Signal Processing for Big Data

Sampling and Recovery of Graph Signals

Lazareva Olga, Pisani Andrea

Exam Project

Abstract

The main purpose of the project is to implement a sampling strategy that collects a subset of samples of a signal defined over a graph and then reconstruct the original signal using only the subset. In our study case, an electromagnetic field is sampled over an area of a city and its power is represented over a graph matching the topology of the area. After verifying that the conditions to recover the signal holds, we used different functions in a greedy algorithm in order to choose the optimal sample set and reconstruct the signal. The final choice of the sample set is made analyzing the error between the real signal and the reconstructed signal. The whole code is written in Python 3.

1 Methods

1.1 Initial data

The initial data for the project was given as:

- the adjacency matrix of the graph
- the topological position of each node (on x and y axis)
- the signal mapped over the graph means the values representing the power of the electromagnetic field in each node

The whole code is written in Python and the visualization of our data over the graph is obtained using the **Networkx** and **matplotlib.pyplot** libraries. All the function for linear algebra calculation were used from **numpy.linalg** library.

In Figure 1 we showed the signal mapped over the graph with 486 nodes. The initial adjacency matrix had weighted edges, so the first step was changing all values greater than 0 to 1. Then we built the diagonal degree matrix D and computed the Laplacian matrix $L = D - A$. The first eigenvalue of the Laplacian was zero with multiplicity one and the first eigenvector has constant values thus the graph is fully connected.

1.2 Reconstruction

In order to chose a sample set that can reconstruct the original signal by interpolation we had to choose a sample set S, and the bandwidth $|F|$. In order to choose F we picked the highest n

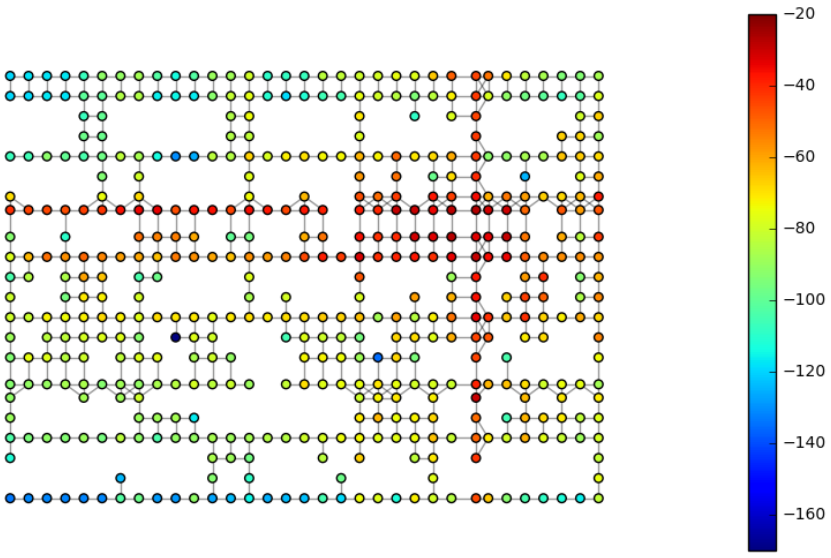

Figure 1: Signal mapped over the graph

frequencies that we obtained by applying Graph Fourier Transform. Vertex-limiting operator was defined as $D_S = diag\{1_S\}$ and P_S is the sampling matrix with dimensions $486 \times |S|$, such that:

$$D_S = P_S P_S^T$$

The observation model y_S contains all the samples chosen from the original vector x:

$$y_S = P_S^T x$$

In order to reconstruct the initial signal x using the sampled version y_S we have used the following reconstruction stratagy:

$$\hat{x} = U_F (U_F^H D_S U_F)^{-1} U_F^H P_S y_S$$

1.3 Sampling Algorithms

The selection of the sample was made using a greedy algorithm, a method that iteratively adds to the graphs those nodes that lead to the largest (or smallest) increment of a function chosen as a performance metric. For the *A-optimal Design* the function $f(S)$ is:

$$f(S) = Tr\{(U_f^H D_S R_v^{-1} U_f)^- 1\}$$

Where R_v is the covariance matrix $R_v = diag\{{r_1}^2, ..., {r_N}^2\}$ represents zero-mean uncorrelated gaussian noise. For the *E-optimal Design* the function $f(S)$ is:

$$f(S) = \sigma_{min}(D_S U_f)$$

At this point, we have to note that this approach may lead to numerical problems and hence we slightly modified the formula in order to get more stable results:

$$f(S) = \sigma_{max}(D_S^c U_f)$$

And in this way we have to consider argmin of the function.

For the *D-optimal Design* the function $f(S)$ is:

$$f(S) = \log_{10} p \det(U_f^H D_S R_v^{-1} U_f)$$

And then we compute the argmax.
In order to have the same scale to evaluate our results, we computed the normalized mean squared error NMSE for each optimal design:

$$NMSE = \frac{\|\hat{x} - x\|^2}{\|x\|^2}$$

1.4 Convex relaxation

Also we adopted another possible way to get a solution for the sampling problem using convex relaxation techniques. Thus, we used the indicator vector $d = \{d_i\}_{i=1}^N$, such that the i-th entry is binary and given by $d_i = 1$ if node i belongs to the sampling set S, and $d_i = 0$ otherwise. But this would lead us to a combinatorial problem rather than a convex one and hence we relaxed the indicator variable d to be a real vector belonging to the hypercube $[0, 1]^N$, thus the problem can be formulated as:

$$\begin{aligned} &\min_{d\in[0,1]} f(d) \\ &s.t. 1^T d = M \end{aligned} \tag{1}$$

We can easily apply A/E/D optimal design if we set $D_s = diag(d)$ and then select M largest elements of d^* - the optimal solution.

2 Results

2.1 Fixed bandwidth with increasing sample size

In the Figure 2 is shown the plot of the NMSE vs number of samples with a fixed bandwidth $|F| = 20$. As we can see that with the A-optimal or D-optimal design we are always able to get lower error rather than with E-optimal.

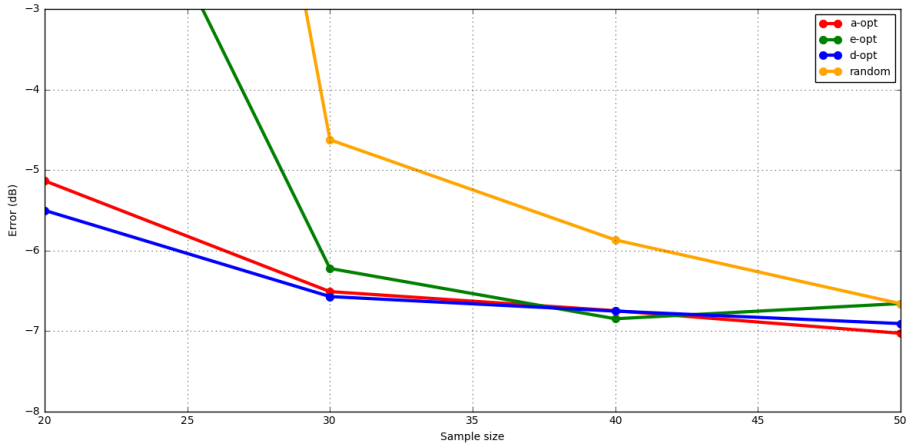

Figure 2: NMSE in dB versus sample size

We chose to reconstruct the signal using 40 sample from the D-optimal design. As we can see in the Figure 3, the samples are spread over the graph more or less uniformly with respect to the second eigenvector plotted over the graph.

As we can see at the Figure 4, the reconstructed signal is quite close to the original one.

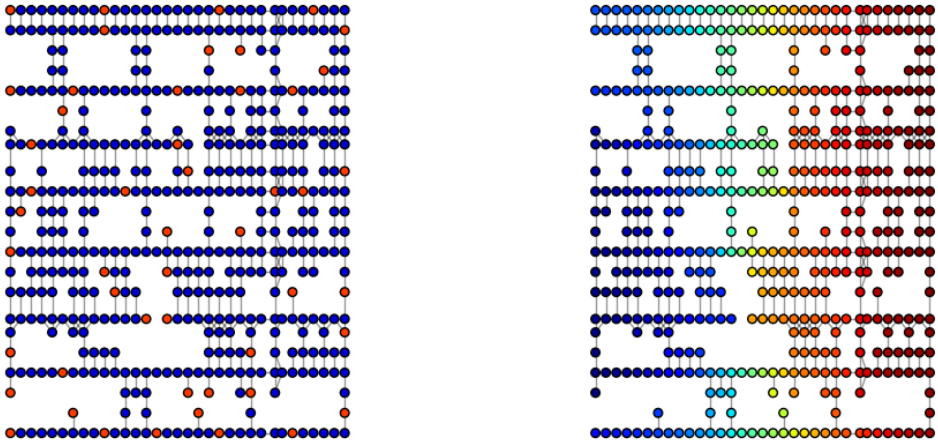

Figure 3: Left: nodes chosen with the A-opt stratagy, right: second eigenvector

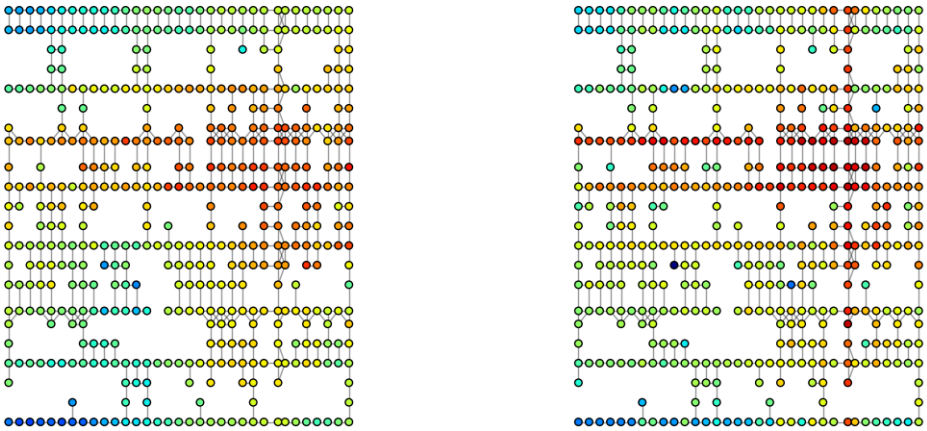

Figure 4: Left: reconstructed signal. Right: original signal

2.2 Convex relaxation results

After archiving the first results we got curious if we can improve them if we use convex problem formulation. Thus the initial setting was the same: $|F| = 20$ and the sample size is increasing from 20 to 50 samples. As it shown on the figure 6 the best result was archived with combination of D-optimal

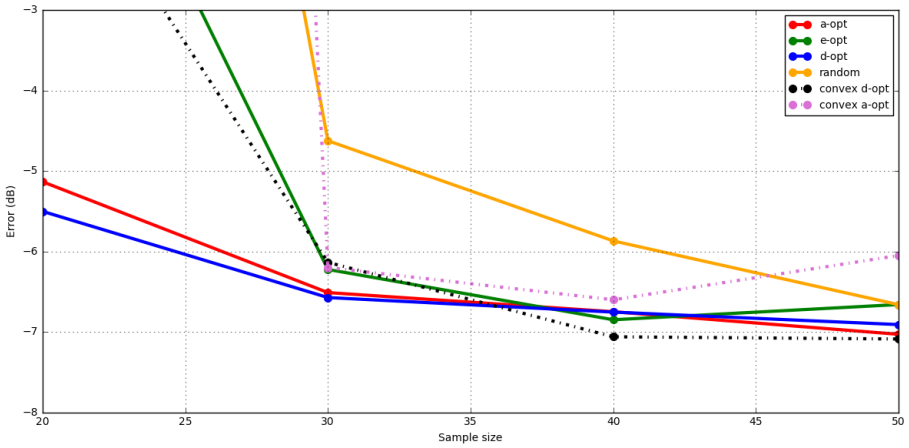

Figure 5: NMSE for greedy approach and convex relaxation solution

design and convex relaxation. Unfortunately, E-optimal desigh generally didn't show any performance with this aproach.

2.3 Bandwidth equal to the sample size

We also wanted to study an effect of increasing bandwidth and thus we performed greedy search for D-optimal and A-optimal design and $|s| = |f| + 10$ (in order to guarantee that matrix will always be full-rank) As it was expected the best performance was archived with bandwidth $|f| = 60$ and $|s| = 70$

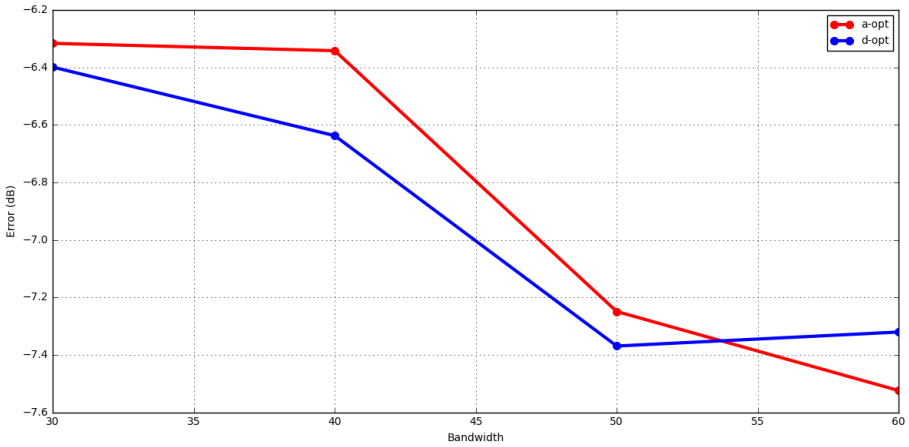

Figure 6: Bandwidth vs NMSE

3 Conclusion

During the project, we have illustrated sampling strategies, based on greedy methods or convex relaxations and although the results seem to be accurate, we assume that we could improve it with some more study of an effect of noise and some technics to estimate it given the original signal. We assume that sometimes we were getting increasing nmse with increasing bandwidth (like on Figure 6) because we were picking noisy samples and they gave us a decrease in overall performance.

The report was written based on the chapter ”Sampling and Recovery of Graph Signals” by Paolo Di Lorenzo, Sergio Barbarossa, and Paolo Banelli

The full Python notebook can be found at the following link:
https://www.dropbox.com/s/q7cf4qzsm7bngrj/the