Signal Inpainting using Graphs

Hemanth Sabbella
Sheel Priya Gautam
Varun Kumar

Mentor:
Naushad Ansari

Aim

- Our aim is to implement an algorithm that can recreate missing parts of a signal using graph variation minimization and regulation[1].

Converting 1-D signal to graph

- The amplitude values of the discrete time signals form the nodes of the graph.
- The graph is formed by connecting all the nodes to their nearest nodes in time.
- Therefore the adjacency matrix formed is of the size NxN where N is the length of the signal.
- However, the way of forming graph is flexible for 1-D signals.
- If the data is of nature that requires different style of connections then adjacency matrix can be modified suitably and then a signal can be made out of that graph.

Converting Image to Graph

- To convert an image into an 8-way adjacency matrix.
- Give value of weights to be given is 8 for neighbouring nodes and 1 for node connected to itself.
- Finding Distorted pixels(Unknown Pixels).
- Putting them to one corner of the adjacency matrix.
- Apply Inpainting Algorithm Developed by other team on that adjacency matrix.

Concept Behind Making Adjacency Matrix

For ease of visualization lets take a 3x4 image.

1) Convert this 2D image into a 1D signal by joining each row one after another, ie, 3x4 image will become 1x12
2) Make an eight way adjacency matrix of size 12x12 for this 1D signal
3) Value of weights for the neighbouring node is 8.
4) Put 1 in the diagonal of this adjacency to connect node from itself.
5) Now segregate:
    a) Known-Known
    b) Known-Unknown
    c) Unknown-Known
    d) Unknown-Unknown

Let’s Denote Unknown Pixels with 7 at (1,3) and (2,1)

1	8	7	0	8	8	0	0	0	0	0	0
7	1	8	0	8	8	8	0	0	0	0	0
0	8	1	8	0	8	8	8	0	0	0	0
0	0	8	1	0	0	8	8	0	0	0	0
8	8	0	0	1	8	0	0	8	8	0	0
8	8	8	0	8	1	8	0	8	8	8	0
0	8	8	8	0	8	1	8	0	8	8	8
0	0	8	8	0	0	8	1	0	0	8	8
0	0	0	0	8	8	0	0	1	8	0	0
0	0	0	0	8	8	8	0	8	1	8	0
0	0	0	0	0	8	8	8	0	8	1	8
0	0	0	0	0	0	8	8	0	0	8	1

Segregating Unknown Pixels

8	8	0	8	8	8	0	0	0	0	1	0
0	1	0	0	8	8	0	0	0	0	8	0
8	0	1	8	0	0	8	8	0	0	0	8
8	0	8	1	8	0	8	8	8	0	8	8
8	8	0	8	1	8	0	8	8	8	8	0
0	8	0	0	8	1	0	0	8	8	8	0
0	0	8	8	0	0	1	8	0	0	0	0
0	0	8	8	8	0	8	1	8	0	0	0
0	0	0	8	8	8	0	8	1	8	0	0
0	0	0	0	8	8	0	0	8	1	0	0
8	0	8	8	0	0	0	0	0	0	7	1
1	0	8	8	8	0	0	0	0	0	8	7

Known-Known

Known-Unknown

Unknown-Known

Unknown-Unknown

Signal Inpainting on graphs

- In order to implement signal inpainting, signal must be of this form:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\mathcal{M}} \\ \mathbf{x}_{\mathcal{U}} \end{bmatrix}$$

- Now the adjacency matrix is made for the above signal.
- Two methods
  - GTVR (Graph total variation regularization)

$$\mathbf{x}^* = \left(\begin{bmatrix} \mathbf{I}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \lambda \widetilde{\mathbf{A}} \right)^{-1} \begin{bmatrix} \mathbf{x}_{\mathcal{M}} \\ \mathbf{0} \end{bmatrix}$$

  - GTVM (Graph total variation minimization)

$$\widehat{\mathbf{x}}_{\mathcal{U}} = -\widetilde{\mathbf{A}}_{\mathcal{U}\mathcal{U}}^{-1} \widetilde{\mathbf{A}}_{\mathcal{U}\mathcal{M}} \mathbf{x}_{\mathcal{M}}.$$

Implementation

- We implemented the algorithm online blogs[1].
- We took a data set of 40 blogs and classified blogs into two types - conservative and liberal.
- We represented conservative labels as +1 and liberal as -1.
- Adjacency matrix:
  - For any node v its outgoing node has weights 1/deg(v) where deg(v) is the number of outgoing edges.
- Now we implement both the methods on the graph and let's see what's the result.

References

[1] SIGNAL INPAINTING ON GRAPHS VIA TOTAL VARIATION MINIMIZATION

Siheng Chen, Aliaksei Sandryhaila, George Lederman, Zihao Wang, José M. F. Moura, Piervincenzo Rizzo, Jacobo Bielak, James H. Garrett and Jelena Kovačević.

[2] SIGNAL DENOISING ON GRAPHS VIA GRAPH FILTERING

Siheng Chen, Aliaksei Sandryhaila José and M. F. Moura Jelena Kovačević.