SMOOTH GRAPH SIGNAL RECOVERY VIA EFFICIENT LAPLACIAN SOLVERS

Gita Babazadeh Eslamlou[1], *Alexander Jung*[2], *Norbert Goertz*[1]

[1]Institute of Telecommunications, TU Wien, Austria; {gita.babazadeh, norbert.goertz}@nt.tuwien.ac.at
[2]Dept. of Computer Science, Aalto University, Finland; alexander.jung@aalto.fi

ABSTRACT

We consider the problem of recovering a smooth graph signal from noisy samples observed at a small number of nodes. The signal recovery is formulated as a convex optimization problem using Tikhonov regularization based on the graph Laplacian quadratic form. The optimality conditions for this optimization problem form a system of linear equations involving the graph Laplacian. We solve this linear system via the iterative Gauss-Seidel method, which is shown to be particularly well-suited for smooth graph signal recovery. The effectiveness of the proposed recovery method is verified by numerical experiments using a real-world data-set.

Index Terms— graph signal processing, compressed sensing, Tikhonov-regularization, Laplacian solvers.

1. INTRODUCTION

The amounts of data collected by automated software and hardware in various domains, such as social networks, bioinformatics and wireless sensor networks, are exploding. Beside the sheer *volume* of these data-sets also the *high velocity* (rate of generation) and their *variety* (data composed of mixture of audio video text, only partially labeled) pose big challenges on their processing. A particular useful methodology for coping with big data is provided by graph signal processing (GSP), which models data-sets as signals defined over large graphs (complex networks).

GSP can be viewed as a generalization of classical signal processing [1]; the latter is obtained from GSP for the special case of a chain graph (representing the sequence of time instants). The usage of graph models within GSP entails efficient distributed message passing algorithms that are well suited to deal with large volumes of high-speed data. Moreover, graphs allow to organize heterogeneous data by exploiting application specific notions of similarity, thereby addressing the variety of big data.

A key problem studied in GSP is the recovery of a graph signal from its noisy samples at few selected nodes. This problem is relevant, e.g., for semi-supervised learning over graphs, where only few training examples (represented by graph nodes) are labeled and most examples are unlabeled. The problem of determining the labels for the unlabeled data is precisely a graph signal recovery problem. The recovery is feasible for graph signals which are smooth with respect to the graph, i.e., the signal values at neighboring nodes are similar.

Several approaches to the graph signal recovery problem have been put forward. Among them, graph signal recovery method GSDAMP deals with incomplete measurements and recovers the graph signal based on approximate message passing [2]. Also, a wide family of graph recovery algorithms are obtained by convex optimization methods. Based on these methods, [3] provides a recovery solution through the alternating direction methods of multipliers. Within this class of recovery algorithms, methods based on Tikhonov regularization using the graph Laplacian quadratic form are appealing since they amount to solving systems of linear equations involving the graph Laplacian. A recent line of research demonstrates that such Laplacian systems can be solved efficiently using scalable algorithms whose complexity is only linear in the number of edges of the underlying graph [4, 5].

Contribution: We formulate the graph signal recovery problem as an optimization problem using Tikhonov regularization to enforce smoothness of the recovered signal. The optimization problem produces a signal balancing two terms: the empirical error, i.e., the deviation of the recovered signal from the observed noisy samples, and the signal smoothness as measured by the graph Laplacian quadratic form. The optimal signal is characterized by a system of linear equations, which we solve using a block variant of the iterative Gauss-Seidel (GS) method. We relate the convergence properties of this iterative method to the choice of the sampling set and the graph topology. Furthermore, we apply our recovery method to a real-world data-set containing a product rating of a large online retailer. We find our recovery method allows for accurate prediction of product ratings based on few manually reviewed products.

Notation: Boldface lowercase letters denote column vectors, whereas boldface uppercase letters denote matrices. We denote the 2-norm of a vector $\mathbf{x} \in \mathbb{R}^N$ by $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$. The cardinality of a finite set $\mathcal{E}$ is written as $|\mathcal{E}|$. The identity matrix of size $M \times M$ is denoted I_M or just I if the size is clear from the context.

Outline: The rest of the paper is organized as follows: We formalize the graph signal recovery problem in Section 2. The recovery algorithm based on the GS and block GS (BGS) method is presented and discussed in Section 3. The results of illustrative numerical experiments are presented in Section 4.

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

2. THE PROBLEM FORMULATION

Consider an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, with node set $\mathcal{V} = \{1, \ldots, N\}$ and edges set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We assume the graph to be simple, i.e., it does not have any self-loops. Thus, $(r, t) \in \mathcal{E}$ implies $r \neq t$ and $(t, r) \in \mathcal{E}$ (since the edges are undirected). The non-negative entries $W_{r,t}$ of the symmetric weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ quantify the strength of the connections in the graph. In particular, $W_{r,t} \neq 0$ only if $(r, t) \in \mathcal{E}$, i.e., the support of the matrix $\mathbf{W}$ reflects the edge structure of the graph $\mathcal{G}$.

A graph signal $\mathbf{x}[\cdot] : \mathcal{V} \to \mathbb{R}$ defined on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ is a mapping (labeling) of the graph nodes to (with) real numbers, i.e., each node $r \in \mathcal{V}$ is assigned a graph signal value $x[r] \in \mathbb{R}$. We will stack the graph signal values into a vector $\mathbf{x} \in \mathbb{R}^N$ whose rth entry is the graph signal value $x[r]$ at node $r \in \mathcal{V}$. The graph signals arising in many important applications are smooth in the sense that the signal values $x[r], x[t]$ for neighboring nodes $r, t \in \mathcal{V}$ are similar, i.e.,

$$x_r \approx x_t \quad \text{for} \quad (r, t) \in \mathcal{E}. \tag{1}$$

We measure the smoothness of a graph signal $\mathbf{x}$ by [6]

$$S(\mathbf{x}) = \sum_{(r,t) \in \mathcal{E}} W_{r,t}(x_r - x_t)^2 = \mathbf{x}^T \mathbf{L} \mathbf{x}\,. \tag{2}$$

Here, we used the graph Laplacian matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \tag{3}$$

with the diagonal degree matrix $\mathbf{D}$ whose rth diagonal element is given by $D_{r,r} = \sum_{t \in \mathcal{V}} W_{r,t}$.

2.1. The Recovery Problem

We consider the problem of recovering a smooth graph signal $\mathbf{x}_0$ from its noisy samples

$$y_r = x_{0r} + n_r, \quad r \in \mathcal{S} \tag{4}$$

observed at a (typically small) subset $\mathcal{S} \subseteq \mathcal{V}$ of nodes. Let us (without loss of generality) assume henceforth that $\mathcal{S} = \{1, \ldots, M\}$. By stacking the measurements y_r into the vector $\mathbf{y} \in \mathbb{R}^M$, we use a the linear measurement model

$$\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{n}. \tag{5}$$

The noise vector $\mathbf{n} = (n_1, \ldots, n_M)^T$, modeled as additive white Gaussian noise (AWGN) with zero-mean and variance σ^2, i.e., $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2)$, summarizes the effect of modeling and measurement errors. The measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ represents the sampling process and is given by

$$\mathbf{A} = \left(\mathbf{I}_M \; \mathbf{0}_{M \times (N-M)}\right). \tag{6}$$

Each row of $\mathbf{A}$ corresponds to selecting the graph signal value x_{0r} for some $r \in \mathcal{S}$. It should be noted that the special form of (6) is not a strong restriction, as *any* set of M (out of N) graph signal components can be chosen by simple re-labelling.

Our approach for recovering the true graph signal $\mathbf{x}_0$ from the noisy samples y_r (cf. (4)) is based on balancing the empirical error

$$E(\hat{\mathbf{x}}) = \sum_{r \in \mathcal{S}} (y_r - \hat{x}_r)^2 = \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2^2, \tag{7}$$

with the smoothness $S(\hat{\mathbf{x}})$ (cf. (2)) of the recovered signal $\hat{\mathbf{x}}$. Thus, a natural recovery strategy is given by the optimization problem

$$\begin{aligned} \hat{\mathbf{x}} &\in \arg\min_{\mathbf{x} \in \mathbb{R}^N} E(\mathbf{x}) + \lambda S(\mathbf{x}) \\ &= \arg\min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \mathbf{x}^T \mathbf{L} \mathbf{x}. \end{aligned} \tag{8}$$

The parameter $\lambda > 0$ trades off empirical error $E(\hat{\mathbf{x}})$ against smoothness $S(\hat{\mathbf{x}})$ of the recovered signal $\hat{\mathbf{x}}$.

The recovery problem (8) is a convex optimization problem, whose optimal points $\hat{\mathbf{x}}$ are characterized by the linear equation

$$(\mathbf{A}^T \mathbf{A} + \lambda \mathbf{L})\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{y}. \tag{9}$$

As the notation in (8) indicates, there might be different optimal points $\hat{\mathbf{x}}$. For the graph $\mathcal{G}$ with K connected components $\mathcal{C}_l$, a necessary and sufficient condition for the existence of a single unique optimal point $\hat{\mathbf{x}}$ of the optimization problem (8) is when the sampling set $\mathcal{S}$ contains at least one node r_l, $l = 1, \ldots, K$, from each connected component $\mathcal{C}_l$ of the graph $\mathcal{G}$.

To obtain a more compact form of (9), we rewrite it according to

$$\mathbf{C}\mathbf{x} = \mathbf{b} \tag{10}$$

with the system matrix $\mathbf{C} := \mathbf{A}^T \mathbf{A} + \lambda \mathbf{L}$ and the vector $\mathbf{b} = \mathbf{A}^T \mathbf{y}$. The off-diagonal entries of $\mathbf{C}$ are given by $C_{r,j} = -\lambda W_{r,j}$ (cf. (3)), whereas the diagonal entries are given by

$$C_{r,r} = \begin{cases} 1 + \lambda D_{r,r} & \text{for } r \in \mathcal{S} \\ \lambda D_{r,r} & \text{for } r \notin \mathcal{S}. \end{cases} \tag{11}$$

The vector $\mathbf{b}$ in (10) with support $\operatorname{supp}(\mathbf{x}) = \mathcal{S}$, contains the noisy graph signal samples y_r (cf. (4)), i.e., $\mathbf{b} = (y_1, \ldots, y_M, \mathbf{0})^T$. In what follows, we will consider graph signals defined over connected graphs. This does not incur any loss of generality. Indeed, if the graph is composed of several not interconnected sub-graphs, the recovery problem (8) and the associated linear system of equations (10) would split into independent sub-problems, one for each sub-graph.

3. EFFICIENT ITERATIVE GRAPH SIGNAL RECOVERY

In order to obtain the recovered signal $\hat{\mathbf{x}}$, we have to solve a system of linear equations. A basic iterative method to solve large systems of linear equations such as (10) is the Gauss-Seidel (GS) method [7]. In particular, the GS method constructs a sequence $\mathbf{x}^{(t)}$ by iterating, for $t = 1, 2, \ldots$, the following node-wise updates, for $r = 1, 2, \dots, N$.

$$x_r^{(t)} = (1/C_{r,r})\big(b_r - \sum_{j=1}^{r-1} C_{r,j} x_j^{(t)} - \sum_{j=r+1}^{N} C_{r,j} x_j^{(t-1)}\big)$$
$$= \begin{cases} \frac{1}{1+\lambda D_{r,r}}\big(y_r - \sum\limits_{j=1}^{r-1} W_{r,j} x_j^{(t)} - \sum\limits_{j=r+1}^{N} W_{r,j} x_j^{(t-1)}\big) & \text{if } r \in \mathcal{S} \\ \frac{1}{\lambda D_{r,r}}\big(\sum\limits_{j=1}^{r-1} W_{r,j} x_j^{(t)} - \sum\limits_{j=r+1}^{N} W_{r,j} x_j^{(t-1)}\big) & \text{if } r \notin \mathcal{S} \end{cases} \tag{12}$$

We stress that the order of the updates (12) for the components $x_r^{(t)}$ is important: first, we update $x_1^{(t)}$, then $x_2^{(t)}$ and so on, i.e., those graph nodes are updated first, for which measurements are available (see (6)). As to the convergence of the resulting sequence $\mathbf{x}^{(t)}$, we note the following result from [8]: Given that we have a regular (invertible) system matrix $\mathbf{C}$ in (10), it is known from [8] that the GS algorithm will converge to the unique solution $\hat{\mathbf{x}}$, if for the recursively defined numbers

$$p_r = \sum_{j=1}^{r-1} \frac{|c_{r,j}|}{|c_{r,r}|}\, p_j \; + \sum_{j=r+1}^{N} \frac{|c_{r,j}|}{|c_{r,r}|} \quad \text{for} \quad r = 1, 2, ..., N \;, \tag{13}$$

the condition

$$\Big(\max_{r=1,2,\dots,N} p_r\Big) \;<\; 1 \tag{14}$$

is fulfilled ($p_r \geq 0$ because all terms in (13) are non-negative); also

$$c_{r,r} \neq 0 \;, \quad \text{for} \quad r = 1, 2, ..., N, \tag{15}$$

holds because of (11) and the weights being non-negative.

The Block Gauss-Seidel (BGS) method [9, Chap 10] generalizes the GS method by updating during each iteration whole blocks of the current estimate $\mathbf{x}^{(t)}$ in one step instead of single entries (as in (12)), which allows for parallel computations and, hence, significant increase in speed, even though the complexity of one update step may appear to be higher for BGS in general. However, a major advantage of BGS is that it typically requires fewer iterations to reach a given solution accuracy. The BGS method is based on partitioning the system matrix $\mathbf{C}$, solution vector $\mathbf{x}$ and vector $\mathbf{b}$ of (10), into p blocks according to

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \dots & \mathbf{C}_{1p} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \dots & \mathbf{C}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{p1} & \mathbf{C}_{p2} & \dots & \mathbf{C}_{pp} \end{pmatrix}, \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_p \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} \tag{16}$$

The iterations for the BGS method for solving (10) are defined by generalizing (12); the scheme is stated in Algorithm 1.

Algorithm 1 Smooth graph signal recovery via BGS

```
Input: $\mathbf{x}^0 \leftarrow 0$, $\mathbf{C} = \mathbf{A}^T\mathbf{A} + \lambda\mathbf{L}$ (cf. (6), (3)), $t = 0$,
  noisy samples $\{y_r\}_{r\in\mathcal{S}}$, sampling set $\mathcal{S}$ and parameter $\lambda$
repeat
    for r=1:p do
        $\tilde{\mathbf{b}} \leftarrow (\mathbf{b})_r - \sum_{j=1}^{r-1} \mathbf{C}_{rj}\mathbf{x}_j^{(t)} - \sum_{j=r+1}^{N} \mathbf{C}_{rj}\mathbf{x}_j^{(t-1)}$
        Solve $\mathbf{C}_{rr}\hat{\mathbf{x}}_r = \mathbf{b}_r$ using (12) to obtain $x_r^{(t)}$
    end for
    $t \leftarrow t + 1$
until the stopping criterion is met.
Output: recovered graph signal $\hat{\mathbf{x}} = \mathbf{x}_r^{(t-1)}$
```

With respect to convergence of the algorithm we consider the coefficient matrix $\mathbf{C}$ in equation (10): its diagonal entries are obtained from (11) and its off-diagonal entries are given by $C_{r,j} = -\lambda W_{r,j}$. With the definition of irreducibly diagonally dominant matrices from [7, Def 4.5], the rth row of the matrix $\mathbf{C}$ is either strictly dominant for $r \in \mathcal{S}$, as $\sum_{j\in V, r\neq j} \mid \lambda W_{rj} \mid < 1 + \lambda D_{rr}$, or weakly dominant for $r \notin \mathcal{S}$, as $\sum_{j\in V, r\neq j} \mid \lambda W_{rj} \mid \leq \lambda D_{rr}$. This makes the matrix $\mathbf{C}$ irreducibly diagonally dominant. From [7, Theorem 4.9], if the matrix $\mathbf{C}$ is an irreducibly diagonally dominant matrix, then $\hat{\mathbf{x}}$ generated by Algorithm 1 converges to the unique solution.

4. NUMERICAL RESULTS

In order to asses the BGS, we compared the performance of the recovery method given by Algorithm 1 with the incomplete Cholesky factorization conjugate gradient (ICCG) method [10, Chap 10] in the context. The convergence rate of conjugate gradient (CG) method depends on the condition number of the matrix $\mathbf{C}$ but unfortunately the matrix $\mathbf{C}$ is often ill-conditioned, i.e., it has large condition number.

In order to asses the accuracy of the Algorithm 1, we applied it to a real-world product rating data-set, which was collected by crawling the website of a large Internet-based retailer [11]. This data-set consists of rating information of four different product categories: books, music CDs, DVDs and video tapes. The products are represented by the nodes of the graph. The nodes representing two particular products are connected by an edge if they are co-purchased often. Each product is assigned a rating taking on values in the set 1/2$\{0,$ $1, \dots, 9, 10\}$. We then obtain a graph signal $\mathbf{x}_0$ by setting its value at node r to the average of all ratings for the product r. The graph of the raw data contained isolated nodes and several small components. We selected the largest connected subgraph $\mathcal{G}$ for our numerical experiments. In this graph there are $N = 290744$ nodes and $|\mathcal{E}| = 729048$ edges. We omitted edge directions to obtain an undirected graph. We randomly selected M signal samples x_r and added zero-mean AWGN noise with variance σ^2. Thus we obtain a measurement vector $\mathbf{y}$ conforming to the model (5).

In order to recover $\mathbf{x}_0$ from the noisy measurements (4), we applied Algorithm 1 using a partitioning of the graph into $p = 8$ blocks of equal size. The stopping criteria used for Algorithm 1 was either it reaches the maximum number of 20

Table 1. NMSE of the BGS and ICCG algorithms.

	BGS		ICCG	
	$\sigma^2 = 0$	$\sigma^2 = 4$	$\sigma^2 = 0$	$\sigma^2 = 4$
SR=0.1	0.1472	0.1743	0.1469	0.1744
SR=0.2	0.1307	0.1671	0.1308	0.1674
SR=0.3	0.1151	0.1590	0.1148	0.1590
SR=0.4	0.0996	0.1500	0.0997	0.1502
SR=0.5	0.0846	0.1414	0.0846	0.1414
SR=0.6	0.0699	0.1329	0.0698	0.1328
SR=0.7	0.0552	0.1244	0.0552	0.1242
SR=0.8	0.0407	0.1158	0.0406	0.1157
SR=0.9	0.0263	0.1072	0.0263	0.1073

iterations or the relative progress $\frac{\|\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}\|_2}{\|\mathbf{x}^{(t)}\|_2} \leq 10^{-3}$. We ran Algorithm 1 for different values of λ, via the simulations, and set $\lambda = 0.1$ which yields good performance in terms of the normalized mean square error (NMSE), NMSE = $\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}$.

We analyzed the effect of different noise levels σ^2 and varying sampling rates (SR) M/N on the NMSE of the Algorithm 1. The results obtained for Algorithm 1 together with those obtained for an alternative method, based on incomplete Cholesky factorization preconditioned conjugate gradient are shown in Table 1. As evident from Table 1 both BGS and ICCG show good recovery performance even for very low SR M/N and high noise variance σ^2. What is also evident from Table 1, ICCG and BGS do not show much difference in NMSE. Table 1 indicates that the noise variance σ^2 has negligible effect on NMSE, specially for lower SR's.

Another figure of merit, beside the NMSE, is the label recovery ratio (LRR) a_η (17) defined as the fraction of nodes $r \in \mathcal{V}$, for which the recovery error $|x_r - \hat{x}_r|$ does not exceed the threshold η, i.e.,

$$a_\eta = | \{ r \in \mathcal{V}, | x_r - \hat{x}_r | \leq \eta \} | / N. \qquad (17)$$

Here $\eta \in 1/2 \{0, 1, 2, \ldots, 10\}$, since the rating of the products come from the set $1/2 \{0, 1, 2, \ldots, 10\}$ we round the value of recovered signal to the nearest signal value in the set. In (17), x_r is the original signal value at node r, $\hat{x}_r$ is the recovered signal value.

The LRR of BGS obtained for Algorithm 1 for varying M/N and noise variance σ^2 is shown in Fig. 1 and Fig. 2, respectively. In both figures, the solid lines show the LRR a_0, the dotted lines show the LRR $a_{1/2}$ and the dashed lines the LRR a_1. From Fig. 1 we have that, for SR $M/N = 0.2$ and $\sigma^2 = 0$, the algorithm is able to recover the correct ratings of more than 29% of the products ($a_0 \geq 0.29$). If we accept the maximum error threshold $\eta = 1/2$, the algorithm recovers 60% of the graph values ($a_{1/2} \geq 0.6$) and for the threshold $\eta = 1$, it recovers more than 76% of the graph values ($a_1 \geq 0.76$). For noise variance $\sigma^2 = 4$, the recovery performance deteriorates to $a_0 = 0.19$, $a_{1/2} = 0.47$ and $a_1 = 0.66$. As expected, the LRR of the BGS algorithm improves for increasing SR. Beside, the LRR curve for noise variance $\sigma^2 = 0$ is much steeper compared to $\sigma^2 = 4$. The effect of increasing SR is less pronounced for higher noise variance.

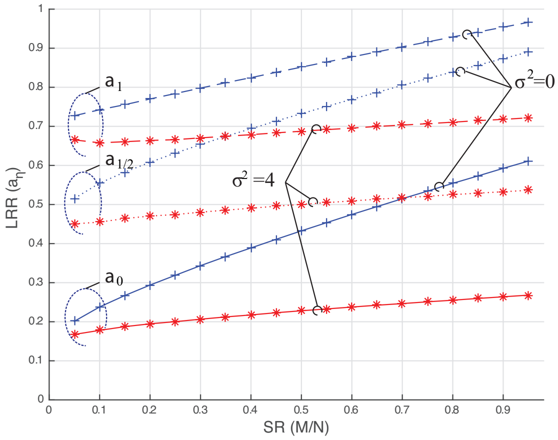

Fig. 1. LRRs (a_0, $a_{1/2}$, a_1) of Algorithm 1 for varying M/N and noise variance $\sigma^2 \in \{0, 4\}$.

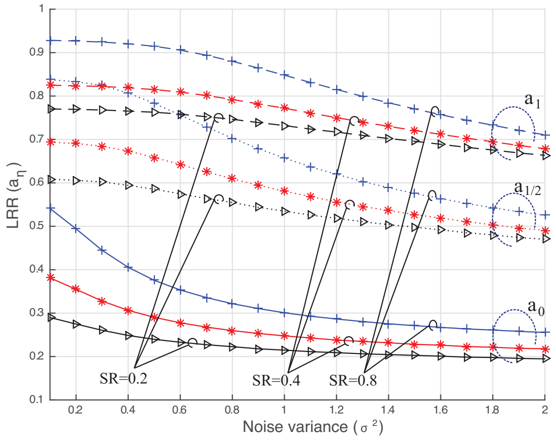

Fig. 2. LRRs of Algorithm 1 (a_0, $a_{1/2}$, a_1) for noise variance σ^2 and SR $M/N \in \{0.2, 0.4, 0.8\}$.

In Fig. 2, we illustrate the recovery performance over noise variance σ^2 for three different SR $M/N \in \{0.2, 0.4, 0.8\}$. Again, incrementing the noise level harms the recovery performance. We observe that ICCG and BGS deliver similar recovery performance, so we only show the results of the BGS in the figures.

5. CONCLUSION

We formulate the problem of recovering a smooth graph signal from a small number of noisy samples as a convex optimization problem which, in turn, amounts to solving a system of linear equations involving the graph Laplacian. An efficient recovery method for smooth graph signals is then obtained by applying a BGS method to this Laplacian system. The effectiveness of the proposed recovery method is verified by numerical experiments via a real-world data-set containing product ratings of a large Internet-based retail shop.

REFERENCES

[1] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, apr 2013.

[2] G. B. Eslamlou, A. Jung, N. Goertz, and M. Fereydooni, "Graph signal recovery from incomplete and noisy information using approximate message passing," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6170–6174.

[3] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4609–4624, 2015.

[4] N. K. Vishnoi, "$Lx = b$ Laplacian solvers and their algorithmic applications," *Foundations and Trends in Theoretical Computer Science*, vol. 8, no. 1–2, pp. 1–141, 2012.

[5] O. E. Livne and A. Brandt, "Lean Algebraic Multigrid (LAMG): Fast Graph Laplacian Linear Solver," *SIAM Journal on Scientific Computing*, vol. 34, no. 4, pp. B499–B522, jan 2012.

[6] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[7] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics, 2003.

[8] A. Meister, *Numerik linearer Gleichungssysteme*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015.

[9] W. J. Stewart, *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton University Press, 2009.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.

[11] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Transactions on the Web*, vol. 1, no. 1, p. 5, may 2007.