Review Article

An Overview of Methods for Activity Graph Study of Movements

Abstract: Graph-based data structures have emerged as a fundamental tool across a wide range of applications, offering an intuitive and powerful way to visualize, model, and analyze complex information systems. One notable application is the study of discrete movement patterns observed between defined key points or locations. By representing these movements as graph structures, underlying trends, identify benchmarks, and establish predictive models can be uncovered. Such analyses are crucial for understanding and modelling the behaviours of various populations, including individuals with movement or decision-making impairments, where tailored interventions or designs might be required. This paper provides an overview of graph-based methodologies employed in the literature to analyze and model movement data. Specifically, it focuses on three techniques: a) Markov Chains, which model probabilistic transitions and sequence dependencies within the movement data; b) PageRank, originally devised for web-page ranking but adapted here to evaluate importance of nodes within a movement graph and c) Graph Signal Processing, as an approach that facilitates the analysis of signals distributed over graph structures to detect patterns and anomalies. Each method is detailed and demonstrated through illustrative examples, highlighting its unique contributions to the study of movement patterns. **Keywords:** graph data structure, movement representation, graph analysis, Markov chains, PageRank algorithm, graph signal processing

1 Introduction

Graph representation and visualization of data as nodes (or vertices) and edges (or links) has been pivotal for capturing inter-nodal relationships and applying discrete mathematics to diverse domains. Their intuitive representation of data and in particular in capturing relationships between the nodes through various interpretations which can be associated to the connecting has found many applications. Formally, a graph $\mathcal{G}$ is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of vertices and $\mathcal{E}$ represents the set of edges connecting the vertices, e.g. see (Diestel, 2017).

Robotics and sensing are two fields that have significantly benefited from graph theory. For example, in multi-robot coordination, graphs are used to model and solve problems involving multiple robots working together. Nodes can represent robots or tasks, and edges can represent communication links or dependencies. Graph-based algorithms help in task allocation, coordination, and collision avoidance, (Pistilli & Averta, 2023). Graphs are used to model the relationships between objects and the robot's end-effector. This helps in planning and executing complex manipulation tasks, such as assembling parts or handling objects, (Wan et al., 2024, Keshmiri & Payandeh, 2024,). In sensor networks, graphs simplify system modelling by treating sensors as nodes and their communication links as edges. This representation enables optimization in network design, data routing, and fault tolerance. Graph theory helps in optimizing network topology, data routing, and fault tolerance, (Ghosh et al., 2024, Rasoulidanesh & Payandeh, 2020). Graphs are used to represent the structure of images or signals. Graph-based algorithms help in tasks such as image segmentation, feature extraction, and pattern recognition (Bran et al., 2020). In image processing, graph cuts address segmentation challenges by framing them as energy minimization problems. Here, each pixel (or region) becomes a graph node, with edges linking nodes based on feature similarity (e.g., color, intensity, texture). By computing the minimum cut (or equivalently, solving a max-flow problem), the algorithm separates the graph (and hence the image) into distinct segments. This method is highly effective for interactive segmentation tasks where the user provides seeds for foreground and background regions (Boykov & Jolly, 2001). Normalized cuts view the image as a weighted graph where pixels (or superpixels) are connected by edges weighted by feature similarity. The segmentation problem is then cast as partitioning the graph such that the cut cost (i.e., the sum of weights of the edges being removed) is minimized relative to the total connection within each segment. This criterion is solved via spectral analysis (namely, the eigen-decomposition of the graph Laplacian) to obtain a low-dimensional embedding of the data that naturally lends itself to clustering (Shi & Malik, 2000).

Beside applications in sensing and robotics, graphs are also used in modelling movements and activities of people. In skeleton-based action recognition, the human body is modelled as a graph. Each node represents a joint (such as the wrist, elbow, or knee), and the bones connecting these joints form the edges. When extended over time, these graphs become spatio-temporal representations that encode the human motion sequence. The natural anatomical structure sets up a fixed graph topology where the distance and orientation between joints are critical, (Yan et al. , 2018, Mohsin & Payandeh, 2022). In many real-world scenarios (such as surveillance footage, sports analytic, or public event monitoring)the focus is on understanding not only what an individual is doing but also how groups interact. Graphs in this context are constructed by treating each person as a node while edges capture relationships such as spatial proximity, directional movement, or interaction cues. Edges may represent various aspects of interpersonal interactions (for example, distance thresholds, gaze direction, or synchronized motion) (Choi & Savarese, 2012). When monitoring human movement (say, in crowded public spaces or transit hubs) trajectory data become a rich source of information. Here, trajectories are modelled as graphs where nodes represent key positions (or even short segments) along an individual's path, and edges capture transitions between these points.By creating a graph of trajectories, clustering and community detection methods can uncover common movement patterns, preferred paths, or anomalous behaviour (Lee et al., 2007).Beyond immediate movement trajectories, graph methods can model longer-term social interactions, mapping the network of contacts, encounters, or movements of groups over time. In this perspective, individuals are nodes in a social graph, and edges represent recurring interactions or physical proximity events.These graphs can evolve over time as relationships form or dissolve based on spatio-temporal proximity (Helbing & Molnar, 1995).

Graph-based representations offer a versatile and robust framework for investigating the intricate and interconnected patterns of human movement. In trajectory analysis, graphs serve as a powerful tool to cluster and identify prevalent pathways, uncover patterns, and detect anomalies with precision. Meanwhile, in the study of social interactions, graphs enable the modelling of relational dynamics that underpin collective behaviour, offering insights into how individuals and groups interact within complex systems. These methodologies not only enhance our ability to interpret and predict individual movement patterns but also shed light on the emergent properties of group behaviour and the underlying principles of social organization in dynamic and often unpredictable environments.

This paper explores three distinct graph-based methods designed to model, analyse, and interpret movements within a monitored setting. These techniques are especially relevant in applications such as modelling the behaviours of individuals with physical or decision-making challenges who are confined to a specific living environment. By leveraging these methods, researchers can gain a deeper understanding of movement dynamics and their implications in specialized scenarios, paving the way for improved interventions and designs of assistive systems.

The structure of this paper is as follows: Section 2 provides an in-depth exploration of the interpretation of Markov chains as applied to the analysis of movement graphs; Section 3 examines the extension of the PageRank algorithm for the modelling and analysis of movement graphs, highlighting its utility in identifying key nodes and pathways; Section 4 highlights the application of Graph Signal Processing, illustrating its potential for analysing movement data encapsulated within graph structures; finally, Section 5 concludes with a summary of findings and a discussion of future research directions.

2 Markov Chains

Markov chains are general frameworks which can be used to establish a graphical representation between various state of the system. Their graphical representations also establishes the dependencies on reaching to a current state given all or part of the past history of the connecting states to the current state. In this representation, each state of the system is represented as a node in a graph with having either bidirectional or directional edges connecting these nodes from or to the other nodes. The conditional dependencies in terms of conditional probabilities between the connected states are represented with the weighted edges, represented as vectors for each edges (Stroock, 2014).

In matrix form, the graph can be represented as weighted adjacency matrix which also represents the transition probability matrix of the Markov chain. Such matrix can be structured to have either row-stochastic property where each row sums to 1 (i.e. weights of outgoing edges from a node sum to 1 which also implies the weights represent the probability of moving from a given state to another) or as column-stochastic property where each column sums to 1 (i.e the weights of incoming edges to a node sum to 1 which also implies the weights represent the probability of arriving at a state from another). The graph may be strongly connected if every state is reachable from every other state (directly or indirectly). If some states are unreachable, the graph is not strongly connected, and there may be absorbing states (nodes with no outgoing edges) (Privault, 2013).

Markov chains can be applied in several ways in order to be able to analyse an activity graph.

For example, they can be used as *sequential pattern mining* to model sequences of movement activities. Each state represents a specific location, activity, or behaviour, and transitions represent the probability of moving from one state to another. For example, it can be used to predict the next location of an individual based on their current location and past trajectory. Another example of application of Markov chains can be in *behavioural prediction and anomaly detection approaches* where typical movement behaviours form a predictable pattern of state transitions. Deviations from these patterns can indicate anomalies.

As an example, let us consider a living space consisting of four observation states labelled as Living Room (L), Kitchen (K), Bedroom (B) and Bathroom (R). Let us define the transition probabilities of two persons A and B between the states of the system (rooms) as row stochastic matrices P_1 and P_2, where P_{ij} represent probability of moving from state i to state j, or (Payandeh, 2022):

$$P_1 = \begin{bmatrix} 0.1 & 0.6 & 0.2 & 0.1 \\ 0.3 & 0.3 & 0.3 & 0.1 \\ 0.2 & 0.1 & 0.6 & 0.1 \\ 0.4 & 0.2 & 0.2 & 0.2 \end{bmatrix}; \qquad P_2 = \begin{bmatrix} 0.3 & 0.4 & 0.1 & 0.2 \\ 0.2 & 0.3 & 0.4 & 0.1 \\ 0.1 & 0.2 & 0.6 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.4 \end{bmatrix}. \tag{1}$$

In sequential pattern mining, the goal is to predict the next room the person is likely to move to. For example, given the current state L, based on P_1, the next likely state is K with the conditional probability of $Pr(K|L) = 0.6$. However, transitions to other states from the exiting current state L also have the accumulated probability of 0.4. Following this initial transition, the states L, K and B will have equal likelihood to transition to.

Let us now analyse a hypothetical sequence of state transitions in order to find frequent sequential patterns. The focus is on identifying patterns with high transition probabilities. Consider the follow sequence of transitions recorded over a period of time: $L \to K \to B \to B \to K \to L \to R \to L \to K \to B$. This sequence describes the movement of a person between rooms over time. From the transition matrix P_1, we can identify high-probability transitions to be: From $L \to K : Pr(K|L) = 0.6$, $B \to B : Pr(B|B) = 0.6$ and $R \to L : Pr(L|R) = 0.4$. Given the recorded sequence, we can now extract frequent sub-sequences as:

$$\begin{aligned} &\text{sub-sequence 1:} L \to K \to B \to B = Pr(K|L) \cdot Pr(B|K) \cdot Pr(B|B) = 0.6 \cdot 0.3 \cdot 0.6 = 0.108, \\ &\text{sub-sequence 2:} R \to L \to K = Pr(L|R) \cdot Pr(K|L) = 0.4 \cdot 0.6 = 0.24, \\ &\text{sub-sequence 3:} K \to B = Pr(B|K) = 0.3, \end{aligned}$$

where the following interpretation can be extracted from the above sub-sequential patterns: a) *frequent patterns*- $L \to K \to B \to B$ shows repeated visits to B suggest node B as a primary destination; $R \to L \to K$ shows movement from R to L and then K indicates certain routine at a given observation instance and $K \to B$ can suggests frequent visit of node B after being observed at node K; and b) *anomalous patterns*- transition with low probabilities (e.g. $B \to K$) might represent anomalies or less common activities. As an example of application of the above sequential pattern mining is in optimizing energy usage (e.g., turn on lights sequentially) given the expected movement sequence: $R \to L \to K$.

Given independent Markov graphs represented by transition matrices defined in equation (1), one can study various joint behaviours activities for as *interaction dynamics* study. For example, the long-term behaviours for both person can be obtained by solving the eigenvalue problem of the form $\pi P = \pi$ where π corresponds to eigenvector of the transition matrix P, or:

$$\text{For } P_1, \pi_1 = [0.227, 0.281, 0.380, 0.111],$$
$$\text{For } P_2, \pi_2 = [0.272, 0.252, 0.351, 0.175],$$

where for example, we can estimate the joint probability of both person being in state L as: $P_{joint}(L) = \pi_1(L) \cdot \pi_2(L) = 0.061$. Following similar computation, if both person have high probabilities of being in the same room simultaneously, check for potential conflicts. In this example, we can estimate the state K has highest probability which suggest frequent shared presence of both person. Such model also as stated above allows to track observed movements and compare them to expected probabilities. For example, if the first person stays frequently in state R despite $\pi_1(A) = 0.2$, investigate health or behavioural changes.

3 PageRank Method

Given a directed or undirected graph that represents the transition of movements associated with activities between various observation points, the PageRank(PR) algorithm and its various extensions, Brin & Page, 1998, Gleich, 2015), can be utilized as a tool in order to analyse and interpret the graph of activity of movements. For example, for the case of a graph representing of the movements of a person between various rooms in a living space, the PR can be used to assign a score to each room, representing the likelihood of the person being in each room given the structure of the activity graph. In this case, each room is represented as a node in the graph. Each doorway or paths between rooms is represented as an edge connecting nodes in an undirected manner. This also implies that the person can have access to go between the rooms in an unobstructed way.

Using the movement of the person between the room analogy, let us assume that if there is a path between two rooms, there is an equal likelihood of moving between them in either directions. Let N be the number of rooms (nodes on the graph) and let us also define a transition matrix P where P_{ij} represents the probability of moving from room i to room j. In the pagerank definition, P represents a column stochastic matrix and it is based on incoming links. This implies that for each node, P represents the probability of arriving at a node i from nodes j. Thus P must be column-stochastic to correctly model these probabilities. For an undirected graph, the probability of moving from room i to any adjacent room j can be defined as: $P_{ij} = 1/deg(i)$ for neighbouring node j, where $deg(i)$ is the degree of node i (i.e. number of paths (edges) connected to it).

The PR algorithm offers a computational method to estimate the stationary distribution of the above definition of the Markov chain, providing a ranking of rooms based on how likely the person is to be in each room over the long term observation (Payandeh & Chiu, 2019). The classical definition of the ranking of a room i, i.e. $PR(i)$ is defined in the following equation which can iteratively be updated:

$$PR(i) = \sum_{j \in M(i)} \frac{PR(j)}{deg(j)} \tag{2}$$

where, $PR(j)$ is the pagerank of the adjacent room to i, $deg(j)$ is the degree of room j and $M(i)$ is the set of rooms directly connected to i. The above equation is stating that the probability of the person being in room i is weighted by the sum of the probability of the person being in the adjacent rooms connected to room i.

Equation (2) can further be modified as:

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in M(i)} \frac{PR(j)}{deg(j)} \tag{3}$$

where, d is the *damping factor* (typically 0.85), representing the probability that the person continues moving verses starting a new random transition, $\frac{1-d}{N}$ is the *random jump* factor capturing the possibility that the person might choose a room randomly instead of following the graph structure and N is the number of nodes (rooms) in the graph. As an example, let us consider three rooms A, B, C such that they are connected in the following order $A \to B$, $B \to C$ and $C \to A$. Given the initial rank probability for each room to be $PR(A) = PR(B) = PR(C) = 1/N = 1/3$, equation (3) can be iteratively updated until the value stabilized.

The process of calculating the PageRank values for nodes in the graph (e.g. rooms) can also be written as a matrix-vector multiplication problem:

$$\overrightarrow{PR}^{i+1} = d \cdot P \cdot \overrightarrow{PR}^{i} + (1+d) \cdot \vec{e}, \tag{4}$$

where, $\overrightarrow{PR}$ is the PageRank vector, holding the PageRank values for each room, P is the transition matrix of the graph and $\vec{e}$ is a vector which for now we assume represents a uniform distribution over all possibilities of $\frac{1}{N}$ values. The localized attention vector, $\vec{e}$, can be used to represent the focus of the random walker on a region of the graph. The term $\frac{1-d}{N}$ in equation (3) as originally defined by (Brin & Page, 1998) can represent random exploration behaviour.

For the cases when we do have some prior knowledge about the stochastic learned movement history of the person which is being observed, a modified PageRank algorithm in a form of a Weighted PageRank can be used where weights are introduced to each edge of the graph, Xing & Ghorbani, 2004. This adjustment reflects the idea that certain connections (edges) between nodes (rooms, in this context) may have different levels of importance or probability, which influences the rank distribution. Weighted PageRank assigns a score to each room based not only on the connections but also on the strength or significance of each connection, represented by the weight of each edge.

Rationale for Weighted PageRank can be that in many real-world networks, connections between nodes vary in importance such as: a) distance between rooms: a short hallway might indicate a strong connection between two rooms, while a long or narrow passage might imply a weaker link; b) frequency of movement: in human movement patterns (such as elderly), some paths are more frequently travelled, making those edges effectively *stronger* in the network; and c) access constraints: some connections might be constrained by factors such as doors, cluttered passages or presence of furnitures that are rarely used or areas with restricted access.

An interpretation of equation (4) for weighted PageRank implementation is shown in the following Gleich, 2015, Payandeh & Chiu, 2019. Let us again define three rooms with the following connectivities: $A \longleftrightarrow B$ with weight 3, $B \longleftrightarrow C$ with weight 2 and $C \longleftrightarrow A$ with weight 1. This will results in the weighted transition matrix P as:

$$P = \begin{bmatrix} 0 & 3/5 & 1/3 \\ 3/4 & 0 & 2/3 \\ 1/4 & 2/5 & 0 \end{bmatrix}, \text{with } PR^0 = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \text{ as the initial distribution vector.}$$

where we can compute the following terms associated with the equation (4) with $d = 0.85$. The scaled matrix-vector multiplication is:

$$d \cdot P \cdot \overrightarrow{PR}^1 = 0.85. \begin{bmatrix} 0 & 0.6 & 0.33 \\ 0.75 & 0. & 0.667 \\ 0.25 & 0.4 & 0. \end{bmatrix} \begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix} = \begin{bmatrix} 0.347 \\ 0.391 \\ 0.259 \end{bmatrix}$$

with the teleportation term:

$$(1-d) \cdot \vec{e} = (1-0.85) \cdot \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.05 \\ 0.05 \end{bmatrix}$$

which iteratively converges to the results in room B to have the highest room ranking at steady state value.

4 Graph Signal Processing

Analyzing the movements and activities of a person within a living space involves leveraging graph modelling and algorithms to extract meaningful insights from spatial and temporal data. In this section we present an overview of how the notion of graph signal processing (GSP),(Shuman et al.,2013, Ortega et al., 2018, Leus et al., 2023), which can be applied to study the graph representation of movements and activities. Similar to the previous examples, consider again a weighted graph where: Nodes (N) represent specific areas within a living space; Edges (E) represent the paths between these areas, with weights reflecting the frequency or difficulty of traversal between nodes and also Node Weights W (e.g. Dwelling Times) represent the average time a person spends at each node (e.g., 30 minutes in the Living Room, 10 minutes in the Kitchen), Or a graph $G = (N, E, W)$ where $N = \{n_1, n_2, n_3, n_4\}$ with node attributes $f(n_i)$.

GSP enables the analysis of signals defined on the nodes of a graph, such as dwelling times, to also represent as graph signals that can vary across different nodes. Examples of application of GSP which can be used as a part of study of activity graph of movements can be: a) *Graph Filtering* which is similar to filters in traditional signal processing, can smooth or enhance signals across nodes. For example, Smoothing Filter can be used to identify zones with similar dwelling times by smoothing out abrupt differences, or high-pass filter can be used to detect anomalies where dwelling times deviate sharply from surrounding areas (e.g., spending excessive time in unusual locations); b) *Graph Fourier Transform (GFT)* which decomposes the graph signal into components associated with different frequencies. This can be low-frequency components that can represent routine and stable patterns, such as regularly spending 30 minutes in the Living Room or, High-Frequency Components that can indicate irregular behaviours, such as sudden spikes in time spent in the Bathroom. This can be used for anomaly detection and *Spectral Analysis* of the graph's Laplacian eigenvalues helps capture movement dynamics and clustering tendencies. For example, to identify clusters of nodes where a person transitions frequently with similar dwelling times (e.g., Living Room-Kitchen cluster).

Let the graph example of movements be defined as above where nodes $n_i, i = 1, \cdots, 4$ represents *L,K,B,R* respectively and the weight on the connecting edges are defined as: $w_{(1\leftrightarrow 2)} = 0.35, w_{(1\leftrightarrow 3)} = 0.17, w_{(1\leftrightarrow 4)} = 0.04, w_{(2\leftrightarrow 3)} = 0.15, w_{(3\leftrightarrow 4)} = 0.1$. These weight on the connecting edges can be function of a number of factors which can be for example, the probability of each node being visited from any connecting nodes and other factors such as the time it will take to reach the other nodes or degree of difficulties of moving from one room to other. In this working example, the weights are defined as: $w_{(i\leftrightarrow j)} = (1/t_{(i\leftrightarrow j)}) \cdot p_{(i\leftrightarrow j)}$ where $t_{(i\leftrightarrow j)}$ is the time of travel and $p_{(i\leftrightarrow j)}$ is the transition probability of the edge.

An initial step for carrying-out the GSP is the definition of the graph Laplacian matrix L which is defined as $L = D - W$, where D is the degree matrix of the weighted graph (where D_{ii} is the sum of the weights of edges connected to vertex i) and W is its adjacency matrix (where W_{ij} represents the weight between node i and j). The resultant L matrix has a property where the sum of its row values is equal to 1. For the above example, the representation for the D and L matrices can be written as:

$$D = \begin{bmatrix} 0.56 & 0 & 0 & 0 \\ 0 & 0.50 & 0 & 0 \\ 0 & 0 & 0.42 & 0 \\ 0 & 0 & 0 & 0.14 \end{bmatrix}; \qquad W = \begin{bmatrix} 0 & 0.35 & 0.17 & 0.04 \\ 0.35 & 0 & 0.15 & 0 \\ 0.17 & 0.15 & 0 & 0.10 \\ 0.04 & 0 & 0.10 & 0 \end{bmatrix},$$

where the graph Laplacian can be defined as:

$$L = D - W = \begin{bmatrix} 0.56 & -0.35 & -0.17 & -0.04 \\ -0.35 & 0.5 & -0.15 & 0 \\ -0.17 & -0.15 & 0.42 & -0.10 \\ -0.04 & 0 & -0.10 & 0.14 \end{bmatrix}$$

One of the important features of the graph Laplacian is its ability to capture properties of the graph such as its connectivity and a representation of the flow of information or signals across the graph. Analogy to this representation is made in the context the Laplacian operator (e.g. shift operator) of a continuous or discrete function f, Isufi et al., 2024. The Laplacian operator measures how much the value of f at a point (sample point) deviates from its average value in a small neighbourhood around that point. It appears in various physical phenomena, such as heat conduction, wave propagation, and fluid dynamics.

One of the applications of Graph Laplacian is in graph smoothing which can be constructed as an operator of the form $S = (I + \alpha L)$, where α is the smoothing coefficient. The smoothing operator can be used in both a *direct application* (explicit diffusion iteration) of the form $f_s = Sf$ or as in *matrix inversion* form: $f_s = S^{-1}f$ where f is the vector of nodal attributes and f_s the vector value of nodal attributes after smoothing operation, Zhou & Scholkopf, 2004. Application of direct smoothing is often preferred for its simplicity and efficiency. The use of the inverse approach can provide enhanced smoothing and is used in more advanced scenarios, Dang et al., 2018.

As an example of smoothing task of the nodal value function f as defined above, i.e. $f = (30, 15, 45, 5)^T$ and $\alpha = 0.5$, one can computed the smooth nodal function f_s using *matrix inversion method* as:

$$f_s = (I + \alpha L)^{-1} f = \begin{bmatrix} 0.87 & 0.12 & 0.05 & 0.01 \\ 0.12 & 0.85 & 0.07 & 0.02 \\ 0.05 & 0.07 & 0.83 & 0.04 \\ 0.01 & 0.02 & 0.04 & 0.89 \end{bmatrix} \begin{bmatrix} 30 \\ 15 \\ 45 \\ 5 \end{bmatrix}$$

which results in $f_s \approx (27.8, 15.6, 43.2, 6.4)^T$. The smoothed signal shows stabilized dwelling times by reducing abrupt changes between the signals of connecting edges. As a comparison, the following shows a computational example using *direct application* method which can be written as:

$$f_s^{(k+1)} = f_s^{(k)} + \alpha L f_s^{(k)}; \qquad f_s^{(0)} = f$$

or,

$$f_s^{(1)} = \begin{bmatrix} 30 \\ 15 \\ 45 \\ 5 \end{bmatrix} + 0.5 \times \begin{bmatrix} 0.56 & -0.35 & -0.17 & -0.04 \\ -0.35 & 0.5 & -0.15 & 0 \\ -0.17 & -0.15 & 0.42 & -0.10 \\ -0.04 & 0 & -0.10 & 0.14 \end{bmatrix} \begin{bmatrix} 30 \\ 15 \\ 45 \\ 5 \end{bmatrix} = \begin{bmatrix} 27.6 \\ 13.4 \\ 47.6 \\ 6.2 \end{bmatrix},$$

where as it can be seen, the iterative method converges to the same result as matrix inversion but takes multiple steps. Matrix inversion is exact but computationally expensive, especially for large graphs.

Another application of the graph Laplacian is Fourier analysis of the graph. Here, one can solve for the eigenvalues and eigenvectors of the Laplacian L using $L\mathcal{U} = \mathcal{U}\Lambda$ where $\mathcal{U}$ is a matrix of eigenvectors of L and Λ is a matrix of eigenvalues. However, L is a symmetric matrix (i.e. the representation of the graph is undirected), which imply the eigenvector matrix $\mathcal{U}$ orthogonal to each other. Hence, one can rewrite the eigenvalue problem as: $L = \mathcal{U}\Lambda\mathcal{U}^T$ with eigenvalues $0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots \lambda_{n-1}$ and with $\mathcal{U}$ as a matrix of eigenvectors ($\mathcal{U}$ is also referred to as the graph Fourier basis of L and Λ is the graph frequency). Eigenvectors associated with smaller eigenvalues have values that vary less rapidly along the edges. The above decomposition also will allow to project the initial nodal value vector f onto the eigenspace $\hat{f}$ represented by the eigenvetors $\mathcal{U}$.

Solving the eigenvalue problem of the above example of the form: $det(L - \lambda I) = 0$, we obtain the corresponding eigenvalues and eigenvector matrix as:

$$\Lambda = diag(0, 0.14, 0.56, 0.92) \qquad ; U = \begin{bmatrix} -0.50 & -.56 & 0.63 & 0.23 \\ -0.50 & -0.56 & -0.63 & -0.23 \\ -0.50 & 0.43 & 0.27 & -0.69 \\ -0.50 & 0.70 & -0.27 & 0.67 \end{bmatrix},$$

where the first eigenvector associated with $\lambda_0 = 0$ is constant, meaning it corresponds to the low-frequency (smooth) mode.The higher eigenvalues correspond to higher frequency components, capturing more rapid variations.

Given the initial graph signal distribution associated with the nodal dwelling times, f, the Graph Fourier Transform $\hat{f}$ is given as:

$$\hat{f} = \mathcal{U}^T f = \begin{bmatrix} -0.50 & -0.50 & -0.50 & -0.50 \\ -0.56 & -0.56 & 0.43 & 0.70 \\ 0.63 & -0.63 & 0.27 & -0.27 \\ 0.23 & -0.23 & 0.69 & 0.67 \end{bmatrix} \begin{bmatrix} 30 \\ 15 \\ 45 \\ 5 \end{bmatrix},$$

which results in: $\hat{f} \approx (-47.5, -3.8, 2.9, -1.2)^T$, where $\hat{f}_0 = -47.5$ is the dominant low-frequency component represents the global average trend of the dwelling times. $\hat{f}_1, \hat{f}_2$ and $\hat{f}_3$ are high-frequency components capture local variations.The first coefficient of the Graph Fourier Transform, $\hat{f}_0$, is associated with the smallest eigenvalue $\lambda_0 = 0$, of the Laplacian matrix L.

Another interpretation of the above results can be to apply a filter to remove some frequency components. For example, let us apply a high-pass filter H to f in order to remove low-frequency components:

$$\hat{f}_{hp} = H\hat{f} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -47.5 \\ -3.8 \\ 2.9 \\ -1.2 \end{bmatrix} = \begin{bmatrix} 0 \\ -3.8 \\ 2.9 \\ -1.2 \end{bmatrix},$$

where by applying the Inverse Graph Fourier Transform of the form $f_{ana} = \mathcal{U}\hat{f}_{hp}$, we are able to reconstruct the corresponding nodal values as:

$$f_{ana} = \begin{bmatrix} -0.50 & -0.50 & -0.50 & -0.50 \\ -0.56 & -0.56 & 0.43 & 0.70 \\ 0.63 & -0.63 & 0.27 & -0.27 \\ 0.23 & -0.23 & 0.69 & 0.67 \end{bmatrix} \begin{bmatrix} 0 \\ -3.8 \\ 2.9 \\ -1.2 \end{bmatrix} = \begin{bmatrix} 1.6 \\ -1.4 \\ -2.3 \\ 2.1 \end{bmatrix},$$

where f_{ana} can be interpreted as the high-frequency variations, which indicate irregular behaviours (anomalies). For example, if node n_4 in the graph of activity indicate the bathroom (K), it has an unexpected increase from the smooth pattern and suggesting an unusual pattern. Or, the bedroom (B) is is represented by node n_3 shows a decrease, potentially indicating a missing event.

5 Conclusions

The advancement and widespread accessibility of pervasive sensing technologies have increased their applicability across diverse monitoring environments. These technological progress have particularly facilitated the tracking and monitoring of movements, a critical early stage in anomaly detection systems. This application domain has gained significant attention in the literature due to its potential for addressing challenges in health, environmental, and behavioural monitoring.

One main application of these sensing modalities lies in the detection of movement presence or absence within the sensors' monitoring range. This basic functionality support the design of more complex systems aimed at comprehensively analysing movement behaviours over time. The representation of these time-sequenced movement data in the form of graph structures has been proposed as an efficient and versatile approach. Graph-based modelling not only supports the visualization of movement patterns but also enables deeper analytical insights into their underlying dynamics.

In this paper, we presented three inter-related methodologies for the analysis and further exploration of graph representations of movement activities: *Markov Chains*- This probabilistic approach captures the transitional relationships between states in movement sequences, offering a statistical perspective on activity patterns; *PageRank*- leveraging principles from network theory, PageRank identifies key nodes within the movement graph, identifying the prominence or influence of particular activities or locations; and *Graph Signal Processing*- which examines movement graphs as signals over nodes, facilitating the extraction of spatial and temporal movement features through spectral analysis.

These methodologies provide complementary insights into the characterization and study of movement graphs, each can be tailored to specific objectives and requirements of monitoring systems. Whether the focus is on anomaly detection in health monitoring or on optimizing environmental tracking mechanisms, these approaches can be adapted and integrated to enhance the overall system's performance. By leveraging these graph-based analytical techniques, researchers and practitioners can achieve a more characterized understanding of movement dynamics, paving the way for advancements in predictive modelling, real-time monitoring, and anomaly detection.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

References

[1] Diestel, R. (2017), *" Graph theory"*, Springer.

[2] Pistilli, F. and Averta,G. (2023), *"Graph learning in robotics: a survey,"*, IEEE Access, vol. 11, pp. 112664-112681, doi: 10.1109/ACCESS.2023.3323220.

[3] Wan, W., Zhu, Y., Shah, R. and Zhu, Y. (2024), *"LOTUS: continual imitation learning for robot manipulation through unsupervised skill discovery"*,Proceedings of International Conference on Robotics and Automation, https://doi.org/10.48550/arXiv.2311.02058

[4] Keshmiri, S. and Payandeh, S. (2013), *" Multi-robot, dynamic task allocation: a case study"*,Journal of Intelligent Service Robotics, , Vol. 6, pp. 137–154 , https://doi.org/10.1007/s11370-013-0130-x

[5] Ghosh, S., Mallick, A., Chowdhury, A. et al. (2024), *"Graph theory applications for advanced geospatial modelling and decision-making"*, Applied Geomatics, Vol. 16, pp. 799–812, https://doi.org/10.1007/s12518-024-00586-3

[6] Rasoulidanesh, M. S. and Payandeh S. (2020), *"A novel change-detection scheduler for a network of depth sensors"*, Journal of Visual Communication and Image Representation, Vol. 66, https://doi.org/10.1016/j.jvcir.2019.102733.

[7] Brun, L.,Foggia, P. and Vento, M. (2020), *" Trends in graph-based representations for Pattern Recognition"*,, Pattern Recognition Letters, Vol. 134, pp. 3-9, https://doi.org/10.1016/j.patrec.2018.03.016.

[8] Boykov, Y. and Jolly, M.-P. (2001), *"Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 11, pp. 1285–1298,

[9] Shi, J., and Malik, J. (2000), *"Normalized cuts and image segmentation"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, 888–905

[10] Yan, S., Xiong, Y. and Lin, D. (2018), *"Spatial temporal graph convolutional networks for skeleton-based action recognition"*, AAAI Conference on Artificial Intelligence.

[11] Mohsin, N. and Payandeh, S. (2022), *"Clustering and identification of key body extremities through topological analysis of multi-sensors 3D data"*, Journal of Visual Computer, Vol. 38, pp.1097–1120, https://doi.org/10.1007/s00371-021-02070-0

[12] Choi, W., and Savarese, S. (2012), *" A unified approach for multi-target tracking and collective activity recognition"*,CVPR.

[13] Lee, J.G., Kim, K.H., and Lee, S.W. (2007), *"Trajectory clustering: A partition-and-group framework"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No.4, pp. 918–935

[14] Helbing, D., and Molnár, P. (1995), *"Social force model for pedestrian dynamics"*, Physical Review E, Vol. 51, Non. 5,pp. 4282–4286,

[15] Stroock, D. W. (2014), *"An introduction to Markov Processes"*, Springer

[16] Privault, N. (2013), *"Understanding Markov Chains: examples and applications"*, Springer

[17] Payandeh, S. (2022), *" An overview of Markov Chains for monitoring indoor movements and ambient motion detection"*. Journal of Gerontechnology, Vol. 21, No. 1, pp. 1-11, https://doi.org/10.4017/gt.2022.21.1.590.07

[18] Brin, S. and Page, L. (1998), *"The anatomy of a large-scale hypertextual Web search engine"*, Computer Networks and ISDN Systems, Vol. 30, Issues 1–7, pp. 107-117, https://doi.org/10.1016/S0169-7552(98)00110-X.

[19] Gleich D. (2015), *"PageRank beyond the web"*, SIAM Review. Vol. 57, No. 3, pp. 321-363, https://doi.org/10.1137/140976649

[20] Payandeh, S. and Chiu, E. (2019), *"Application of modified PageRank Algorithm for anomaly detection in movements of older adults"*, International Journal of Telemedicine and Applications, Vol. 2019,Article ID 8612021, 9 pages, https://doi.org/10.1155/2019/8612021

[21] W. Xing, W. and A. Ghorbani, A. (2004),*"Weighted PageRank algorithm"*,Proceedings. Second Annual Conference on Communication Networks and Services Research, pp. 305-314, doi: 10.1109/DNSR.2004.1344743.

[22] G. Leus, A. G. Marques, J. M. F. Moura, A. Ortega and D. I. Shuman (2023), "Graph signal processing: history, development, impact, and outlook," in IEEE Signal Processing Magazine, vol. 40, no. 4, pp. 49-60, doi: 10.1109/MSP.2023.3262906.

[23] Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A. and Vandergheynst,P., *"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, May 2013, doi: 10.1109/MSP.2012.2235192.

[24] Ortega, A., Frossard, P., Kovačević, J., Moura, J.M.F and and Vandergheynst, P. (2018), *"Graph signal processing: overview, challenges, and applications,"*, Proceedings of the IEEE, vol. 106, no. 5, pp. 808-828, doi: 10.1109/JPROC.2018.2820126.

[25] Isufi, E., Gama, F., Shuman, D. I. and Segarra, S. (2024), *"Graph filters for signal processing and machine learning on graphs,"*, IEEE Transactions on Signal Processing, Vol. 72, pp. 4745-4781, doi: 10.1109/TSP.2024.3349788.

[26] Zhou, D. and Schölkopf, B. (2004), *"A regularization framework for learning from graph data"*, Proceedings of ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields, pp. 132-137, https://hdl.handle.net/11858/00-001M-0000-0013-F3AB-E

[27] Dang, Q., Wang, Y. and Wang, F. (2018), *" Accelerated matrix inversion approximation-based graph signal reconstruction"*, In: Liu, X., Cheng, D., Jinfeng, L. (eds) Communications and Networking. ChinaCom 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 262. Springer, Cham. https://doi.org/10.1007/978-3-030-06161-6_61