Graph Signal Processing Exercise Set 2

Polynomials - Products - Applications

April 6, 2025

Learning Goals

By the end of this exercise, you should be able to...

- Understand how graph filters are polynomials
- Know how to combine graphs into one graph, depending on the application
- Use graph filters for value prediction

Filters are polynomials in shift

Proof that all linear, shift-invariant graph filters are polynomials in the shift, that is, proof that for any $H \in \mathbb{C}^{N \times N}$ with

$$H \cdot (T) = T(Hs)$$

it holds

$$H = p(T) = \sum_{k=0}^{N} a_k T^K$$

for some $a_k \in \mathbb{C}$.

Products of graphs

If one wants to build a graph out of smaller graphs, one has several options. We will discuss three here (though there are more). Consider two graphs G (with N nodes) and H (with M nodes) with corresponding adjacency matrices A_G and A_H. The identity matrix is

$$1_N = \begin{bmatrix} 1 & 0 & \dots & \\ 0 & 1 & 0 & \dots \\ & & \dots & \\ \dots & & 0 & 1 \end{bmatrix}$$

and the all-one matrix is

$$J_N = \begin{bmatrix} 1 & 1 & \dots & \\ 1 & 1 & 1 & \dots \\ & & \dots & \\ \dots & & 1 & 1 \end{bmatrix}$$

. The symbol $\otimes$ denotes the Kronecker product. Consider the following three graph products, defined by their adjacency matrices:

- Cartesian product, adjacency matrix $A_G \otimes 1_M + 1_N \otimes A_H$,
- Tensor product, adjacency matrix $A_G \otimes A_H$,

- Lexicographic product, adjacency matrix $A_G \otimes J_M + 1_N \otimes A_H$.

Take as graph G an undirected cycle graph one 3 nodes and as graph H a directed path on 3 nodes and draw all three graph products.

Predicting traffic

Consider the PEMS-BAY or METR-LA traffic data (graph `https://github.com/liyaguang/DCRNN` and signals `https://drive.google.com/drive/folders/10FOTa6HXPqX8Pf5WRoRwcFnW9BrNZEIX`). These data sets contain traffic speed measurements from loop detectors of a highway system.

Choose a small subgraph (e.g.500 nodes) and combine it with the time information to a graph ((e.g. considering the Cartesian or lexicographic product of graphs, or maybe something smarter?).

Compare the results of graph-based smoothing with traditional time-series smoothing (e.g. rolling average over time for each sensor). Implement a simple prediction model where traffic at time $t+1$ is predicted using a graph filter applied to traffic at time t (that is, similarly to label propagation as in the lecture, but instead of randomly sampling s^{known} you have s^{known} from the nodes with timestep t). Evaluate the prediction accuracy and discuss how incorporating the graph structure compares to individual time series forecasting.