Graph Learning Based on Total Variation Minimization

Peter Berger, Manfred Buchacher, Gabor Hannak, and Gerald Matz

Institute of Telecommunications
Vienna University of Technology, Austria

Introduction

Motivation

- Graph learning is key for clustering and semi-supervised learning
- Nearest neighbor graphs are the most widely used
- Recently graph learning from smooth signals
- Goal: clear and easily accessible graph learning algorithm

Contributions

- Novel algorithm: We formulate the graph learning problem as a constraint quadratic program in the graph's edge weights
- We quantify signal smoothness in terms of total variation
- For noisy data we combine graph learning with total variation denoising
- All parameters in our scheme have a natural interpretation
- We demonstrate in numerical experiments that our learning algorithm is well suited for cluster/community detection

Noise-free Graph Learning

- **Given data:** graph signal vectors $\mathbf{x}^m \in \mathbb{R}^N$, $m = 1, \dots, M$
- **Graph:** described by edge weights W_{ij}, $1 \leq i, j \leq N$
- **Signal smoothness:** quantified in terms of total variation

$$\|\mathbf{x}_m\|_{\mathrm{TV}} = \sum_{i=1}^{N} \sum_{j=1}^{N} |x_{m,i} - x_{m,j}| W_{ij}$$

- Construct edge weights W_{ij} with small value

$$\sum_{m=1}^{M} \|\mathbf{x}_m\|_{\mathrm{TV}} = \mathrm{tr}\{\mathbf{W}\mathbf{D}\},$$

with discrepancy matrix $\mathbf{D}(\mathbf{X}) \in \mathbb{R}^{N \times N}$

$$D_{ij} = \sum_{m=1}^{M} |x_{m,i} - x_{m,j}|$$

- **Basic optimization problem:**

$$\begin{aligned} \min_{\mathbf{W}} \quad & \mathrm{tr}\{\mathbf{W}\mathbf{D}(\mathbf{X})\} + \frac{\beta}{2}\|\mathbf{W}\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & \mathbf{W} = \mathbf{W}^T \geq \mathbf{0}, \\ & \mathrm{diag}(\mathbf{W}) = \mathbf{0}, \\ & \|\mathbf{W}\|_1 = 2N \end{aligned}$$

- **Solution:**

$$W_{ij} = \frac{1}{\beta}(\nu - D_{ij})_+$$

with ν determined by

$$\sum_j \sum_j (\nu - D_{ij})_+ = \beta\, 2N$$

- **Sparsity:** β controls the number of edges of the graph

Weight and Degree Constraints:

- We can incorporate node degree constraints $\mathbf{a} \leq \mathbf{W}\mathbf{1} \leq \mathbf{b}$
- We can incorporate upper bounds on the weights $W_{ij} \leq c$
- $\mathbf{a} = \mathbf{b} = 2\,\mathbf{1}$ and $c = 2/L$ yields a graph where all nodes have weighted node degree 2 and at least L neighbors
- The resulting constraint quadratic program can be solved via ADMM or interior point methods

Graph Learning from Noisy Data

- **Given data:** $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$ are noisy measurements of unknown actual graph signals $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_M)$
- **Strategy:** We propose to simultaneously learn the graph and denoise the data
- **The optimization problem:**

$$\begin{aligned} \min_{\mathbf{W},\mathbf{X}} \quad & \mathrm{tr}\{\mathbf{W}\mathbf{D}(\mathbf{X})\} + \frac{\beta}{2}\|\mathbf{W}\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & \mathbf{W} = \mathbf{W}^T \geq \mathbf{0} \\ & \mathrm{diag}(\mathbf{W}) = \mathbf{0} \\ & \|\mathbf{W}\|_1 = 2N, \\ & \mathbf{a} \leq \mathbf{W}\mathbf{1} \leq \mathbf{b} \\ & \|\mathbf{x}_m - \mathbf{y}_m\|_2 \leq \varepsilon \end{aligned}$$

- **Noise level:** ε controls the amount of measurement noise

Solving the optimization problem

- The total variation term

$$\mathrm{tr}\{\mathbf{W}\mathbf{D}(\mathbf{X})\} = \sum_{m=1}^{M} \|\mathbf{x}_m\|_{\mathrm{TV}} = \sum_i \sum_j \sum_{m=1}^{M} W_{ij}\, |x_{m,i} - x_{m,j}|,$$

is not jointly convex in $\mathbf{W}$ and $\mathbf{X}$

- Calculate a local minimum by alternately performing

> – minimization w.r.t. $\mathbf{W}$: graph learning
> – minimization w.r.t. $\mathbf{X}$: signal denoising

Algorithm 1 TV-based graph learning and denoising

Input: $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$, β, $\mathbf{a}$, $\mathbf{b}$, c, ε
Initialize: $\hat{\mathbf{X}} = \mathbf{Y}$
1: **repeat**
2: compute $\mathbf{D}(\widehat{\mathbf{X}})$
3: learn the weights $\widehat{\mathbf{W}}$ on the current graph signal estimates $\widehat{\mathbf{X}}$
4: **for** $m = 1, \dots, M$ **do**
update $\hat{\mathbf{x}}_m$ by solving

$$\begin{aligned} \min_{\mathbf{x}_m} \quad & \|\mathbf{x}_m\|_{\mathrm{TV}} = \sum_{i=1}^{N} \sum_{j=1}^{N} |x_{m,i} - x_{m,j}|\, \widehat{W}_{ij} \\ \text{s.t.} \quad & \|\mathbf{x}_m - \mathbf{y}_m\|_2 \leq \varepsilon, \end{aligned}$$

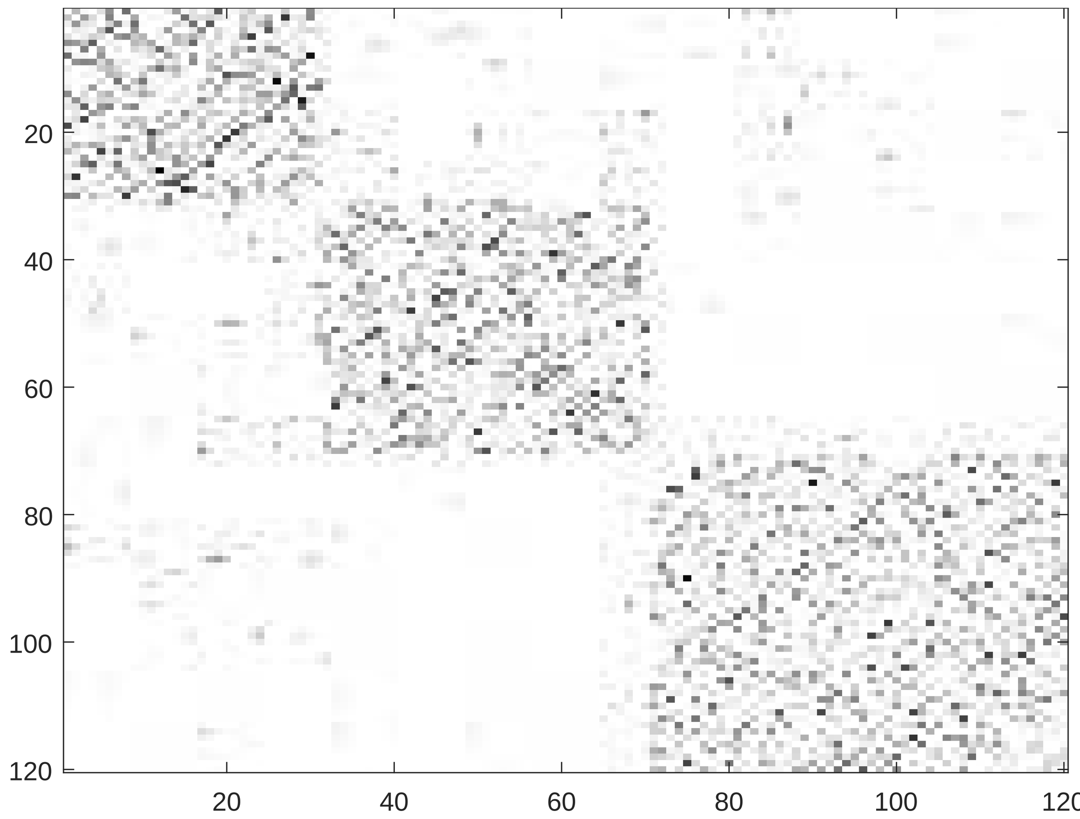
5: **end for**
6: Set $\widehat{\mathbf{X}} = (\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_M)$
7: **until** stopping criterion is satisfied
Output: $\widehat{\mathbf{W}}$, $\widehat{\mathbf{X}}$

- Signal denoising (step 4) can be solved efficiently via augmented ADMM or primal dual algorithms

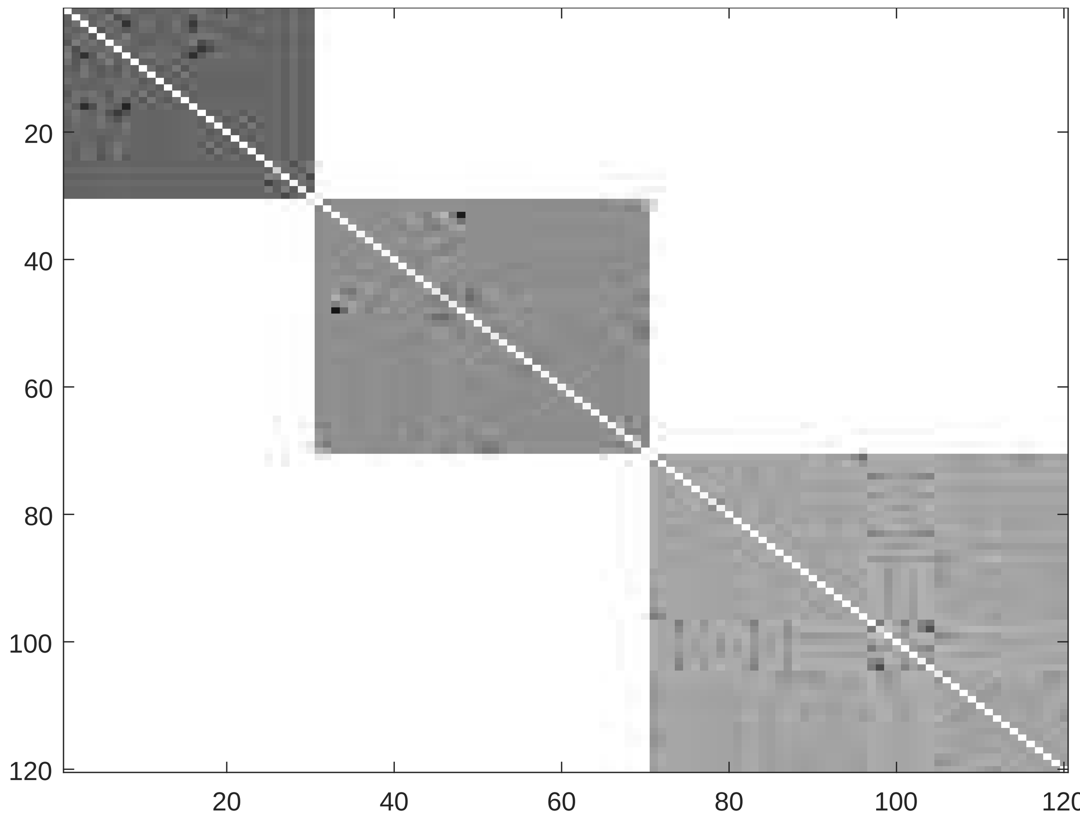
Simulation Results

- We consider a graph consisting of three clusters with 30, 40 and 50 nodes
- Nodes within the same cluster are all connected and there are no edges between different clusters
- Graph signals are piecewise constant on the clusters
- The cluster signal values are drawn from a standard Gaussian distribution
- The observed data consists of $M = 50$ graph signals that are corrupted by zero-mean Gaussian noise with average power σ^2
- The signal-to-noise ratio was $-3\,\mathrm{dB}$
- The bound on the empirical errors was set to $\varepsilon = \sqrt{N}\sigma$
- β was chosen such that in the unconstrained setup each node is guaranteed to have at least one neighbor in the learned graph
- The reconstruction performance is assessed via the F-score, the harmonic mean of edge precision and recall. The better the learned graph approximates the ground truth, the closer the F-score is to 1.
- Comparison algorithm: diagonally dominant generalized Laplacian (DDGL) by Egilmez, Pavez, Ortega 2016

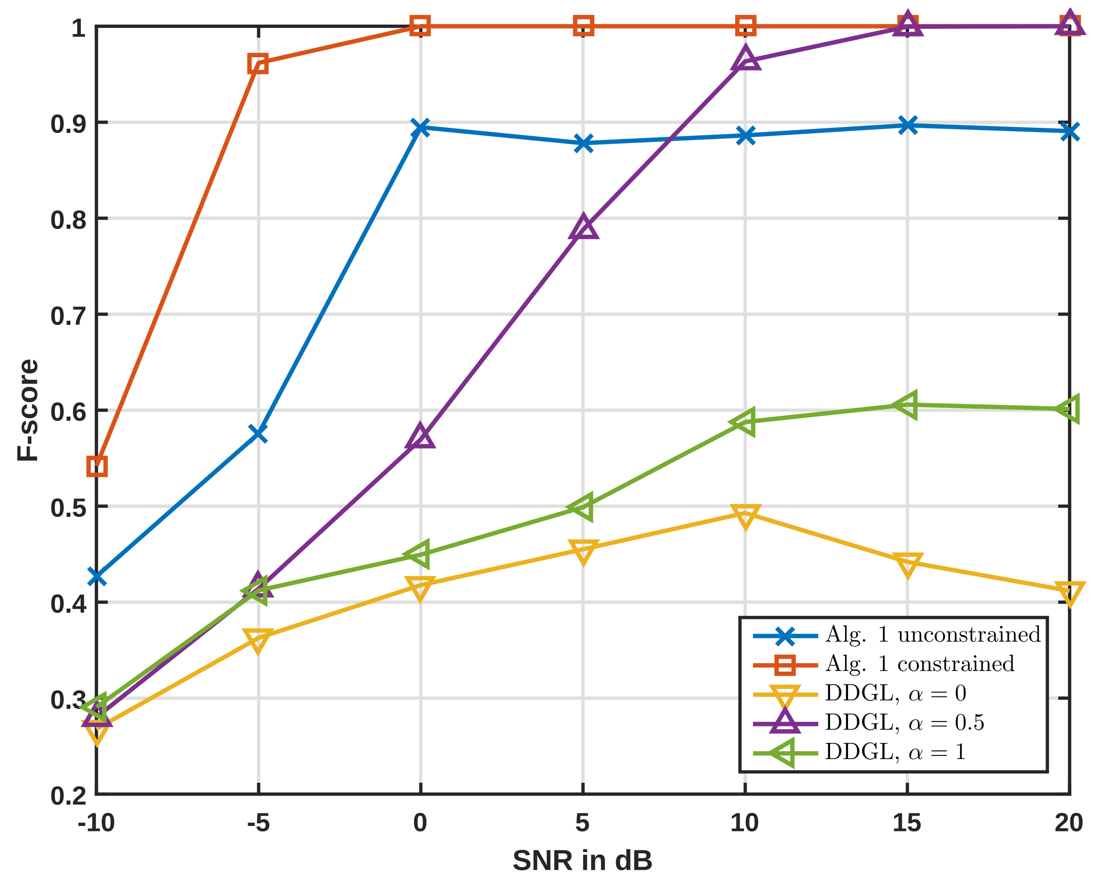
Edge weight matrix $\mathbf{W}$ with pure graph learning, $\mathbf{a} = \mathbf{b} = 2\mathbf{1}$, F-score 0.61



Edge weight matrix $\mathbf{W}$ with graph learning and denoising, $\mathbf{a} = \mathbf{b} = 2\mathbf{1}$, F-score 1



F-score vs. SNR for different learning schemes



Funded by WWTF Grant ICT15-119