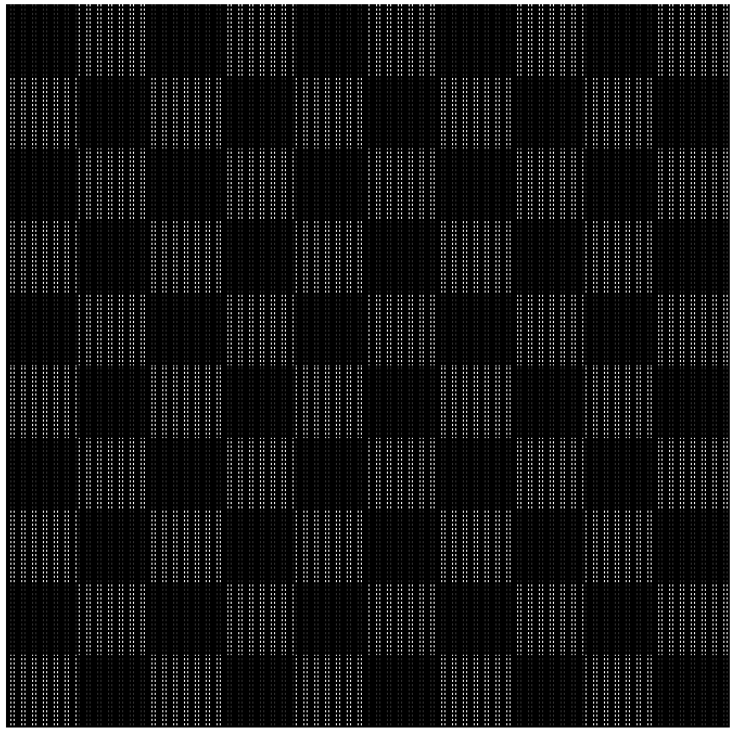


Figure : Sampled nodes

Grid Graphs

- Recovered labels:

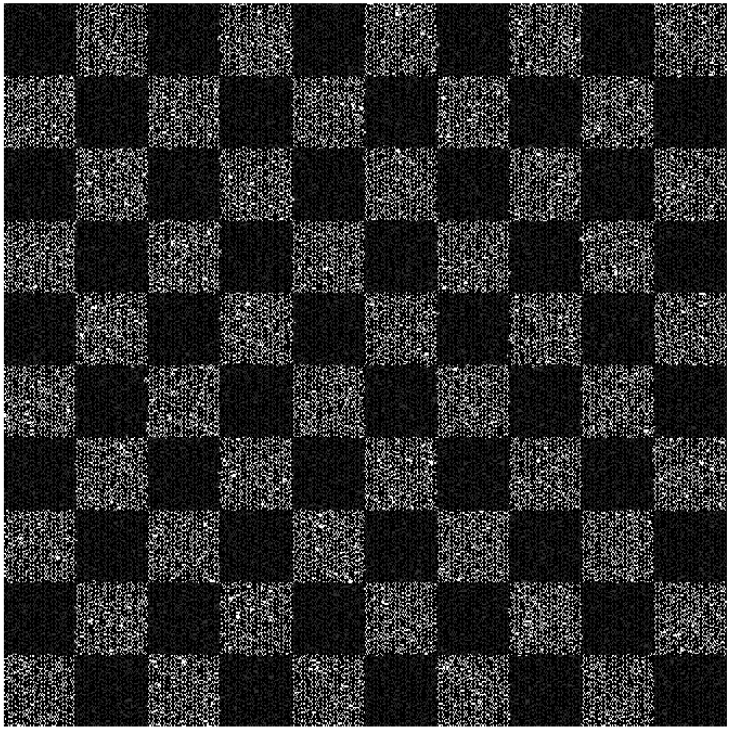


Figure : Label Propagation for CD

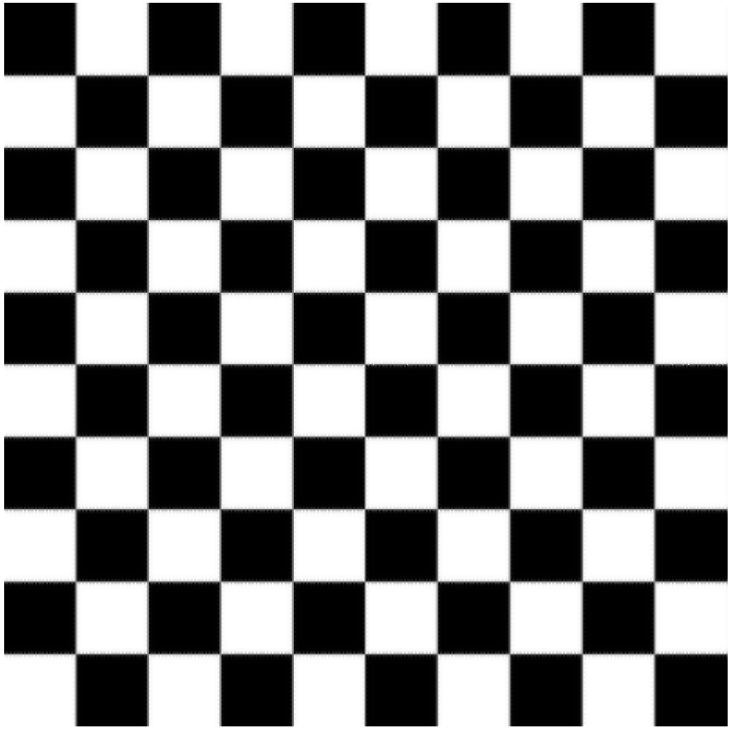


Figure : Label Propagation for SR

Grid Graphs

- Recovered Labels:

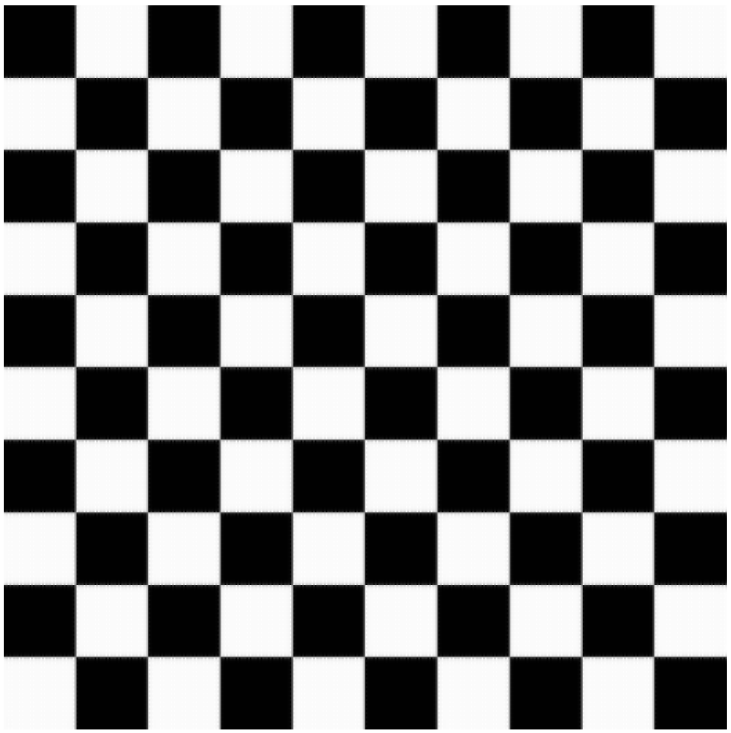


Figure : Sparse Label Propagation

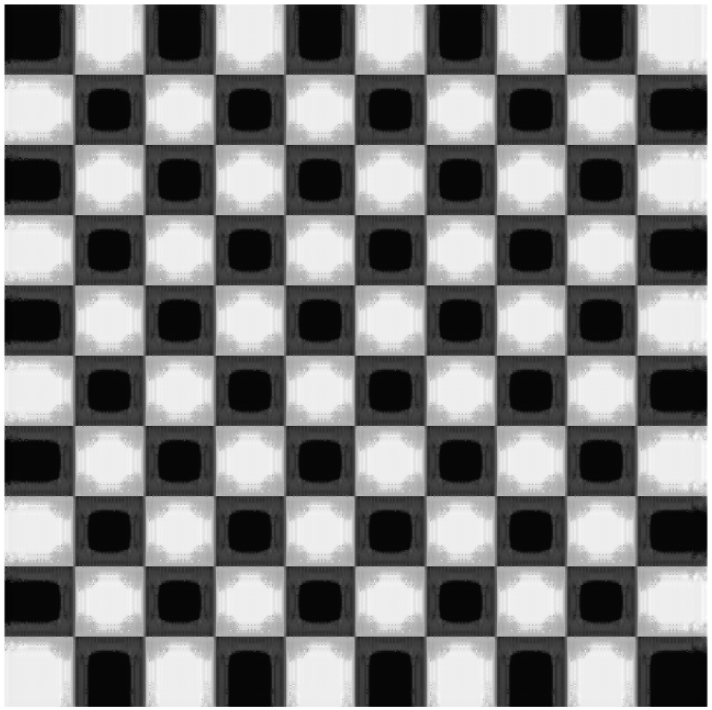
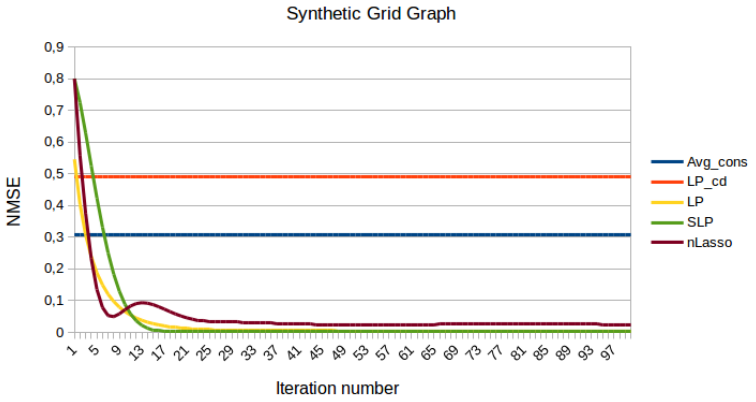
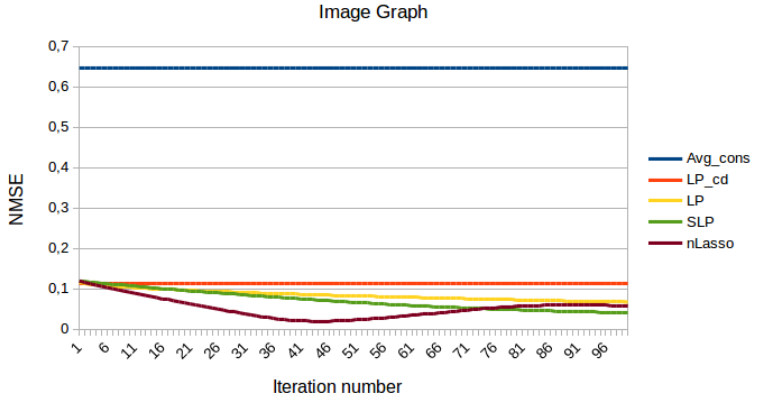


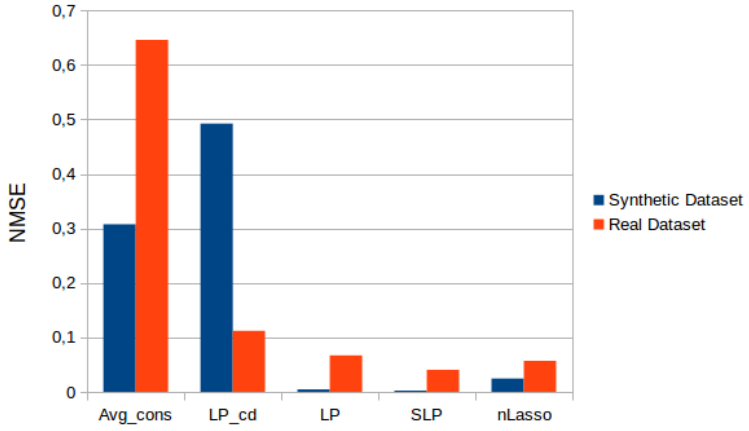
Figure : Network Lasso

Grid Graphs

- Accuracy:

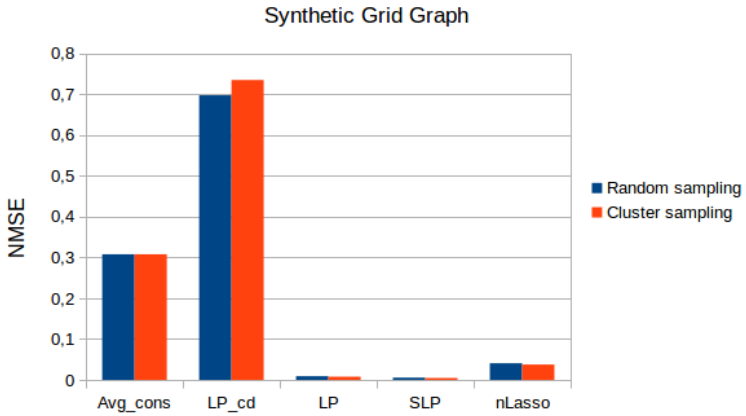


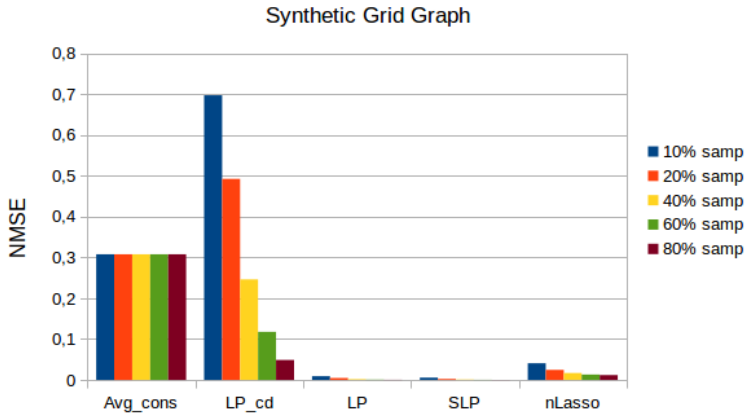




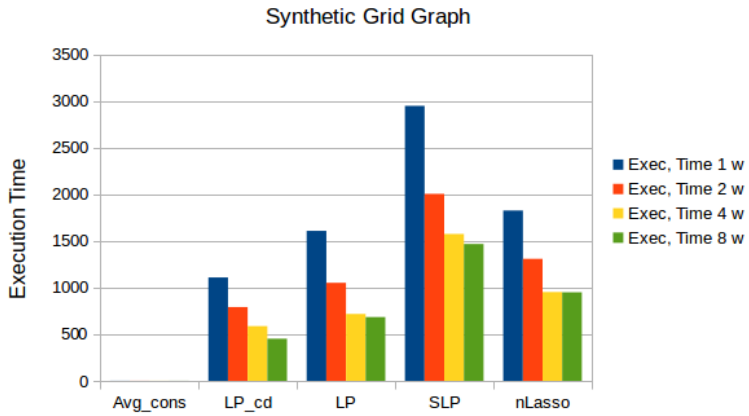
Grid Graphs

- Sampling sets:



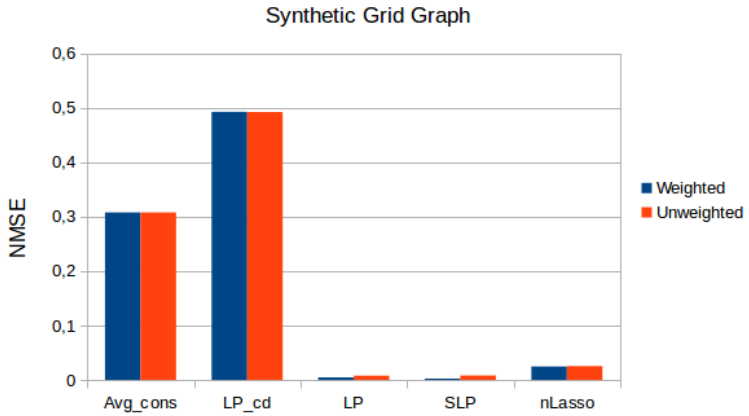


- Scalability:

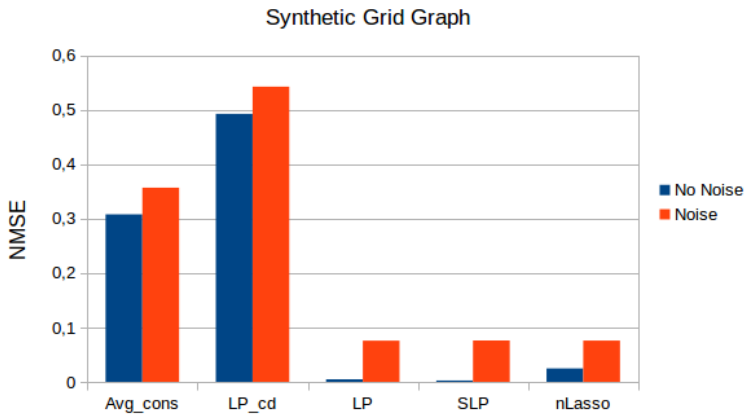


Grid Graphs

- Weight effect:

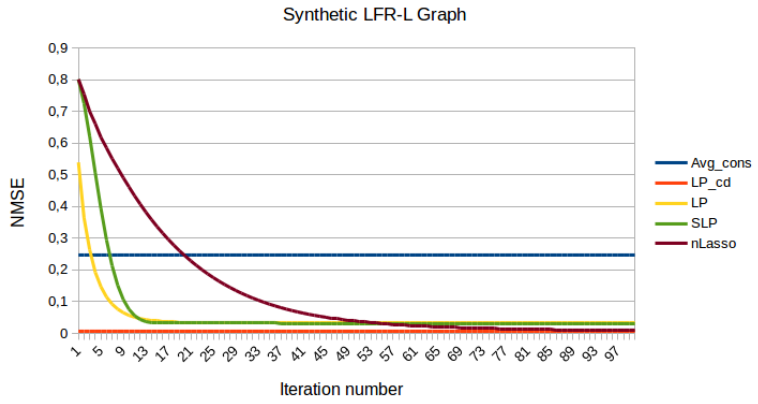


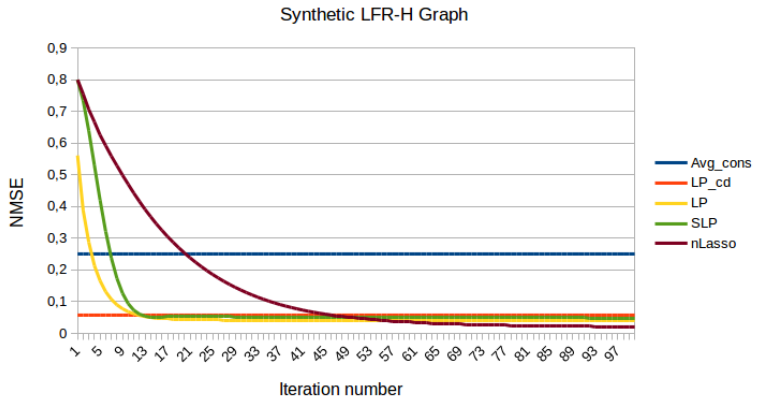
- Noise:

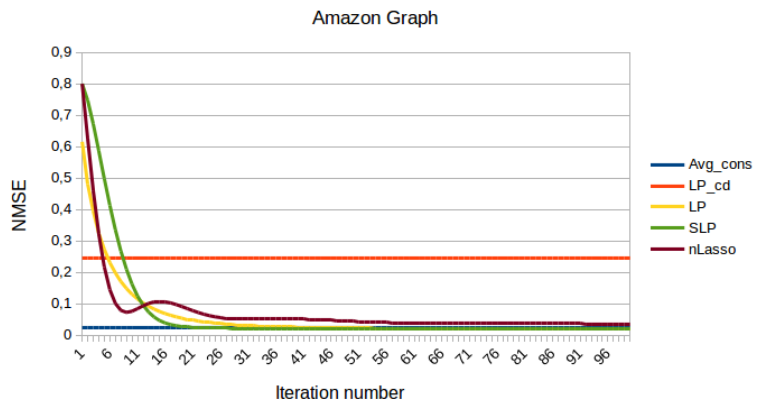


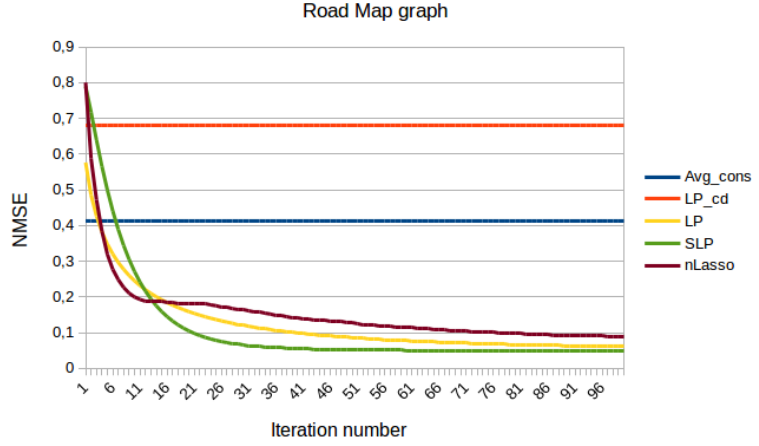
Power Law Graphs

- Accuracy:



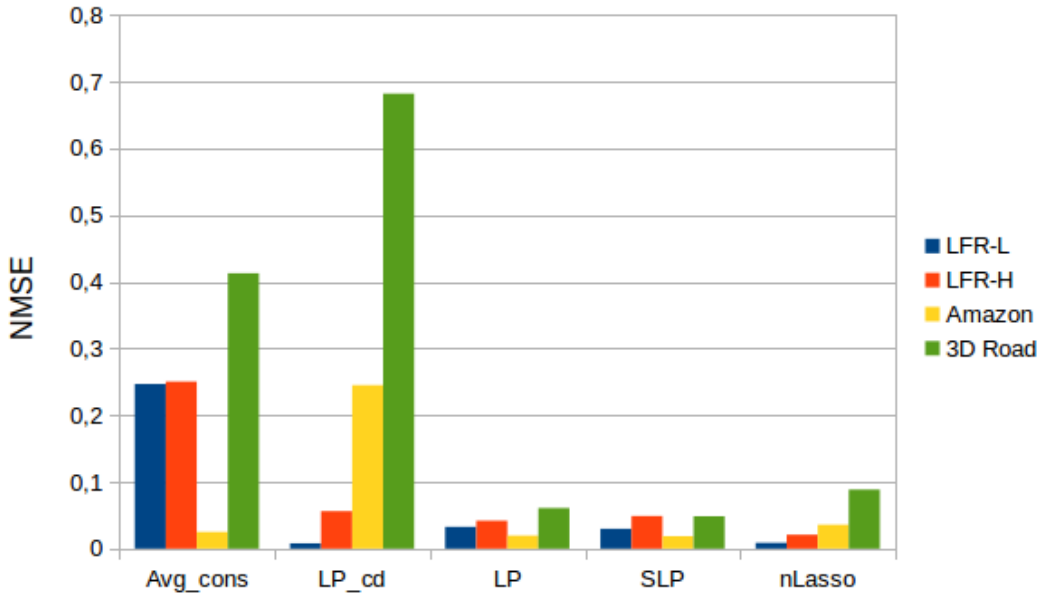






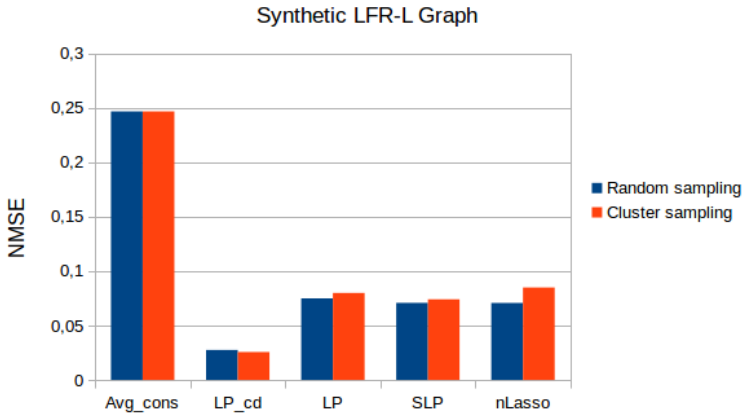
Power Law Graphs

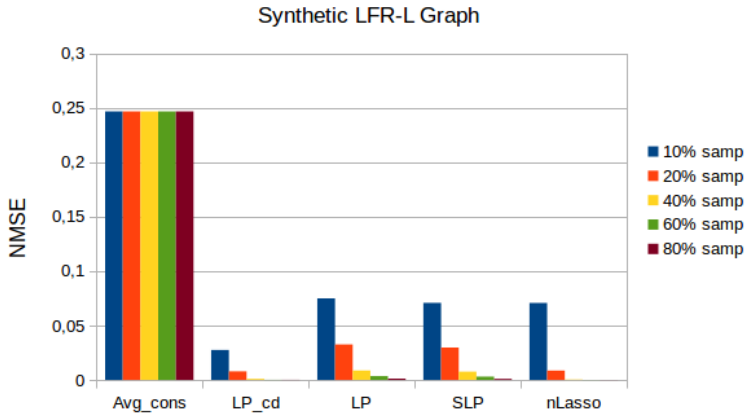
- Accuracy:



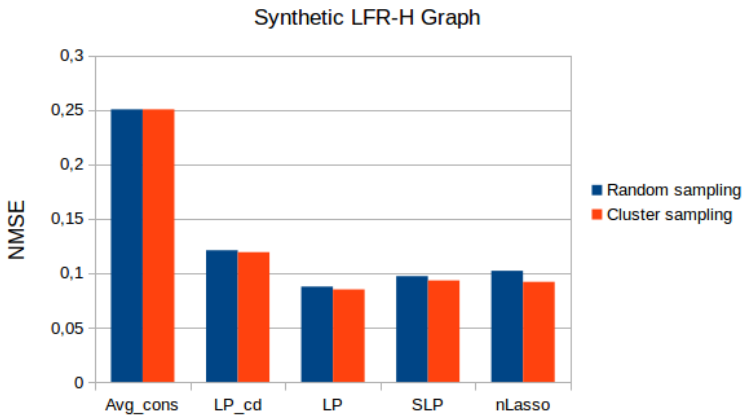
Power Law Graphs

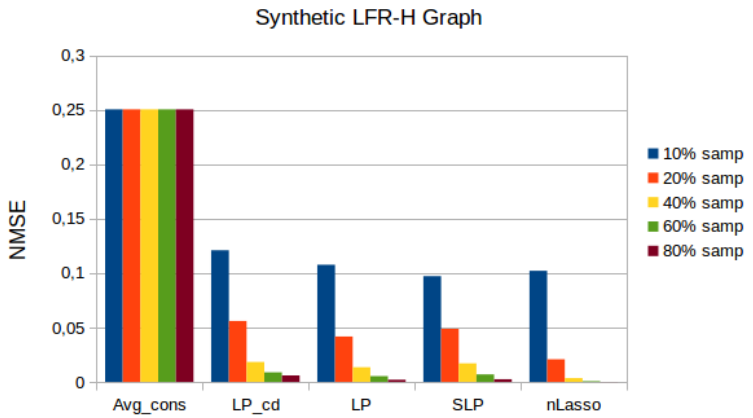
- Sampling sets (LFR-L):





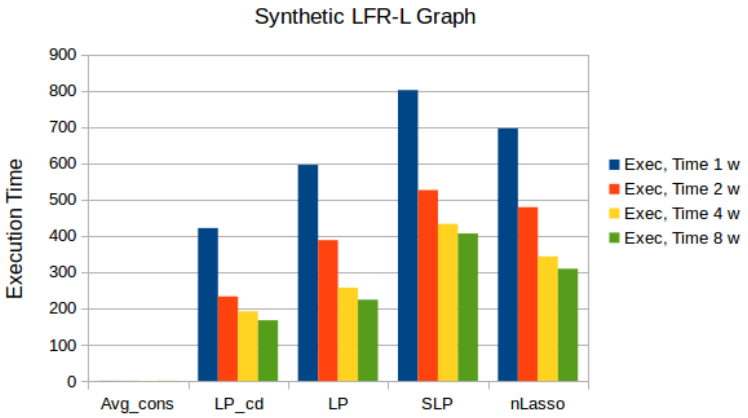
- Sampling sets (LFR-H):

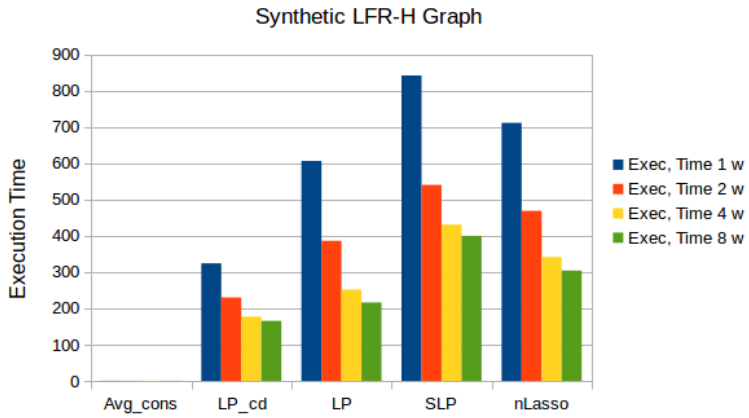




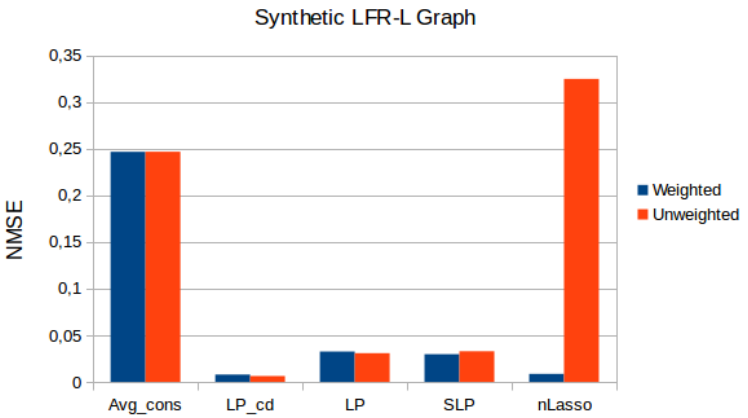
Power Law Graphs

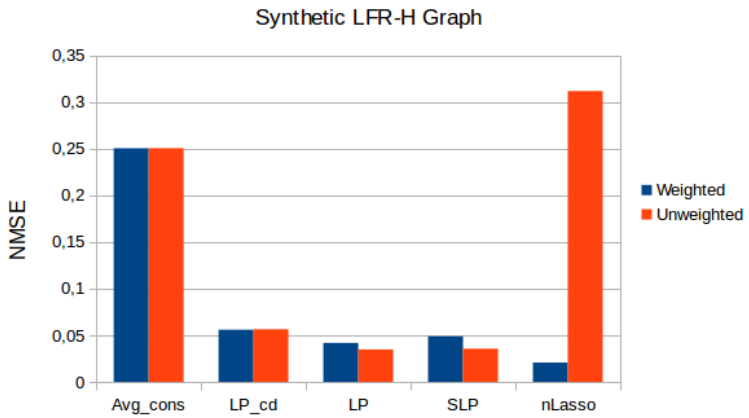
- Scalability (LFR-L/LFR-H):





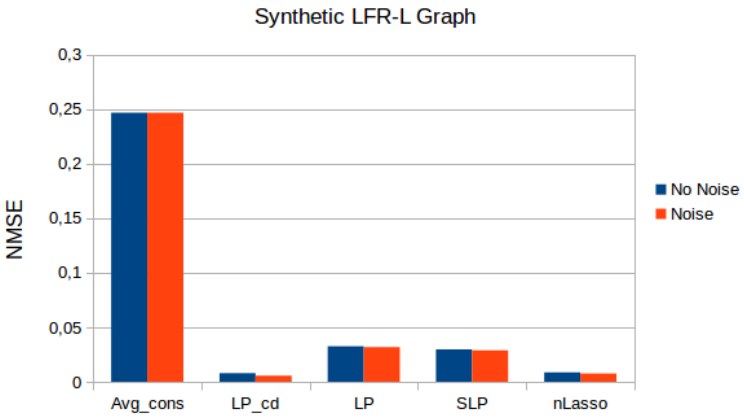
- Weight effect (LFR-L/LFR-H):

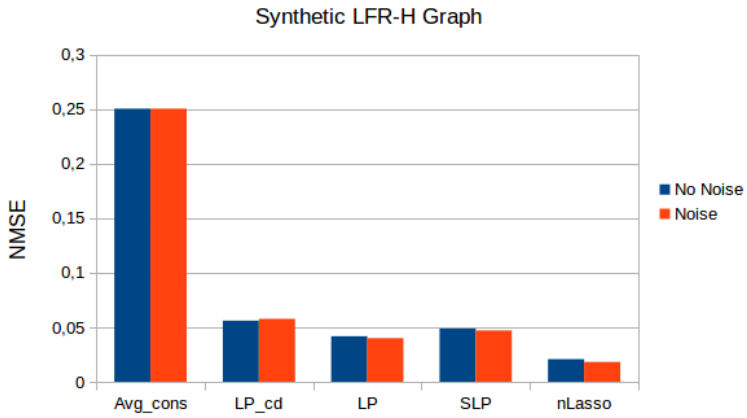




Power Law Graphs

- Noise (LFR-L/LFR-H):





Conclusions

- Accuracy:
  - On chain and grid graphs SLP is the best performing algorithm
  - On power law graphs SLP and nLasso outperforms all other methods
  - The convergence of the algorithms is quite similar (aprox. 50 iterations). On the LFR graph nLasso converges the slowest but achieves the best accuracy.
  - The accuracy on real datasets is in general lower than on the synthetic ones due to the lower clustering coefficient
- Sampling sets:
  - Increasing the sampling set size, as expected, increases accuracy
  - nLasso and SLP are the algorithms that obtains the most benefit out of increasing sampling set size

Conclusions

- Scalability:
  - The LP for community detection algorithm is obtained from the GRAPHX library
  - All iterative algorithms follow the same proportional decreases in execution times as LP_cd when the number of workers increases
- Weight effect:
  - Removing the graph weights affects LP, SLP, nLasso
  - The effect is much less clear on power law graphs. This is due to the signal being much higher than the edge weights.
  - The most affected algorithm by missing edge weights is nLasso
- Noise effect:
  - Noise affects all algorithms in a very similar way.
  - The effects on LFR graphs are less visible due to the high signal to noise difference.

Conclusions

- Due to the use of TV as smoothness measure, SLP and nLasso recover better clustered signals where the difference between clusters is big. For this reason nLasso performs very well on the power law graphs.
- LP performs well in all cases but returns a very smooth signal.
- LP_cd is the adequate method if the graphs is strongly clustered with high average degree and the smoothness hypothesis is closely followed by the dataset.

Future Work

- Include in the evaluation other algorithms such as Label Spreading by Zhou et. al. or Label Propagation using Jacobi iteration algorithm.
- Study the behaviour of algorithms such as nLasso and SLP with tuned parameters.
- Extend experiments related to sampling set selection strategies. Include more refined methods such as the one recently presented by Jung et. al. in *When is network Lasso accurate*.
- Test the scalability of the methods on a cluster of machines.

Questions