A framework for learning how information changes under diffusion in networks

July 22, 2024

I suppose this is more of a description of Nonlocal Operators living on graphs and Vertex-Frequency Analysis in Graph Signal Processing but from a Functional Analytic perspective...?

0.1 General Model Description

0.1.1 Introduction and Motivation

This topic should explore the interplay of discrete mathematics and functional analysis, with an introduction of modern computer learning algorithms. The idea is to show that the network structure can be used for actual computational mathematical analysis. The idea is to shift from analysis of resolvent operators for evolutionary equations on network structured data, and to learn solution operators to these equations. This dissertation should provide a framework for learning such mappings systematically, not only for descriptive purposes but for constructing and computing direct solutions to differential equations living on networks. Here are some preliminary ideas [subject to change].

0.1.2 Mathematical Structure

Suppose we have a time varying graph $G(t) = (V(t), E(t))$ where $V(t)$ represents agents on the graph, and $E(t)$ represents connections between the agents. Here time is continuous, $t \in \mathbb{R}^+$. The manipulation of information between agents v_i and v_j as it diffuses through a network is described by an operator Ψ that operates on information θ_i . A transfer function ψ_i describes the change of information from θ_i to θ_j contained in the agents $v_i \in V$ and $v_j \in V$, respectively for all $(i, j) \in \mathbb{N}$. Each transfer function ψ_i is an element an of a vertex function ψ on the graph G. Both names shall be adopted. The vertex function shall correspond to a vector-valued function $\psi(\boldsymbol{\theta})$ when individual transfer functions are not considered.

Information should encompass any data or signals being transmitted and processed by an agent. Here, the information θ_i and θ_j is contained in a set Θ, the set of all possible information on G. There is a space $\mathcal{H}(V)$ which contains all vertex functions ψ. $\mathcal{H}$ is a separable Hilbert Space, and $\mathcal{H}(V) \cong \mathbb{R}^{|V|}$, the space of real-valued vector-functions of dimension equal to the number of agents, more precisely $\mathcal{H} = \mathscr{L}^p\left(\mathbb{R}^{|V|}\right)$. There is an operator Ψ that operates on G to modify the information θ_i from agent v_i to all other agents v_j. Ψ acts as a unifying framework to model information transformations given collectively by ψ in complex networks.

When we place a dynamical process on a G, the graph can be referred to as a “substrate”.

Definition 1. If a differential equation

$$\psi(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\theta}}{\partial t}$$

has a solution for all $\boldsymbol{\theta}^* \in \Theta$ and $t \in \mathbb{R}^+$, then there is a one parameter semigroup of operators $\Psi(\cdot;t)$ parametrised by time $t \geq 0$ such that

$$\begin{aligned}
\boldsymbol{\theta}(t) &= \Psi\left(\boldsymbol{\theta}^*;t\right), \quad t \in (0,\infty) \\
\Psi\left(\boldsymbol{\theta}^*, t+s\right) &= \Psi\left(\Psi\left(\boldsymbol{\theta}^*;s\right);t\right), \quad t,s \in \mathbb{R}^+ \\
\Psi\left(\boldsymbol{\theta}^*;0\right) &= \boldsymbol{\theta}^*
\end{aligned} \tag{1}$$

Then we call Ψ a solution operator for DE $\psi(\boldsymbol{\theta}) = \nabla_t \boldsymbol{\theta}$.

In this scenario we consider the iterated composition mapping

$$\begin{aligned}
&\bigcirc_{k=1}^{\infty}\Psi\left(\cdot;t_k\right), \quad k>0, \quad t \in \mathbb{R}^+ \\
\Psi\left(\cdot, \sum_{k=1}^{\infty} t_k\right) &= \bigcirc_{k=1}^{\infty}\Psi\left(\cdot;t_k\right), \quad \text{by (1)}
\end{aligned}$$

which links discrete time mappings on graphs with continuous time differential equations. [a better notation may be $\Psi^{(J)}(\cdot;t_k)$, $J \to \infty$ and $\{t_k\}_{k=1}^{\infty}$].

0.1.3 Overarching Goal

The aim is to build a framework to find a solution operator Ψ for the nonlinear-vector-differential equation

$$\psi(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\theta}}{\partial t}$$

using machine learning. Where each transfer function ψ_i itself is a dynamical system living on a network G. The solution operator Ψ maps the initial state $\boldsymbol{\theta^*} = \boldsymbol{\theta}(0) \to \boldsymbol{\theta}(t)$ at any later time t such that

$$\boldsymbol{\theta}(t) = \Psi(\boldsymbol{\theta}(0);t).$$

The operator Ψ generates a trajectory of $\boldsymbol{\theta}$ over time $t \in \mathbb{R}^+$ based on dynamics given by the vertex function ψ. The function ψ need not be your general mathematical function, take an image, images can be considered as functions of light defined on a continuous region - so the information Θ is light spectra. How does an image change as it flows through a network?

Example 1. (simple non-time varying graph) Let an undirected graph G be time-independent, the structure does not change over time, and consists of four agents with initial information states $\{1,2,3,4\} \subset \Theta$. All information reaches every node and is transformed uniformly, and no information is lost. Ψ generates a conservative system where information is preserved as it moves through the network. Finally, $\theta_0 = 0$. There exists a transfer function $\psi_i : \theta_i \to \theta_j$ that maps the information state θ_i at agent v_i to the information state θ_j at agent v_j. Thus, Ψ is a operator that acts in the set of all vertex functions ψ in $\mathcal{H}(V)$. Given that $\mathcal{H}(V)$ is isomorphic to the $|V|-$dimensional real vector space $\mathbb{R}^{|V|}$, we can further specify Ψ as an operator that maps each vertex function ψ in $\mathcal{H}$ to another vertex function in $\mathcal{H}$,

$$\Psi : \mathcal{H} \to \mathcal{H}.$$

By taking θ_i and applying Ψ to get $\Psi(\theta_i(\cdot);t)$ we can build a picture of how information θ_i changes as it is diffused by ψ_i on each agent v_i over the graph G. A very simple transfer function may be

$$\begin{aligned}
\psi_i(\theta_i) &= \theta_i + \theta_{i-1} \\
&= \frac{d\theta_i}{dt}
\end{aligned}$$

where $\Theta \subseteq \mathbb{N} \cup \{0\}$ and we <u>know</u> how the information changes locally, i.e. that the information θ_{i-1} contained at previous node v_{i-1} has now become part of the information contained at v_i in the form $\theta_i + \theta_{i-1}$, like they are boxes that collect information and send it across to the next node - eventually to the last node at a terminal time t^* (the graph structure is time-independent not the dynamic system on the graph). We aim to find what $\Psi(\boldsymbol{\theta})$ is. A starting point is

$$\begin{aligned}
\psi(\boldsymbol{\theta}) &= \frac{d\boldsymbol{\theta}}{dt} \\
&= \begin{pmatrix} \theta_1 + \theta_0 \\ \theta_2 + \theta_1 \\ \theta_3 + \theta_2 \\ \theta_4 + \theta_3 \end{pmatrix} \\
&= \begin{pmatrix} \theta_1 \\ \theta_2 + \theta_1 \\ \theta_3 + \theta_2 \\ \theta_4 + \theta_3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{pmatrix} \\
&= \mathbf{P}\boldsymbol{\theta}
\end{aligned}$$

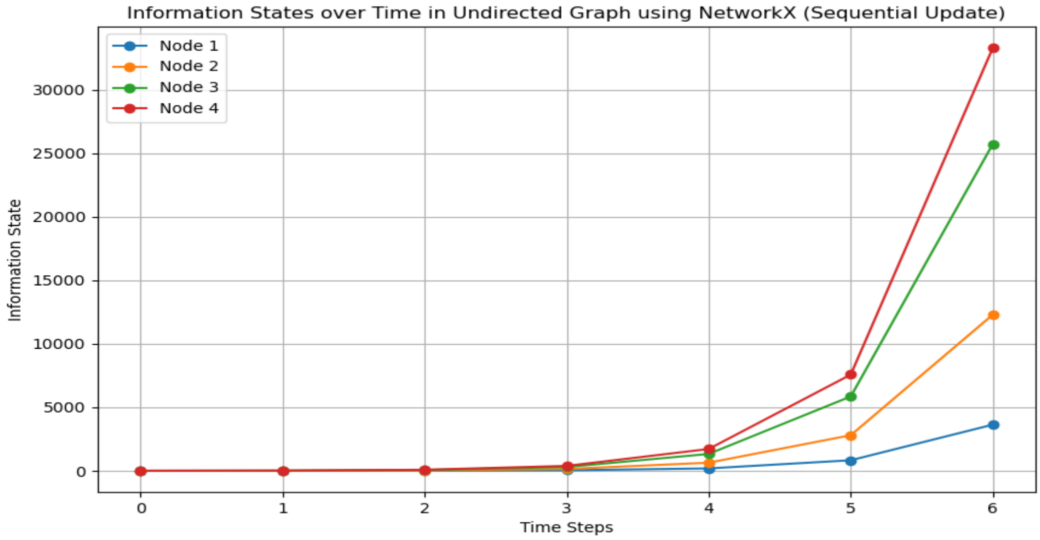
This matrix $\mathbf{P}$ is a lower triangular Toeplitz matrix. The general solution to this First-order system of linear differential equations is given by

$$\boldsymbol{\theta}(t) = \begin{pmatrix} \theta_1 e^t \\ (\theta_1 + \theta_2) e^t \\ \frac{1}{2}(\theta_1) e^t t^2 + \theta_2 e^t t + \theta_3 e^t \\ \frac{1}{6}(\theta_1) e^t t^3 + \frac{1}{2}(\theta_1) e^t t^2 + \theta_2 e^t t + \theta_3 e^t \end{pmatrix}$$

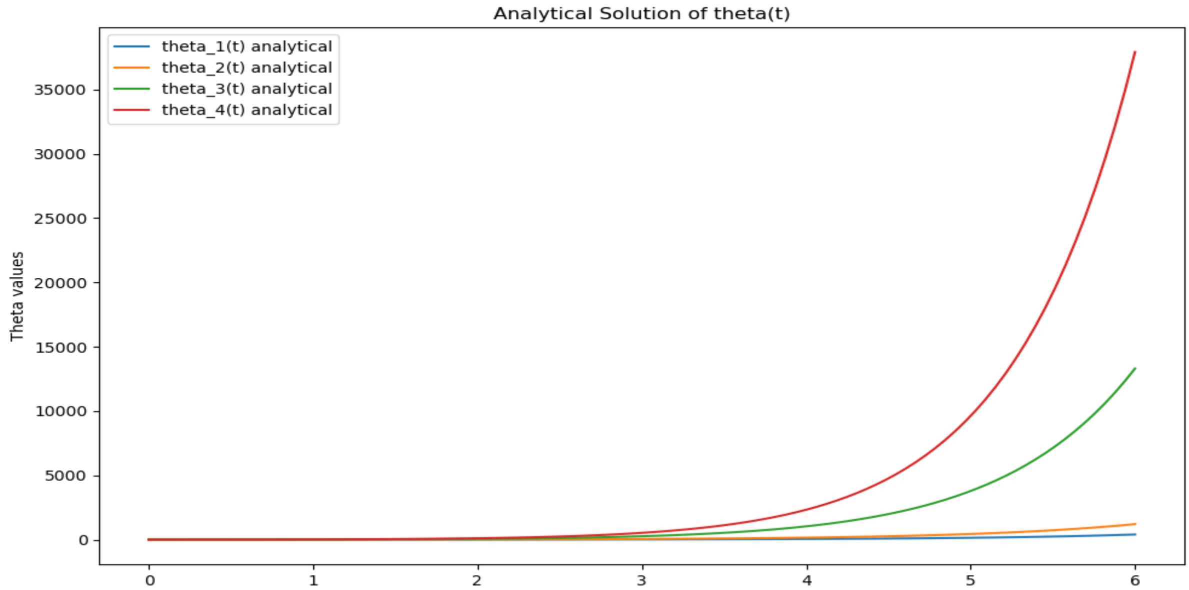
The solution operator for this system is

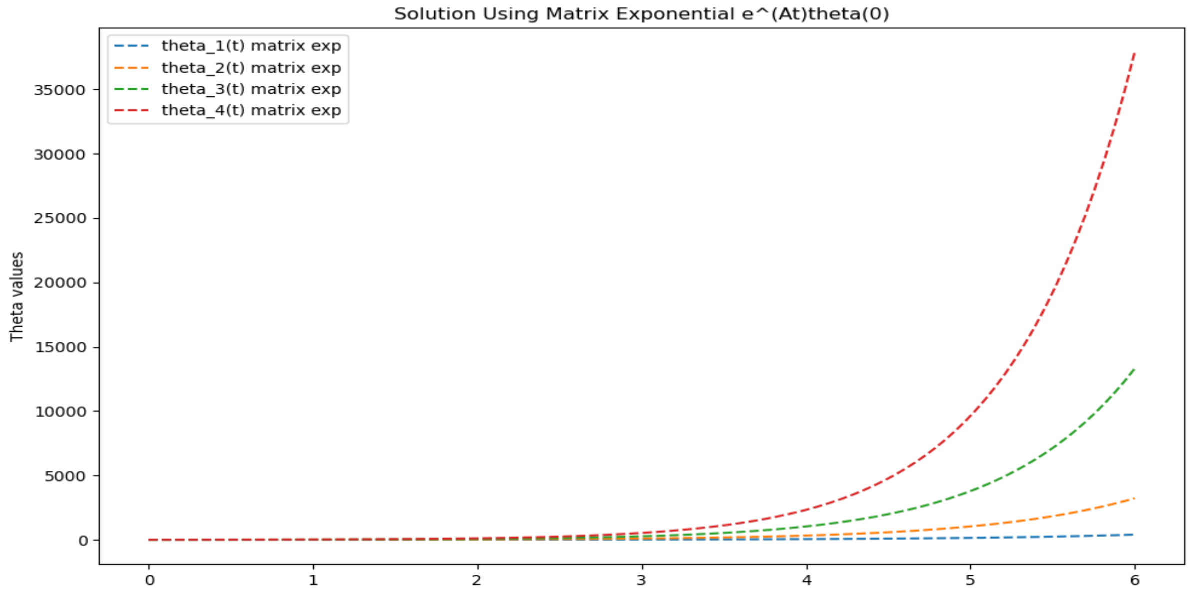
$$\Psi(\boldsymbol{\theta}(0); t) = e^{\mathbf{P}t}\boldsymbol{\theta}(0)$$

which is a strongly continuous one-parameter semigroup, known as a C_0-semigroup. We can run some simulations to validate these results. The first figure is a direct simulation of the transfer of information between the nodes on a graph structure. The nodes on the graph have initial states $\{1, 2, 3, 4\}$ and the transfer function is know $\psi_i(\theta_i) = \theta_i + \theta_{i-1}$ and it is simulated over time for $0 \leq t \leq 6$. At $t = 0$ the initial (non-state) information $\theta_0 = 0$.



Here is the results from the analytical solutions that have been found. From the solution $\boldsymbol{\theta}(t)$ and $\Psi(\psi)$ we have an alignment with the data





If the state of each node in a network is governed by its own ODE (or PDE map), then working with a dynamical process on a network entails examining a potentially large system of coupled ODEs or PDEs. Network structure can affect dynamical processes on a network, but also that dynamical processes can affect the dynamics of the network itself. These propositions outline and breakdown the existence of linear or nonlinear operators Ψ that can be recovered from observations of ψ. Each proposition can be deterministic or stochastic $\xi_i = 0$ or $\neq 0$.

0.1.4 Questions needing to be answered first...

Proposition 1. *There exists a linear or nonlinear operator* $\Psi : \mathcal{H} \to \mathcal{H}$*, on a non-time varying graph* $G = (V, E)$ *that takes the initial state* $\boldsymbol{\theta}(0)$ *at time* $t = 0$ *to* $\boldsymbol{\theta}(t)$ *at time* $t > 0$.

$$\Psi(\boldsymbol{\theta}(0); t) = \boldsymbol{\theta}(t)$$

such that Ψ *is the solution operator to the differential equation in the form*

$$\begin{aligned}\psi(\boldsymbol{\theta}) &= \frac{\partial \boldsymbol{\theta}}{\partial t} + \xi_i \\ &= \Phi_i\left(\theta_i(t); \{\theta_j(s) | (i,j) \in V; s \leq t\}\right) + \xi_i\end{aligned}$$

where ξ_i *represents external stochastic influences. The operator* Ψ *can be recovered from known* $\psi_i(\theta_i)$ *on agents* v_i *and* v_j*.*

Proposition 2. *There exists a linear or nonlinear operator* $\Psi : \mathcal{H} \to \mathcal{H}$*, on a time varying graph* $G(t) = (V(t), E(t))$ *that takes the initial state* $\boldsymbol{\theta}(0)$ *at time* $t = 0$ *to* $\boldsymbol{\theta}(t)$ *at time* $t > 0$.

$$\Psi(\boldsymbol{\theta}(0); t) = \boldsymbol{\theta}(t)$$

such that Ψ *is the solution operator to the differential equation in the form*

$$\begin{aligned}\psi(\boldsymbol{\theta}) &= \frac{\partial \boldsymbol{\theta}}{\partial t} + \xi_i \\ &= \Phi_i\left(\theta_i(t); \{\theta_j(s) | (i,j) \in V; s \leq t\}\right) + \xi_i\end{aligned}$$

where ξ_i *represents external stochastic influences. The operator* Ψ *can be recovered from known* $\psi_i(\theta_i)$ *on agents* v_i *and* v_j*.* Ψ *is the solution operator to a differential equation.*

Proposition 3. *There exists a linear or nonlinear operator* $\Psi : \mathcal{H} \to \mathcal{H}$*, on a time-varying graph* $G(t) = (V(t), E(t))$ *that takes the initial state* $\boldsymbol{\theta}(0)$ *at time* $t = 0$ *to* $\boldsymbol{\theta}(t)$ *at time* $t > 0$.

$$\Psi(\boldsymbol{\theta}(0); t) = \boldsymbol{\theta}(t)$$

such that Ψ *is the solution operator to the differential equation in the form, with unknown or partially known transfer function* ψ_i

$$\begin{aligned}\psi(\boldsymbol{\theta}) &= \frac{\partial \boldsymbol{\theta}}{\partial t} + \xi_i \\ &= \Phi_i\left(\theta_i(t); \{\theta_j(s) | (i,j) \in V; s \leq t\}\right) + \xi_i\end{aligned}$$

where ξ_i *represents external stochastic influences. The operator* Ψ *can be recovered from observations of* $\psi_i(\theta_i)$ *on agents* v_i *and* v_j*.* Ψ *is the solution operator to a differential equation, that should be inferred from data.*

The symbol Φ just highlights that some interactions may have memory effects and be non-markovian. These propositions could be further broken down in situations where the noise terms $\xi_i = 0$, deterministic, markovian, non-markovian, iid, non iid, additive or multiplicative. The definitions and propositions, while suggestive, lack rigorous mathematical formalism in the current state.

0.2 Why Transfer Functions?

The idea is to formalise local interactions through vertex-specific transfer functions ψ_i, which capture how each agent processes and transmits information to its connected nodes. By observing these local interactions, we aim to understand or even predict the global dynamics of information change for the entire network G, which are governed by the operator Ψ. If Ψ can be effectively characterised, it could be used to predict future information states of the network under various scenarios, hopefully making it a powerful tool for forecasting and planning in complex environments. Some other considerations to the usage of vertex functions is given below:

1. Each node's transfer function can be studied and modified independently, if needed. This means the operator can be used to study how a specific node changes an input.

The mapping for the vertex function can be examined through a dynamical systems perspective. Looking at iterations of the information and how information is transformed in the network as $t \to \infty$ and is given by the mapping

$$\psi_i : \Theta \to \Theta.$$

As $t \to \infty$ the information contained in the agent v_i is transmitted to connecting nodes v_j across the edges E. The information diffusion in the network is a dynamic process as can be expressed as an iterated mapping of transfer functions describing how information is manipulated over time

$$\left(\psi_1 \circ \psi_2 \circ \cdots \circ \psi_{|V|}\right)(\theta_i),$$

which allows for identification of steady states in the network, cycles, attracting sets and other types of classifications in dynamical systems theory. Additionally, the theory of groups and other algebraic structures formed by the iterative mapping can be taken advantage in special cases, for example, C_0-semigroup operators and transition functions in the case of Markov Random Fields.

A steady state in this form would look like

$$\frac{\partial \theta_i}{\partial t} = 0$$

0.2.1 Why choose $\psi_i : \Theta \to \Theta$?

The primary purpose of the vertex function ψ in this framework is to handle the dynamics of information state transformation among agents within the graph, we may not know ψ. The transfer functions ψ_i for each agent v_i directly modifies the information state θ_i, aiming to capture the unique information-processing characteristics of each agent.

0.2.2 Encoding a Traditional SIR Model into this framework, without any sort of tuning.

One may have noticed that if we want to employ the SIR model from epidemiology, we can not do this for a network with more than three nodes in this framework, directly. Consider a non time-varying graph $G = (V, E)$ with $|V| > 3$. Further to this the space of vertex functions on G, given by $\mathcal{H}(V) \cong \mathbb{R}^{|V|}$ has dimension $|V|$, this means that the vertex function containing the transfer functions on the network has dimension $|V| > 3$. The SIR model will not fit in it's current state, why is this?

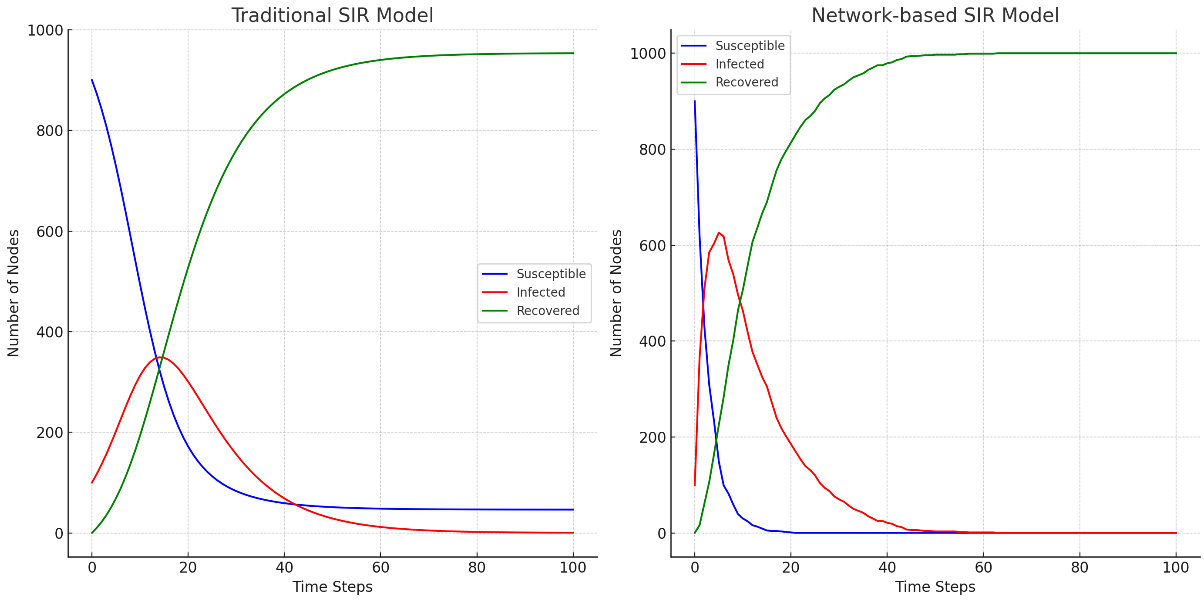
The SIR model is a compartmental model used in epidemiology to describe the spread of a disease through a population on a global scale. It divides the population into three compartments: susceptible (S), infectious (I), and recovered (R).

At a global level this will not fit, since we are taking an individualistic approach, there is simply more transfer functions than information. However, all is not lost! We have, yet, one more method at our disposal. The space of all information allowed in the network (or in this case states). Θ now becomes the set $\{S, I, R\}$ and agent information $\{\theta_1, ..., \theta_{|V|}\}$ take values in Θ. The vertex functions can provide a granular approach, and are given as such:

$$\psi(\boldsymbol{\theta}) = \begin{pmatrix} \psi_1(\theta_1; \{\theta_j | j \in N(1)\}) \\ \psi_2(\theta_2; \{\theta_j | j \in N(2)\}) \\ \vdots \\ \psi_{|V|-1}(\theta_{|V|-1}; \{\theta_j | j \in N(|V|-1)\}) \\ \psi_{|V|}(\theta_{|V|}; \{\theta_j | j \in N(|V|)\}) \end{pmatrix},$$

$$\psi_i(\theta_i; \{\theta_j | j \in N(i)\}) = \begin{cases} (I, t_0) & \text{if } \theta_i = S \wedge \exists j \in N(i) : \theta_j = I \\ (R, t_i) & \text{if } \theta_i = I \wedge t_i \geq t_\infty \\ (I, t_i + 1) & \text{if } \theta_i = I \wedge t_i < t_\infty \\ (R, t_i) & \text{if } \theta_i = R \\ (S, t_i) & \text{otherwise} \end{cases},$$

where t_∞ is the duration of time an agent v_i stays infected. This is compared to the classical approach using differential equations:



What would Ψ even do in this case? Given a node v_j that is introduced in the network, Ψ wouldn't tell us:

- The exact path an infection takes through the network.

However, it should tell us:

- How information (infection) is likely to be transformed as it propagates through the network - so given some input state from Θ then it should produce the evolution of the state - as the analytical SIR model does. The idea is that Ψ would provide the same results as the differential equation solutions in the classical SIR model.

0.2.3 Related to Multi Agent Systems?

The ideas presented here have been informed by Multi - Agent Systems. However, there is a separation. There is in-fact an underlying multi agent system, the idea being that we want to infer some information about a signal on the network. We do not care about the underlying agent behaviours and do not wish to infer information about the agents v_i. We aim to discover operations that are controlling the spread of this signal through G using methods grounded in Graph Signal Processing, Machine Learning/Probability and Statistics, Systems Theory, Group Theory and Graph Theory. It emphasises a holistic understanding of the network's behaviour in terms of signal spread, rather than the specific actions or strategies of individual agents (though you could infer this with post processing). Here are some differences:

Multi Agent Systems Theory:

- Concentrates on the behaviour, decision-making, and interaction of autonomous agents.
- Includes aspects like negotiation, cooperation, strategy, and adaptation among agents - explicitly.
- Emphasises understanding how individual agent behaviors and interactions lead to emergent system properties.

General Diffusion in Networks:

- Focuses on the properties and dynamics of the signal itself as it propagates through the network.
- Aims to understand and manipulate the network and signal properties, not the agents.
- Provides a mathematical understanding and framework to easily apply machine learning methods to apply in areas such as information diffusion.

Most crucially, how is this different to Graph Signal Processing?

Graph Signal Processing

- Primarily focuses on the analysis and manipulation of signals located on the nodes of a graph using techniques derived from signal processing - namely Fourier Transforms

General Diffusion in Networks

- Extends beyond traditional GSP by incorporating operators that act on vector-valued functions representing dynamic information states of a network and aims to develop a method for describing nonlocal operators acting on G.
- Explicitly aims to set up a framework for Deep learning (Deep Operator Networks, Operator Learning/Discovery) to approximate these complex mappings.
- Focuses on using machine learning not just to predict outcomes but to construct the underlying dynamic systems within the network, a concept akin to learning solution operators in Partial Differential Equations and Physics-Informed machine learning, but applied to network science and possible image diffusion.

0.3 Research Problem Statement

The idea is to produce a nice framework for allowing an intuitive application of machine learning to infer the operator Ψ from observations of the vertex function $\psi(\boldsymbol{\theta})$. To start things off, a result in Neural Network Theory is that a neural network with a single hidden layer can approximate accurately any nonlinear continuous operator (indeed any continuous function too). This motivates the idea behind

Deep Operator Networks and Operator Learning in 2019 and 2020 respectively. These systems are typically applied to learning solution operators to partial differential equations in mathematics.

Problem 1. We now have to represent Ψ as a **General Operator** acting on the vertex function ψ. From a collection of observations of transfer functions $\{\psi_1, ..., \psi_n\}$ how do we recover the operator Ψ? Well proposition 3 is useful, but it is not solved...

This is an inverse problem, and an ill-posed one. The goal is to learn a parametrised operator $\hat{\Psi}(\cdot, t; \omega)$ with a neural network such that $\hat{\Psi} \approx \Psi$, where $\omega \in \Omega$ are parameters of the neural network. The neural network model $\hat{\Psi}$ is a Graph Neural Operator or a variant of a Deep Operator Network known as a Deep Graph Operator. These models belong to a class of machine learning models designed to learn mappings between infinite-dimensional function spaces and generalise classical neural networks.

To give some intuition. Imagine there is a network $G = (V, E)$. Each node is an agent on the graph which has many decision making rules. We observe how signals spread around the network from observations of the agent sending signals to other agents in the same network. We can not possibly decode how the agents think. Each agent is a non-deterministic and possibly chaotic system. However, we observe that there is a functional mapping between agents in the graph. We can measure how the signals change over time t. A general functional mapping Ψ is observed. The aim is to use many observations about agent interactions on G and possibly re-construct a parametrised functional representation.

Once we have a functional representation for $\hat{\Psi} = \partial_t \psi(\boldsymbol{\theta}) + \xi_i$, we inject a new agent v_i^* into the network. We know the internal workings of this agent and the information it will spread, θ_i^*. We now want use $\hat{\Psi}$ to predict how a signal from v_i^* will diffuse in the network, how this signal is manipulated and what happens to the signal. This combined with other methods for tracking where information goes (edge dynamics) should provide a well rounded view of information diffusion in networks.

Problem 2. To set this problem up, we can view this problem through the lens of Supervised Learning. We determine an underlying mapping $\Psi : \mathcal{H} \to \mathcal{H}$ from samples of input/output pairs representing information diffusion on a network G,the I/O pairs are given as

$$\{\theta_n, \Psi(\theta_n)\}_{n=1}^{|V|}, \quad \theta_n \sim \mu$$

where, the input is the information states of each agent at some time t and the output, the information state after the diffusion operator Ψ has been applied to the input θ. The function μ is a probability measure supported on the space of all possible information states Θ for any agent θ_n. In this set up we note that the size of vertex set $|V|$ is the dimension of the data.

Problem 3. (Generalisation of Problem 2) Let $\mathcal{U}$ and $\mathcal{V}$ be a Separable Hilbert Spaces of vector valued functions:

$$\mathcal{U} = \left\{u_n : \Theta \to \mathbb{R}^{d_{\text{in}}}\right\}$$
$$\mathcal{V} = \left\{\nu_n : \Theta \to \mathbb{R}^{d_{\text{out}}}\right\}$$

Given $\{\theta_n, \Psi(\theta_n)\}_{n=1}^{|V|}$ we seek to determine an approximate operator $\hat{\Psi} : \mathcal{U} \to \mathcal{V}$ from within a family of parametrised functions

$$\hat{\Psi} : \mathcal{U} \times \Omega \to \mathcal{V}, \quad \Omega \subseteq \mathbb{R}^p$$

where Ω is the parameter space. From the parameter space it is though an optimal choice of parameter $\omega^* \in \Omega$ can be found.

0.4 Some considerations

While promising, the approach requires significant data to accurately train the model and substantial computational resources to handle real-time network changes. Further research will focus on optimising these aspects to enhance feasibility and extend applicability to networks with sparse data availability. As well as continuous revisions and additions to the underlying framework outlined in this passage.

If Θ is not well-defined or if the mapping ψ_i does not preserve certain necessary properties of information states (like continuity, if required), then the application of Ψ might not behave as expected. The stability, convergence, or even the well-posedness of problems involving $\hat{\Psi}$ might not be guaranteed without additional constraints or properties, this would need to be explored in future research.