applied sciences

MDPI

Article

Product Space Clustering with Graph Learning for Diversifying Industrial Production

Kévin Cortial *, Adélaïde Albouy-Kissi and Frédéric Chausse

Université Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal,
F-63000 Clermont-Ferrand, France

* Correspondence: kevin.cortial@doctorant.uca.fr

Abstract: During economic crises, diversifying industrial production emerges as a critical strategy to address societal challenges. The Product Space, a graph representing industrial knowledge proximity, acts as a valuable tool for recommending diversified product offerings. These recommendations rely on the edges of the graph to identify suitable products. They can be improved by grouping similar products together, which results in more precise suggestions. Unlike the topology, the textual data in nodes of the Product Space graph are typically unutilized in graph clustering methods. In this context, we propose a novel approach for economic graph learning that incorporates learning node data alongside network topology. By applying this method to the Product Space dataset, we demonstrate how recommendations have been improved by presenting real-life applications. Our research employing a graph neural network demonstrates superior performance compared to methods like Louvain and I-Louvain. Our contribution introduces a node data-based deep graph clustering graph neural network that significantly advances the macroeconomic literature and addresses the imperative of diversifying industrial production. We discuss both the advantages and limitations of deep graph learning models in economics, laying the groundwork for future research.

Keywords: graph neural networks; community detection; product space

check for updates

Citation: Cortial, K.; Albouy-Kissi, A.; Chausse, F. Product Space Clustering with Graph Learning for Diversifying Industrial Production. *Appl. Sci.* **2024**, *14*, 2833. https://doi.org/10.3390/app14072833

Academic Editors: Barbara Strug and Grażyna Ślusarczyk

Received: 19 February 2024
Revised: 16 March 2024
Accepted: 25 March 2024
Published: 27 March 2024

1. Introduction

Since 2020, the outbreak of the COVID-19 pandemic and the economic consequences of the Russo–Ukrainian armed conflict have posed significant challenges to global supply chains, leading to severe disruptions and stock-outs in essential goods and services. Such disruptions can have far-reaching consequences, impacting not only socio-economic stability but also manufacturing. One of the most promising approaches is for industries to diversify their activities [1]. It has emerged as a viable strategy for mitigating the risks associated with shortages and stock-outs in global supply chains. By reducing reliance on a single source or region, industries can establish alternative supply networks, thereby enhancing resilience and mitigating the impact of disruptions. Diversification involves identifying potential suppliers in different geographically close locations, fostering partnerships with multiple vendors, and strategically managing inventory to ensure a steady local flow of essential goods and services. This approach not only reduces the vulnerability of industries to crises but also promotes local innovation, as different suppliers bring their unique capabilities and expertise to the market [2]. Moreover, diversification can foster economic growth by enabling the development of new industries and employment opportunities, while simultaneously reducing the dependence on a limited set of resources [3].

Several tools are available to support industrial diversification. One of these tools is market research, which entails the examination of market trends and the identification of prospective new suppliers. Another tool is technology assessment, which involves evaluating technological advancements and their potential applications in various industries. By staying updated on technological innovations, industries can identify opportunities to incorporate new technologies into their operations, leading to improved efficiency, new product development, and increased competitiveness [4]. Additionally, collaboration and partnerships with other industries or research institutions can facilitate diversification by leveraging shared resources, knowledge, and expertise. Such local collaborations can foster innovation, enhance capabilities, and open doors to new market opportunities [5]. Overall, these tools empower industries to explore new horizons, diversify their activities, and adapt to crises and evolving market dynamics.

Among the tools available to help industries diversify their product offerings, Product Space, developed by economists Hidalgo and Hausmann [6–8], serves as an open-access framework to identify diversification opportunities for industries, particularly during stock-outs and shortages. It functions as a map representing different product types using a global nomenclature of a harmonized system [9], showcasing relationships between products based on manufacturing methods. By exploring adjacent products within the Product Space graph, policymakers and industry stakeholders can pinpoint potential areas for expansion and diversification. This data-driven approach supports informed decision-making, allowing industries to strategically diversify activities and mitigate risks associated with supply chain disruptions [10]. The Product Space thus serves as a recommendation system, guiding manufacturers toward product diversification by highlighting related products with untapped potential [11]. This approach makes it possible to respond to certain shortages or out-of-stock situations. For example, during the COVID-19 crisis, many countries lacked hydroalcoholic gels. The spirits industry produced these gels because their production equipment was similar to that used to produce spirits or medical gel. In Product Space, these two types of products are linked in Figure 1. Consequently, using the Product Space as a recommendation system enables industries to strategically diversify their product offerings and adapt to shortages.

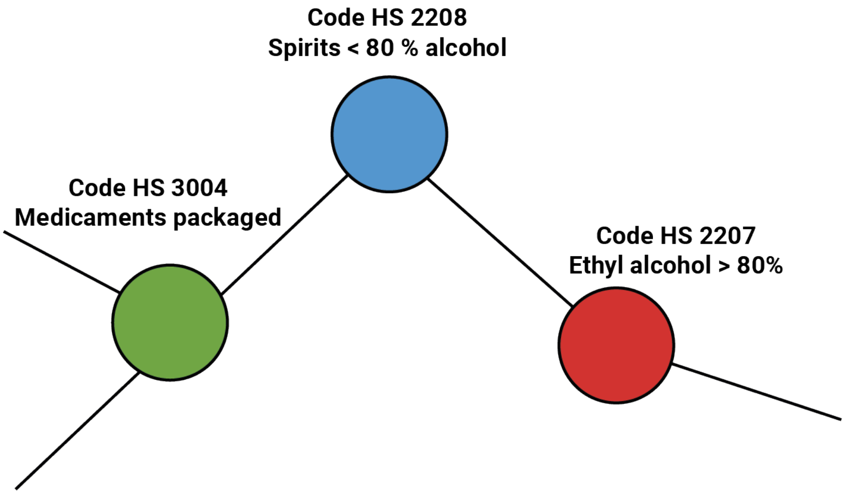


Figure 1. Example of tree nodes in Product Space.

The Product Space establishes connections between the products of the global economy, which can be considered to be nodes linked by edges representing the productive proximities between products. Thus basically, this structure forms a graph in the mathematical context, facilitating the representation of interactions among entities. It operates on a simple formalism capable of modeling complex systems like the Product Space. As a result, graphs serve as versatile tools for representing various facets of reality across numerous domains. The Product Space, composed of interconnected objects, exemplifies such a structure where specific pairs of objects demonstrate relationships. Graph theory is a mathematical method used to model and analyze pairwise relations between objects.

In this study, we propose improving these recommendation systems by clustering the Product Space graph. Clustering is a technique used to group similar elements together. In the case of Product Space, we are grouping products together. Clustering techniques have proven to be effective in enhancing recommendation systems by organizing data into meaningful groups [12]. By applying clustering algorithms to graphs, recommendation systems can identify similar products/nodes and group them together. This enables the system to provide more accurate and relevant recommendations to users based on the product link and the clustering results. Indeed, clustering information can be leveraged to generate recommendations by the product/node belonging to a group, i.e., a cluster, and the product links. By improving the precision and relevance of recommendations, clustering-based approaches improve the overall performance of recommendation systems.

Graph community detection methods have made significant contributions to the analysis and understanding of complex networks. The scientific literature on statistical community detection methods in graphs, such as clustering [13] and the Louvain algorithm [14], is extensive and rich in insights. However, Product Space is a graph that has many textual, numerical, and categorical data in its nodes to characterize each product. These statistical methods for graph clustering do not use node and edge data, but only graph topology. To improve the Product Space-based recommendation system, we need to use the textual data present in the nodes. Deep learning methods are used to perform machine learning on this data, considering the graph structure: these are graph neural network methods [15].

Existing recommendation systems often lack precision and relevance due to their reliance solely on product linkage and generic clustering techniques. While clustering methods have shown promise in enhancing recommendation accuracy, traditional statistical approaches overlook the rich data within product nodes. Moreover, the dynamic nature of the industrial ecosystem necessitates agile production strategies, underscoring the need for recommendation systems that can adapt to evolving market conditions. To address these challenges, there is a need for a comprehensive framework that integrates deep learning techniques to analyze textual data within Product Space and generate more precise recommendations. By bridging the gap between traditional statistical methods and cutting-edge deep learning approaches, this study aims to advance recommendation systems for manufacturers operating within dynamic market environments. In summary, the key contributions of this paper can be outlined as follows:

- We develop a comprehensive clustering graph framework that utilizes node features to enhance a recommendation system for manufacturers, enabling them to diversify their production in response to shortages and stock-outs.
- We develop deep graph learning approaches for community detection approaches on a macroeconomic graph.
- We propose an economic graph analysis by learning nodes and edge features in addition to graph topology.
- We implement the suggested framework on an actual macroeconomic dataset alongside a state-of-the-art graph, and we confirm through validation that our deep graph learning outperforms statistical tools in community detection.

The distinctive scientific contribution of the article lies in its advancement and utilization of deep graph learning techniques within the field of economics. More specifically, the paper presents a new approach to economic graph learning and analysis characterized by learning node data in addition to network topology.

In this article, we will present the literature in Section 2. After a quick approach to the work carried out on the Product Space, we will then insist on the methods of graph learning which make it possible to analyze these relational data in the economy. Section 3 presents the methods that we applied to the Product Spaces and state-of-the-art dataset. Section 4 will detail important steps of our experimentation, such as data preprocessing and evaluation metrics. Section 5 will present the evaluation of results that confirm that a Deep Graph Clustering method provides more meaningful clusters than traditional methods that do not exploit graph attributes. Finally, we will conclude by delving into future research possibilities regarding this topic in Section 6.

2. Related Work

Since Hidalgo's works [6–8], several studies have been conducted on the Product Space to analyze a wide range of economic phenomena. For example, Hausmann [7] used Product Space to study the relationship between trade and innovation, and found that countries with more complex Product Space are more likely to generate new technologies. Complexity in Harvard's Product Space is a measure of the economic and technological sophistication of the products an industry can produce. Product Space identifies environmentally friendly products with the greatest growth potential within a country by assessing their proximity to products that the country produces with a high relative comparative advantage [16]. As well as analyzing the economy theoretically, Product Space can also be used empirically, i.e., through experimentation and observation [17]. In this field, Pachot et al. [11] used the Product Space to respond to shortages caused by the COVID-19 crisis. Product Space highlighted the adaptability of certain companies that quickly adapted their production chains by producing goods experiencing shortages due to the similarity in expertise between the two categories of products [18]. This work shows that Product Space can be used as a recommendation system for industries to diversify their product offerings. Product Space can be described as a graph comprising a collection of interconnected nodes. In this context nodes represent products and edges represent links between products, indicating their similarity and complementarity of Hidalgo. To improve these graph-based recommendations, clustering has proven its effectiveness in obtaining more relevant results [19–21]. On the other hand, studies have shown that GNNs can bring diversity to recommendations [22,23]. Therefore, in our work, we implement GNNs that produce clusters grouping similar and relevant elements. Building upon these results, we enhance industrial diversification recommendations through meaningful diversification. Our literature review was therefore divided into several parts. First, a literature survey was carried out to provide an overview of the main graph learning methods for node clustering in Section 2.1. In parallel, a review of economic articles utilizing graphs was conducted to demonstrate the contributions of this article, as detailed in Section 2.2.

2.1. Graphs Clustering

Identifying communities within graphs is a core issue that has applications in various domains, notably economics, which we will detail in Section 2.2. In this section, we offer a summary of current methodologies, algorithms, and techniques used for community detection in mathematical graphs. It explores the key concepts, focusing on both statistical and deep learning approaches.

Statistical community detection in graphs aims at detecting clusters of closely interconnected nodes, known as communities or clusters. Modularity-based methods [13] have gained significant popularity in the field of community detection due to their ability to uncover cohesive and well-separated communities within a graph. Assessing the quality of community structure in a graph often involves using modularity as a commonly adopted metric. It measures the discrepancy between the observed edges within communities and the anticipated edges in a random graph with equivalent node degrees [24]. Maximizing the modularity value indicates the presence of communities defined by the graph topology. The Louvain method, proposed by Blondel et al. [14], is one of the most popular modularity-based algorithms for community detection. It is a hierarchical and iterative approach that optimizes modularity in a two-step process. In the first step, nodes are moved between communities to maximize the increase in modularity. During the subsequent phase, the communities identified in the initial step are regarded as singular entities, and this procedure iterates until there is no additional enhancement in modularity [14]. Based on the same modular design, the stochastic block model (SBM) serves as a robust probabilistic generative model utilized for examining the community structure within networks. It assumes that nodes in a network can be partitioned into communities, and the connectivity patterns within and between communities follow certain probabilistic rules. SBM provides a framework for studying the fundamental properties of network communities, enabling a deeper understanding of complex network structures and dynamics [25].

Modularity methods [13] have been extensively used for community detection in networks due to their simplicity and interpretability. One limitation of modularity methods is their resolution limit, where they struggle to detect smaller or overlapping communities. Graph Neural Networks (GNNs), on the other hand, can capture more complex structural patterns and capture node and edge features, allowing for the detection of fine-grained and overlapping communities. GNNs offer parallel and scalable operations, making them suitable for handling massive networks efficiently. Additionally, GNNs can leverage both the network structure and node attributes to improve community detection accuracy. The main limitation of modularity-based methods is that they only use graph topology and not node data.

In recent years, many publications have demonstrated the positive value and good results of new machine and deep graph learning methods, popularized by Kipf and Welling [26] for deep learning. To exploit graphs and node data, GNNs, rely on the assumption that many pieces of information of a node reside in its neighborhood. Indeed, nodes and edges data, in the form of embedding (numerical vector), can be inferred by neural networks. However, it is crucial to acknowledge a broader perspective on the mathematical modeling of node data. These variables can manifest not only as scalars but also as vectors, matrixes, and even functions [27]. In our case, given that node data are of finite texts, we will utilize scalars, as elaborated in Section 4.1.

GNNs transmit messages between pairs of nodes to update their embedding thanks to the exchange of information with their neighbors. In this way, GNNs provide better representations of nodes within their environment. From these new embeddings, it is possible to predict links between nodes, to form clusters of nodes (clustering), or to perform classification. Several different GNN architectures have been proposed. Like pixels in an image, structured graphs are grids of nodes, so Graph Convolutional Networks (GCNs) [26] are GNNs applied to grids of nodes like Convolutional Neural Networks (CNNs). Based on this method, Dynamic Graph Convolution Neural Network (DGCNN) has proved efficient in the segmentation of coal mining data with the aim of reducing its environmental footprint [28]. Other GNN architecture is proving effective, such as parsimonious neighbor selection in GraphSAGE [29] or adding the attention mechanism [30] in Graph Attention Networks [31]. These examples implement different types of messages passing between nodes.

While general methods for graph clustering have demonstrated their efficiency in various domains, the application of graph learning techniques, particularly in the field of economics, holds significant potential for uncovering intricate patterns and insights from complex economic systems.

2.2. *Economics Graph Learning*

Economists are mainly interested in understanding economic phenomena using network concepts. The attention toward graph clustering in economic research has increased significantly owing to its capability to capture the intrinsic structure and interconnections present within economic networks. The clustering of economic agents, such as firms, industries, or regions, allows for a deeper understanding of their interactions and the emergence of complex economic phenomena [32].

Recently, there has been an emergence of several graph clustering methodologies, predominantly rooted in classical statistical techniques that solely rely on the topology of the graphs. Many statistical indicators are used in economics such as centrality, clustering tendency, or modularity optimization [33,34]. Modularity optimization aims to increase the density of connections within clusters while decreasing the connections between clusters. Moreover, when nodes are economic agents, their assignments to a community allow predicting economic phenomena related to potential business ecosystems [32]. These economic studies have also shown that graph community detection provides a better cluster than classical clustering on non-relational economic data. When networks are large, community detection on economic graphs use frequently Louvain [35,36], Stochastic Block Model [37] or Leiden algorithms [32]. All these methods obtain good clusters using only graph topography but without exploiting nodes and edges features.

Linkage prediction methods are applied in economics to predict the evolution of future economic networks to guide policymakers. Authors use these methods to predict possible linkages in the future labor market [38]. However, these methods use adjacency matrix perturbations, for example, but without capitalizing on the data within the graphs [39]. The use of data within graphs (embedding) remains rare in economics. For example, Mungo et al. [40] used node data to reconstruct supply chain networks by performing a classification of possible connection pairs with Gradient Boosting. Additionally, Wu et al. [41] used node data in a classification task, although in a different context from our work which is clustering.

In Table 1, we review recent studies exploring graph clustering methods for applications in economics. Our analysis indicates that most of the selected studies concentrate solely on analyzing the graph topology, without any study focusing on the features of nodes and edges. Consequently, these studies restrict the generalizability of the proposed methods to leverage all available information. Moreover, it is noticeable that these methods do not capture high-level dependencies. For example, using attributes of nodes representing economic agents, it becomes possible to group entities that are similar in terms of economic behavior, even if they are not directly connected in the graph's topology. Hence, in our research, we propose a more versatile approach that incorporates node and edge features to capture higher-level dependencies among graph entities. In summary, our approach enables us to take fuller account of the information available, better capture high-level dependencies, improve the accuracy and interpretability of the results, and provide greater flexibility and adaptability to specific economic contexts. In addition, the potential application of advanced GNNs in economic graph analysis could explore aspects such as market dynamics, risk assessment, and supply chain optimization.

This literature review shows that the key scientific novelty in our research emerges from our application of deep graph learning techniques within the realm of economics. To be more precise, our study introduces an innovative method for understanding and utilizing economic graphs, which goes beyond conventional analyses by encompassing the learning of node-specific data in addition to network structure.

Table 1. Comparative analysis of our proposed framework, highlight in bold, with recent pertinent graph clustering methods in economics.

Economic Graph Study	**Use Nodes/ Edges Data**	**Year**	**Micro/ Macroeconomics**	**Graph Clustering Methods**	**Application**
Tajoli [33]	No, but it could have been possible	2019	Macroeconomics	Modularity, clustering coefficient	World trade
Korniyenko et al. [34]	No, but it could have been possible	2017	Macroeconomics	Modularity, clustering coefficient	World supply shock
De Nicolò et al. [32]	Use edges data but not nodes data	2022	Macroeconomics	Leiden, Spin glass	Territorial Development
Chessa et al. [35]	No, Hypergraph	2023	Macroeconomics	Louvain	Trade agreements
Zhang et al. [36]	No	2021	Microeconomics	Louvain	Customer product affinity
Kafkas et al. [37]	No	2021	Microeconomics	Stochastic Block Modeling (SBM)	Product segmentation; Market basket analysis
Benita et al. [42]	No	2020	Macroeconomics	Louvain	Nation industrial clusters
This research	**Yes**	2024	Macroeconomics	**GraphSAGE.**	Diversifying industrial production

3. Graph Clustering Methods

This section explains the graph clustering benchmark algorithms. The popular Louvain method is introduced [14], centered around optimizing the modularity score. Afterward, the extension, I-Louvain [43], is presented, which combines modularity with the consideration of the statistical proximity of feature nodes. Lastly, the GraphSAGE method [29], a graph neural network utilized for community detection, is discussed in detail.

3.1. Louvain and Modularity

Modularity serves as an assessment of how effectively the nodes within a graph are divided into distinct partitions. This concept suggests a prevalence of connections within each partition, known as intra-community edges, contrasted with a lower occurrence of connections between different partitions, termed inter-community edges. Essentially, it indicates that nodes within the same community are more strongly connected to each other compared to nodes in different communities [13]. The modularity score evaluates, for a set of nodes, the ratio of observed edges to expected edges (based on a comparable graph with edges distributed randomly, akin to the Erdös-Rényi model [24]). If the observed edges exceed the expected count, it indicates the likelihood of a community structure. This score quantifies the partitioning quality of a graph using the following formula:

$$Q_{\text{modularity}} = \frac{1}{2m}\sum_{i,j}(A_{ij} - P_{ij})\delta(C_i, C_j) \tag{1}$$

Here, A_{ij} represents the value at position ij in the adjacency matrix. m denotes the total number of edges, while $2m$ signifies the total count of half-edges. The symbol δ refers to the Kronecker delta: $\delta(C_i, C_j) = 1$ if nodes i and j belong to the same community C (i.e., $C_i = C_j$), and $\delta(C_i, C_j) = 0$ otherwise. P_{ij} represents the probability of the number of connections between nodes i and j under a null Erdös-Rényi model, which generates a uniformly connected graph. Consequently, maximizing the modularity Q involves identifying sets of nodes exhibiting an unusually high level of connectedness.

Modularity serves as the foundational principle for extracting communities within graphs, and the Louvain method [14] stands out for its efficacy, especially when dealing with large datasets.

This method operates hierarchically, where in the initial phase, it identifies small communities through local optimization of modularity for each node. Subsequently, nodes within the same community are amalgamated into a single node. This process is iteratively repeated on the updated network until no further increase in modularity is achievable (Figure 2 [14]).

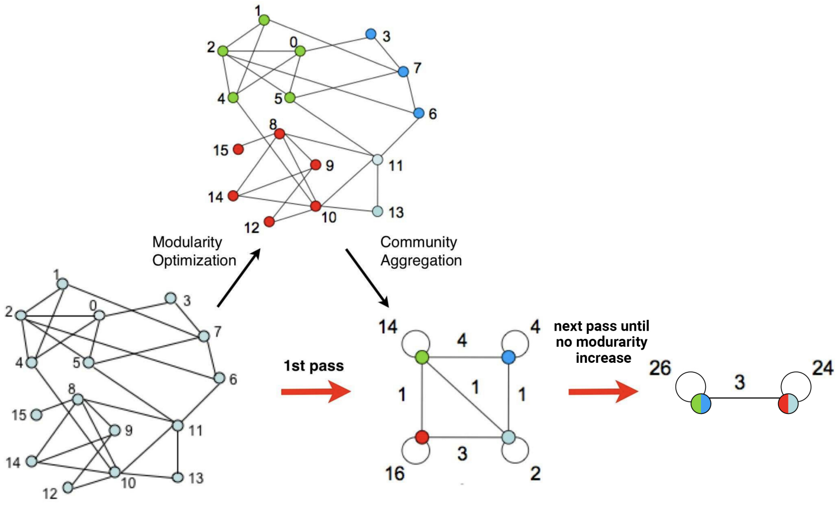


Figure 2. Visualization of the steps of Louvain algorithm.

Unlike the Louvain algorithm, which merges communities at each level, the Leiden algorithm [44] primarily focuses on splitting and merging clusters at each level. As a result, it ensures the formation of more well-connected clusters. In comparison to the Louvain algorithm, the Leiden algorithm incorporates a fast local move approach, enabling the movement of one or more nodes from one cluster to another to enhance the quality of clusters during each iteration of community detection. Nodes are selected for movement only if they are considered unstable. This distinction improves the runtime efficiency of the Leiden algorithm compared to the Louvain method. Additionally, the Leiden algorithm addresses a major inefficiency of the Louvain method, which occasionally generates poorly connected nodes as a community and may result in a fragmented network community.

3.2. I-Louvain

I-Louvain [43] is a technique for identifying groups within a graph, where each node has numerical attributes. It improves upon the modularity measure [13] and includes an additional measure for further optimization. I-Louvain therefore measures the inertia between the data of two nodes to attempt to group together the most similar elements in the embeddings, which are the numerical vectors of each node. This measure of inertia-based modularity is defined by Combe et al. [43] as follows:

$$Q_{\text{inertia}} = \sum_{(v,v')V.V} \left(\frac{I(V,v) \cdot I(V,v')}{(2N.I(V))^2} - \frac{\|v - v\prime\|^2}{(2N.I(V))} \right) \delta(C_v, C_{v\prime}) \tag{2}$$

Let V denote a set comprising N elements, represented within a real vector space, where each element $v \in V$ is characterized by a vector of attributes $v = (v_1, \ldots, v_n) \in \mathbb{R}$. The inertia of V about its center of gravity, denoted as $I(V)$, and the inertia of V about a specific element v, denoted as $I(V,v)$, are defined as the sum of the squared Euclidean distances between v and the other elements of V. Similar to the modular quality metric $Q_{\text{modularity}}$, δ represents the Kronecker delta.

Q_{inertia} evaluates the discrepancy between the anticipated and observed distances between pairs of elements (v, v') within the same community. If the observed distance is less than the expected distance, it suggests that v and v' are potential candidates for belonging to the same cluster.

I-Louvain is a community detection technique tailored for real attributed graphs, leveraging inertia-based modularity Q_{inertia} in conjunction with Newman's Q_{inertia}. Essentially, this method revolves around optimizing the global criterion $Q_{I-\text{Louvain}}$, defined as follows:

$$Q_{I-\text{ Louvain}} = Q_{\text{modularitv}} + Q_{\text{inertia}} \tag{3}$$

As with the Louvain method, I-Louvain works by evaluating the gain in $Q_{I-\text{ Louvain}}$ those results from moving each node v and its adjacent nodes in the graph to different communities. During each iteration, node v is assigned to the community that yields the maximum gain in the global criterion $Q_{I-\text{ Louvain}}$. This process is repeated sequentially for all nodes until no further improvement in $Q_{I-\text{ Louvain}}$ can be achieved.

3.3. Graph Neural Networks for Community Detection

Graph Neural Networks (GNNs) represent a category of deep learning algorithms designed to extract features from graph data. Traditional deep learning methods are typically tailored for structured data. However, graphs can vary significantly in size, exhibit multimodal features, and possess intricate topologies. By leveraging node-level information, GNNs can directly facilitate predictions, classifications, or other analytical tasks.

In leveraging graph data, GNNs operate under the assumption that significant amounts of node information are contained within their neighborhoods. In this way, the topology of the graph can be encoded in node embeddings. These node embeddings contain the initial data relating to the node itself, and neighborhood data thanks to the updating of the node embeddings carried out by the GNNs. Thus, a GNN uses a neural network on the neighbors of each node; this is referred to as a GNN layer.

A single GNN layer compresses a set of embeddings into a single embedding in two steps: the message passing and the aggregation. Equation (4) presents a single layer of Graph Convolutional Network (GCN) [26] where the invariant aggregation function is a sum, and the message passing is a linear matrix operation with a weight matrix U.

Formula (4) delineates the process by which input information, derived from both the target node h_i^t and its neighboring nodes h_j^t, undergoes aggregation by neural networks to generate the updated representation h_i^{t+1}. This mechanism is depicted in the gray and black blocks in Figure 3 [45].

$$h_i^{t+1} = \sigma\left(h_i^t W + \sum_{jN(i)} \frac{1}{c_{ij}} h_j^t U \right) \tag{4}$$

- h_i^t represents the initial vector comprising the node features.
- h_i^{t+1} denotes the updated vector encompassing information regarding the node's neighbors and the graph's topology. This augmentation furnishes richer insights and enhances the representation of the node h_i.
- W represents the weight matrix from a neural network, discerning significant elements to retain within the initial vector h_i^t.
- U is another weight matrix originating from a neural network, tasked with processing the vectors of neighboring nodes h_j^t.
- The expression $\left(\sum_{j\in N(i)} \frac{1}{C_{ij}}\right)$ represents an aggregation function, which normalizes the representations of neighboring nodes in $N(i)$. This necessitates a permutation-invariant function, as we must be agnostic to the order of neighbors to obtain consistent results regardless of their arrangement.
- σ denotes an activation function similar to sigmoid.

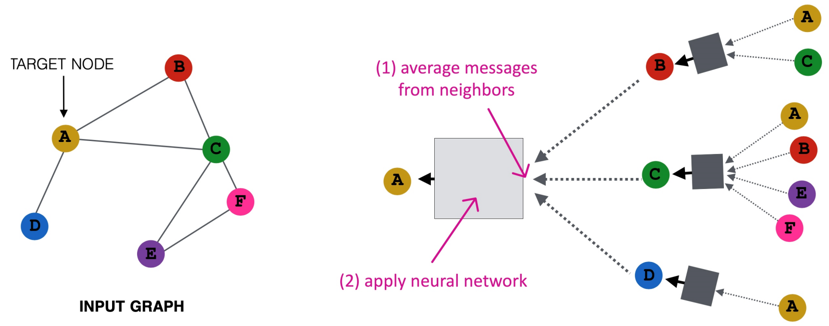


Figure 3. Overview of graph encoding with the GNNs neighborhood aggregation methods.

To determine U and W, supervised learning of a neural network involves adjusting the weights of the network to minimize the difference between the predictions and the expected outputs. This iterative process allows the network to gradually improve its ability to make accurate predictions on new data. However, this supervised learning requires data labeling.

Graph clustering performs unsupervised learning without label data. Indeed, our goal is to do clustering with the new embeddings updated by GNNs to detect communities in the graph. So, GraphSAGE will be used to generate a labeled training sample that will be used to update the node's embeddings.

3.4. Graphsage for Community Detection

GraphSAGE (Graph Sample and Aggregated) [29] is a variant of a Graph Neuronal Network (GNN) in the family of deep learning methods for graphs. GraphSAGE leverages the concept of graph convolution [26], which enables the aggregation of information from adjacent nodes to update the representation of a target node like GCNs. However, in GraphSAGE neighbor sampling to handle large graphs efficiently, GraphSAGE adopts a sampling strategy where it randomly selects a consistent number of neighbors for every node. This helps reduce computational overhead while preserving the overall graph structure.

For a given central node i with current embedding h_i^t the message passing the update rule to transform it into h_i^{t+1} is as follows:

$$h_i^{t+1} = W_{dst} \cdot h_i^t + W_{src} \cdot AGG\left\{h_j^t, \forall jN(i)\right\} \tag{5}$$

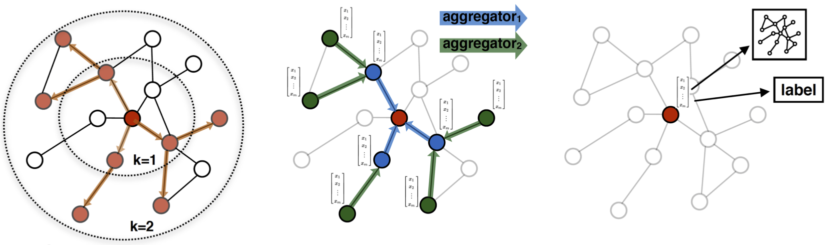
The aggregated representations are then combined with the current node embeddings to create a new embedding that captures both local and global graph information. The new embeddings are computed with W_{dst} and W_{src} learned by neural networks.

After sampling the neighbors, the information from the selected neighbors is aggregated to create a summary representation for each node. This aggregation step is typically performed using an aggregation function such as mean for example.

$$AGG\left\{h_j^t, \forall jN(i)\right\} = \frac{1}{N(i)} \sum_{jN(i)} \left(h_j^t\right) \tag{6}$$

The entire process of neighbor sampling, aggregation, and updating node embeddings is performed in multiple iterations, also known as training epochs. During training, GraphSAGE aims to minimize a loss function assessing the quality of the learned node embeddings.

Our problem is to perform unsupervised learning to update the embeddings. Indeed, the idea here is to update node embeddings using only the structure of the graph and the characteristics of the nodes, without using any known class labels for the nodes. The main goal is to do clustering with the new embeddings to detect communities in the graph. Unsupervised GraphSAGE is adapted to updating node embeddings through the resolution of a classification task [46]. For this, "positive" pairs of nodes are produced by conducting random walks on the graph (Figure 4 [29]). Another equal set of "negative" node pairs is randomly chosen from the graph based on a distribution linked to the average degree of connection in the graph.



1. Sample neighborhood

2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

Figure 4. Illustration and explanation of GraphSAGE.

By mastering the straightforward binary classification task of node pairs, the model naturally develops an inductive mapping that converts node attributes and their neighboring nodes into node embeddings in a high-dimensional vector space. This mapping efficiently preserves the structural and feature similarities among the nodes. Unlike embeddings generated by algorithms such as Node2Vec [47], this mapping is inductive in nature. This means that when presented with a new node (accompanied by its attributes) and its connections to other nodes in an unseen graph (absent during model training), we can readily evaluate its embeddings without the need for retraining the model.

Thus, the embeddings reflect not only node data, but also their relationships with their peers. The resulting node embeddings are then subjected to a conventional unsupervised learning algorithm to determine clusters accordingly. We present the product groups obtained using the k-means algorithm [48]. The k-means algorithm is a data partitioning technique that categorizes data logs into k clusters. Essentially, it aims to distribute the samples into n groups with uniform variances while minimizing the inertia or intra-cluster sum of squares, as defined by the following equation:

$$\sum_{i=0}^{n} \left(\min_{\mu_j \in C} \left(|x_i - \mu_j|^2 \right) \right) \tag{7}$$

The k-means algorithm partitions a set of n samples x into k distinct clusters C, with each cluster characterized by its mean μ_j, commonly known as the cluster centroids.

Primarily, the experimental setup involves determining the optimal number of clusters to retain. Various methods exist for this purpose, including analyzing the percentage of variance explained relative to the number of clusters [49]. This typically entails solving the clustering problem for different values of k and then employing suitable criteria to select the most suitable value. Notably, the proposed method directly furnishes clustering solutions for all intermediate values of k, thereby eliminating the need for additional computational efforts. The selection of the number of clusters hinges on ensuring that the addition of another cluster does not substantially improve data modeling. Specifically, as the percentage of variance explained by clusters is plotted against the number of clusters, the initial clusters contribute significantly to explaining variance. However, there comes a point where the marginal gain diminishes. Thus, the number of clusters is chosen at this juncture, where the addition of another cluster yields little improvement.

This section has presented benchmark algorithms for graph clustering such as statistical methods from Louvain to deep learning methods such as GraphSAGE. To compare the performances of these methods an experimental setup on reference datasets was implemented using a systematic and rigorous approach. These reference algorithms serve as a basis for evaluating the effectiveness and performance of clustering methods, with the aim of improving industrial diversification recommendations using the Product Space graph.

4. Experimentations

This study introduces a machine learning approach aimed at clustering Product Space nodes for the identification of communities, with the goal of enhancing recommendations for industrial diversification. To optimize the task of graph community detection, our approach relies on leveraging the information inherent in the graphs (nodes and edges) to achieve superior results compared to conventional methods that solely rely on graph topology. To validate this hypothesis, we conducted a comparative analysis of three methods, as depicted in Figure 5 and detailed in the preceding section.

The goal of the Product Space [7] is to group product codes from the Harmonized System (HS) [9] nomenclature by considering their dependency/linkage between them. To guarantee the robustness of our results across multiple datasets, we test the same methods on a state-of-the-art graph: Cora [50]. Both graphs contain textual data for each node.

- Product Space: This graph illustrates the proximity of industrial knowledge among product classes within the HS nomenclature, irrespective of the observed country or territory. It comprises 697 nodes and 5556 edges, with each node characterized by a textual description.
- Cora dataset [50] : The Cora dataset comprises 2708 scientific publications categorized into seven scientific domains. The citation network contains 5429 links. Each publication is represented by a binary word vector, indicating the presence or absence of corresponding dictionary words. The dictionary contains 1433 unique words.

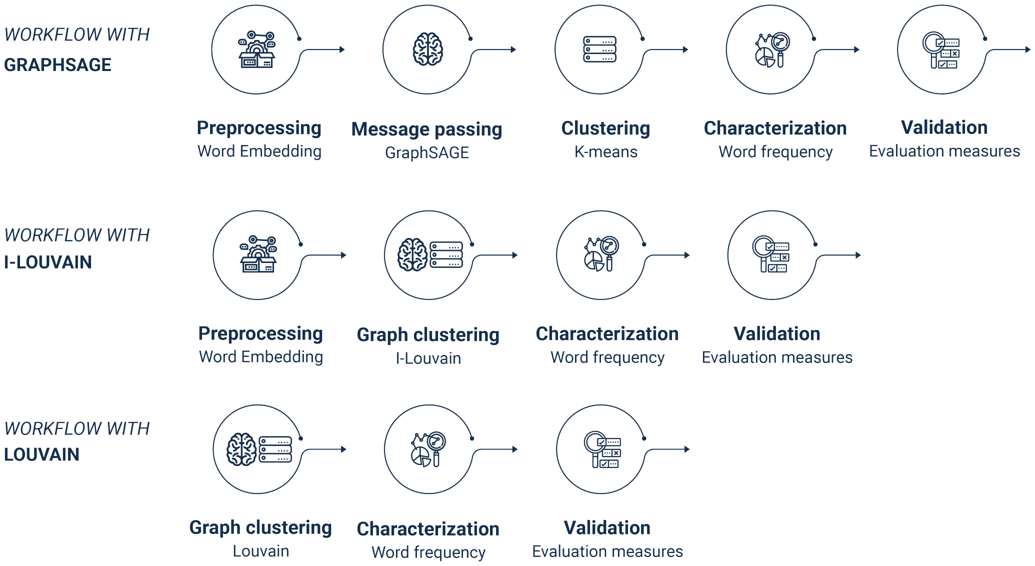


Figure 5. The workflow of the three methods compared in the benchmark.

4.1. Word Embedding Process

To process the Product Space text data, we implemented the Word2Vec technique [51] which consists of representing each word as a numerical vector (embedding) in its linguistic context. By leveraging distributed word representations, Word2Vec captures the semantic relationships between words and phrases in the product descriptions, effectively transforming them into dense numerical vectors. The learning is based on specialized neural networks. However, no labels are required for learning, as the ground truth is directly inferred from the proximity of words within the training corpus. Thus, Word2Vec is self-supervised learning. Word2Vec has already been used for calculating similarities in industrial waste nomenclatures [52], analyzing taxonomy [53], or providing recommendations [18], although the embeddings cannot be interpreted directly.

However, to achieve optimal performance and meaningful embeddings for textual descriptions, careful selection and fine-tuning of the hyperparameters are essential.

One of the key hyperparameters in Word2Vec is the dimensionality of the word embeddings. Choosing an appropriate dimensionality ensures that the embeddings capture sufficient semantic information while avoiding overfitting or excessive computational overhead. After testing, in our case, 100 embeddings size were sufficient to represent the words of each product code of the Product Space nodes. In addition, the window size hyperparameter determines the context in which words are considered to learn their representations. Adjusting this parameter allows the model to capture different levels of word associations, which can significantly impact the quality of the encoded descriptions. Product descriptions are short, so a window size of 3 is sufficient. Our word embedding has five training iterations or epochs because there is a correct balance between training time and convergence. Indeed, inadequate training may result in incomplete embeddings, while excessive training may lead to overfitting. Furthermore, Skip-Gram was chosen for its ability to capture word co-occurrence patterns effectively.

4.2. Characterization and Visualization

The clusters created using the three methods were characterized for the Product Space. Indeed, a grouping of the textual descriptions of each product has been performed for each cluster. By retrieving the most frequently mentioned words, we were able to characterize each cluster by a few words (Figure 6).

Creating an effective two-dimensional visualization of a graph is challenging due to the need to balance the representation of connections and node positions within a confined space. To perform this task, we used the Fruchterman and Reingold force-directed placement process [54] among the collection of edges and nodes within the Product Space

(Figure 6). The algorithm uses a force-directed approach for network representation. It treats edges as springs, which act to maintain proximity between nodes. Simultaneously, it regards nodes as entities that repel each other, similar to an anti-gravity force. This simulation continues until the positions of nodes reach an equilibrium state.

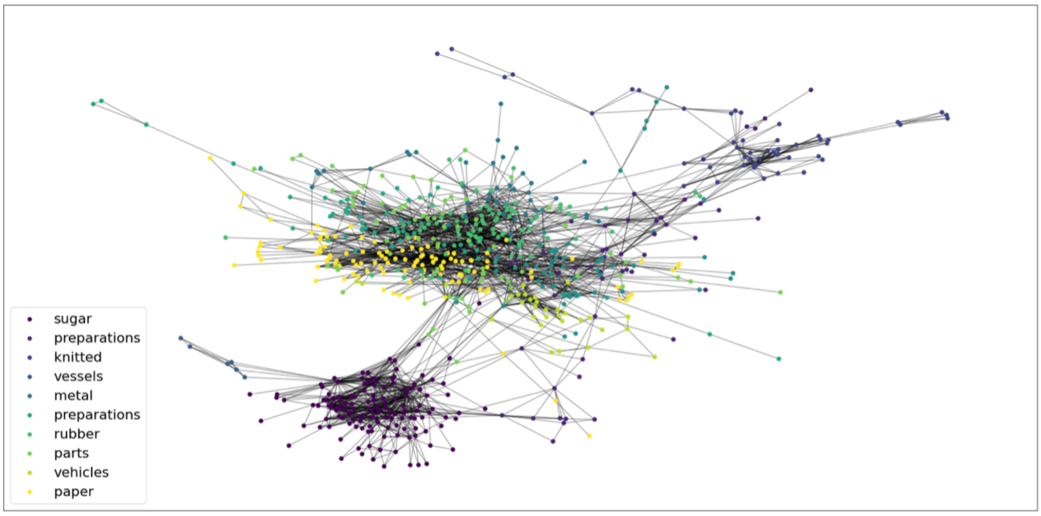


Figure 6. Product Space visualization and GraphSAGE clusters.

4.3. Clustering Performance Evaluation

Assessing the performance of our clustering algorithm is more complex compared to classification. We have selected three evaluation metrics that do not rely on the absolute values of cluster labels, but rather evaluate the clustering based on the separation of similar data into groups akin to a set of ground truth classes. This approach allows us to evaluate the effectiveness of the clustering algorithm in capturing meaningful patterns and groupings within the data, irrespective of noise or non-standard cluster shapes. Moreover, our methodology accounts for the dynamic nature of clustering tasks, wherein clusters may evolve or merge over time, ensuring a comprehensive assessment of performance under real-world conditions.

4.3.1. Rand Index

Therefore, we utilize the Rand Index [55], which calculates a similarity measure between two clusters by considering all pairs of samples. It counts the pairs that are assigned to the same or different clusters in both the predicted and true clusters. The Rand Index is a function that quantifies the similarity between two assignments while disregarding permutations.

Consider C as the ground truth class assignment and K as the clustering.

- a represents the count of pairs of elements that belong to the same set in both C and K.
- b denotes the number of pairs of elements that are in different sets within both C and K.
- $c_2^{n_{samples}}$ represents the total number of potential pairs within the dataset, where $n_{samples}$ denotes the number of samples.

$$\text{Rand Index } = \frac{a+b}{c_2^{n_{\text{samples}}}} \tag{8}$$

4.3.2. Mutual Information Score

Mutual Information [56] measures the similarity between two sets of labels assigned to the same data. This metric is insensitive to the specific numerical values of the labels; rearranging the values of class or cluster labels does not change the score. When $|U_i|$

represents the number of samples in cluster U_i, and $|V_j|$ denotes the number of samples in cluster V_j, the Mutual Information for clustering U and V is defined as follows:

$$MI(U,V) = \sum_{i=1}^{|\mathrm{U}|}\sum_{j=1}^{|\mathrm{V}|} \frac{|\mathrm{U}_i \cap \mathrm{V}_j|}{\mathrm{N}} \log \frac{\mathrm{N}|\mathrm{U}_i \cap \mathrm{V}_j|}{|\mathrm{U}_i||\mathrm{V}_j|} \tag{9}$$

4.3.3. V-Measure

With access to the true class assignments of the samples, it becomes feasible to establish a meaningful metric through the examination of conditional entropy. Specifically, Rosenberg and Hirschberg [57] delineated two commendable goals for any cluster allocation:

- Homogeneity: Every cluster is comprised solely of members belonging to a singular class.
- Completeness: Every member belonging to a particular class is assigned to one cluster.

The V-measure, which is their harmonic mean, is calculated using the following formula, where we use β by its default value 1:

$$V = \frac{(1+\beta) \times \text{ homogeneity } \times \text{ completeness}}{\beta \times \text{ homogeneity } + \text{ completeness}} \tag{10}$$

Rand Index, Mutual Information Score, and V-measure metrics address the inherent complexity of clustering tasks by robustly evaluating the similarity between obtained clusters and true classes. They accommodate challenges such as noise and varying cluster shapes by focusing on structural correspondence rather than absolute label values. Thus, we used these metrics to compare the clusters with the classification that is proposed in the datasets. For the Product Space, each HS code (the nodes), belong to a sector of activity in the sense of the HS nomenclature. For the Cora dataset, each publication is attached to a potential scientific field.

5. Results

5.1. Theoretical Evaluation

Tables 2 and 3 displays the outcomes delivered by Louvain and I-Louvain and GraphSAGE with k-means. In this experiment, we obtain identical results for both datasets when using ground truth. These results confirm the effectiveness of employing the deep graph learning method: GraphSAGE. For Product Space, the Mutual Info Score of GraphSAGE is equal to 0.498, when it is equal to 0.378 for Louvain. Moreover, with a Mutual Info Score of 0.498, GraphSAGE outperforms I-Louvain which obtains only 0.408. We obtain the same results on the Cora dataset, except for I-Louvain. This variance can be elucidated by the utilization of one-hot encoding for Cora's textual data, so the calculation of inertia is less efficient with binary data. For all the tests we have carried out on Product Space, the order of performance is almost always the same, with I-Louvain slightly outperforming Louvain and GraphSAGE outperforming the other two modularity-based methods by a larger margin. These findings validate the value of deep graph learning in enhancing community detection. Indeed, the Louvain algorithm tends to restrict the number of discovered communities, which can explain the results according to the Rand Index. By contrast, the number of clusters for GraphSAGE tended to be higher and were better matches for the expected number of categories. However, the whole Louvain process is relatively faster than with the computationally intensive GraphSAGE method. This is an advantage when applying these approaches to large networks. There were no significant differences in computation time between the GraphSAGE and I-Louvain methods. However, it is during the pre-processing of textual data into numerical representations using Word2Vec that the learning time seems to increase in proportion to the volume of textual data to be encoded.

Table 2. Evaluation of Product Space according to metrics. Assessments were calculated five times, and the results are the mean and standard deviation for the five runs.

	Louvain	**I-Louvain**	**GraphSAGE**
Rand Index	0.674 ± 0.003	0.675 ± 0.001	0.689 ± 0.001
Mutual Info Score	0.378 ± 0.014	0.408 ± 0.001	0.498 ± 0.003
V-measure	0.167 ± 0.006	0.176 ± 0.001	0.207 ± 0.002

Table 3. Evaluation of Cora according to metrics. Assessments were calculated five times, and the results are the mean and standard deviation for the five runs.

	Louvain	**I-Louvain**	**GraphSAGE**
Rand Index	0.839 ± 0.002	0.801 ± 0.001	0.848± 0.007
Mutual Info Score	0.911 ± 0.027	0.902 ± 0.001	0.924 ± 0.030
V-measure	0.448 ± 0.005	0.395 ± 0.001	0.498 ± 0.018

5.2. Practical Evaluation

To evaluate the influence of graph clustering on industrial diversification recommendations, we conducted interviews with industrial professionals. For anonymization and confidentiality regarding strategic details, we refrained from disclosing the names of the companies involved. The focus of these companies predominantly revolves around the production of two specific product codes:

- 9031: Measuring or checking instruments, appliances, and machines
- 8465: Machine tools, incl. machines for nailing, stapling, gluing or otherwise assembling, for working wood, cork, bone, hard rubber, hard plastics, or similar hard materials.

From these product codes and Hidalgo's Product Space, the diversification recommendations follow the edges of the graph (as in Figure 1). In our work, we made the choice to also use neighboring products at the second level (neighbor of neighbor), so we had many possible recommendations for the products 9031 and 8465, respectively.

Thus, the use of graph clustering methods in the Product Space makes it possible to refine the number of recommendations to select only the neighbors that belong to the same clusters as the products 9031 and 8465.

This filtering of industrial diversification recommendations by Product Space clustering reduces the possibilities by offering more relevant recommendations. The relevance of the recommendations was verified through interviews with industrial companies that manufacture products 9031 and 8465. The results of these interviews show that the recommendations filtered in the GraphSAGE algorithms are the most relevant and possible for manufacturers without much modification of their production equipment. In addition, for a manufacturer, GraphSAGE's recommendation for the 8465 product provided the 8480 product, which is a product that the manufacturer already produced several years ago, which is a molding box for a metal foundry.

These field results show that the filtering of recommendations by GraphSAGE provides more relevant industrial diversification recommendations than with the Louvain and I-Louvain methods. Moreover, the filtering of these modularity-based methods is less important because they maintain many more irrelevant recommendations than GraphSAGE.

6. Conclusions and Perspectives

This study assesses the effectiveness of graph learning techniques for filtering recommendations from macroeconomic graphs such as the Product Space. The proposed graph learning method is applied to the Product Space dataset, which reflects the similarity in industrial expertise among product classes within the HS nomenclature, irrespective of geographical boundaries. The resulting clusters filter the many recommendations offered by the Product Space. The recommendations propose diversifications of production for manufacturers to mitigate the risk of shortages while promoting local innovation and green economic growth. The use of deep graph learning methods, in particular, GraphSAGE, allows updating node embedding for better representation in their graph. This approach demonstrates improved performance in community detection across multiple datasets, compared with alternative methods that are based on modularity. Additionally, the recommendations offered by GraphSAGE filtering are more relevant when tested in the field with manufacturers.

While GraphSAGE performs better for node clustering in graphs, it is essential to consider these scientific limitations when applying them in specific contexts. GraphSAGE can be computationally and memory-intensive, especially for large graphs. Handling massive graphs requires significant computational power, which may limit its usage in certain applications. Like many neural-network-based machine learning techniques, interpreting GraphSAGE results can be challenging. Understanding how clusters are formed and why certain nodes are grouped together is complex. Also, the random neighbor sampling approach can lead to sampling bias and in some cases not adequately explore the various neighborhoods of the graph. GraphSAGE might exhibit suboptimal performance on graphs characterized by a high degree of homophily, wherein nodes sharing similar attributes are likely to be connected. To make diversification recommendations based on the Product Space, we are limited by this data. Indeed, the high degree of homophily in the Product Space graph is a limitation. The consequence is that products with similar text descriptions tend to be already connected. So, for products with a low number of neighbors, recommendation filtering will not add value in these cases. One of the shortcomings of our work concerns the generalization of our results to other industry-specific contexts. By working with Product Space, the application of our methods limits us to other contexts such as the service sector and made-to-measure production, as we are working with products from the harmonized system. This specificity limits the contribution of our results and highlights the need for caution when extrapolating them to different fields. The lack of adaptability to sectors other than industrial production justifies further research on databases other than Product Space to obtain results in non-standardized or service-oriented environments. As a result, Product Space may lack universality, underlining the importance of context-specific interpretations and applications. Working with Product Space means being time sensitive. Indeed, industrial production is subject to changes over time, such as market dynamics, technological advances, and economic fluctuations. Although we have worked with several historical versions of Product Space, the time dimension influences future recommendations for industrial diversification.

The limitations described above will guide future plans and research prospects. This study may spark further research into more advanced and tailored graph learning techniques for Product Space clustering. Exploring variations in existing graph-based methods or developing novel algorithms can lead to more accurate and efficient clustering results, especially in scenarios where the Product Space is complex and high-dimensional. Exploring innovative algorithms and incorporating advanced methods like federated learning or edge computing could enhance the accuracy and efficiency of clustering results, especially in complex and high-dimensional Product Space scenarios. Considering the dynamic nature of industrial production and economic activities, future studies could investigate how to incorporate temporal dynamics into Product Space clustering. This may involve analyzing the evolution of product connections and studying how the clustering results change over time, enabling policymakers to make informed decisions about diversification strategies. Leveraging external data sources, such as trade data, supply chain information, or macroeconomic indicators, could enrich the Product Space analyses. Integrating such data could lead to a more comprehensive understanding of the factors that influence industrial diversification and help in identifying potential growth opportunities. Also, the study can expand its applicability by developing a framework that accommodates industries beyond the standard harmonized system nomenclature proposed by Product Space, such as service-based sectors and custom production. This could involve devising new clustering and graph learning techniques tailored to diverse industrial landscapes.

Focusing on interpreting the cluster insights obtained from the Product Space clustering can provide valuable knowledge for policymakers and industry stakeholders. This will require the development of user-friendly software tools or platforms that enable industry professionals and policymakers to obtain diversification recommendations without the need for a comprehensive grasp of the underlying technology. By pursuing these research directions, the study does not just contribute to a more comprehensive understanding of industrial diversification, but also offers practical solutions and insights for decision-makers in various industries.

This study offers a foundation for further research in the fields of economic diversification and industrial development. These scientific perspectives can advance the understanding of Product Space analysis and provide valuable insights for policymakers and industrial stakeholders seeking to promote economic growth and diversification.

Author Contributions: Conceptualization, K.C.; methodology, K.C.; software, K.C.; validation, K.C.; formal analysis, K.C.; investigation, K.C.; resources, K.C.; data curation, K.C.; writing—original draft preparation, K.C.; writing—review and editing, K.C. and F.C.; visualization, K.C.; supervision, A.A.-K. and F.C.; project administration, A.A.-K. and F.C.; funding acquisition, A.A.-K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the company OpenStudio through a Convention Industrielle de Formation par la Recherche (CIFRE) implemented by the Association National Recherche Technologie (ANRT) N°2021/0563 supported by the French Ministry of Higher Education, Research, and Innovation (MESRI). The APC was funded by OpenStudio.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data, experimentations and results presented in this study are openly available in CodeOcean at https://doi.org/10.24433/CO.5620796.v1.

Acknowledgments: This paper and the research behind it would not have been possible without the help of Jérôme Cuny and our team at OpenStudio: Taoufik Jarmouni, Jean-Luc Marini, Marion Laurent and Jérémy Boiraud.

Conflicts of Interest: The authors declare that this study received funding from OpenStudio. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript or in the decision to publish the results.

References

1. Hendricks, K.B.; Singhal, V.R.; Zhang, R. The effect of operational slack, diversification, and vertical relatedness on the stock market reaction to supply chain disruptions. *J. Oper. Manag.* **2009**, *27*, 233–246. [CrossRef]
2. Grillitsch, M.; Asheim, B. Place-based innovation policy for industrial diversification in regions. *Eur. Plan. Stud.* **2018**, *26*, 1638–1662. [CrossRef]
3. Wagner, J.E. Regional Economic Diversity: Action, Concept, or State of Confusion. *J. Reg. Anal. Policy* **2000**, *30*, 22. . [CrossRef]
4. Sierzchula, W.; Bakker, S.; Maat, K.; Van Wee, B. Technological diversity of emerging eco-innovations: A case study of the automobile industry. *J. Clean. Prod.* **2012**, *37*, 211–220. [CrossRef]
5. Lu, J.W.; Ma, X. The Contingent Value of Local Partners' Business Group Affiliations. *Acad. Manag. J.* **2008**, *51*, 295–314. [CrossRef]
6. Hidalgo, C.A.; Hausmann, R. The building blocks of economic complexity. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 10570–10575. [CrossRef] [PubMed]
7. Hidalgo, C.A.; Klinger, B.; Barabási, A.L.; Hausmann, R. The Product Space Conditions the Development of Nations. *Science* **2007**, *317*, 482–487. [CrossRef] [PubMed]
8. Hausmann, R.; Hidalgo, C.A. The network structure of economic output. *J. Econ. Growth* **2011**, *16*, 309–342. [CrossRef]
9. Chaplin, P. An Introduction to the Harmonized System. *NCJ Int'l L. Com. Reg.* **1987**, *12*, 417.
10. Desmarchelier, B.; Regis, P.J.; Salike, N. Product space and the development of nations: A model of product diversification. *J. Econ. Behav. Organ.* **2018**, *145*, 34–51. [CrossRef]
11. Pachot, A.; Albouy-Kissi, A.; Albouy-Kissi, B.; Chausse, F. Production2Vec: a hybrid recommender system combining semantic and product complexity approach to improve industrial resiliency. In Proceedings of the 2021 2nd International Conference on Artificial Intelligence and Information Systems, Chongqing, China, 26–28 Novermber 2021; pp. 1–6. [CrossRef]
12. DuBois, T.; Golbeck, J. Improving Recommendation Accuracy by Clustering Social Networks with Trust. *Recomm. Syst. Soc. Web* **2009**, *532*, 1–8.

13. Newman, M.E.J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [CrossRef] [PubMed]
14. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [CrossRef]
15. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
16. Fraccascia, L.; Giannoccaro, I.; Albino, V. Green product development: What does the country product space imply? *J. Clean. Prod.* **2018**, *170*, 1076–1088. [CrossRef]
17. Nomaler, Ö; Verspagen, B. Some New Views on Product Space and Related Diversification. *arXiv* **2022**, arXiv:2203.16316.
18. Pachot, A.; Albouy-Kissi, A.; Albouy-Kissi, B.; Chausse, F. Multiobjective recommendation for sustainable production systems. In Proceedings of the MORS workshop held in conjunction with the 15th ACM Conference on Recommender Systems (RecSys), Amsterdam, The Netherlands, 27 September–1 October 2021; Volume 1.
19. Moradi, P.; Ahmadian, S.; Akhlaghian, F. An effective trust-based recommendation method using a novel graph clustering algorithm. *Phys. A Stat. Mech. Appl.* **2015**, *436*, 462–481. [CrossRef]
20. Rostami, M.; Oussalah, M.; Farrahi, V. A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering. *IEEE Access* **2022**, *10*, 52508–52524. [CrossRef]
21. Li, X.; Hu, Y.; Sun, Y.; Hu, J.; Zhang, J.; Qu, M. A Deep Graph Structured Clustering Network. *IEEE Access* **2020**, *8*, 161727–161738. [CrossRef]
22. Yang, L.; Wang, S.; Tao, Y.; Sun, J.; Liu, X.; Yu, P.S.; Wang, T. DGRec: Graph Neural Network for Recommendation with Diversified Embedding Generation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 661–669. [CrossRef]
23. Ren, Y.; Ni, H.; Zhang, Y.; Wang, X.; Song, G.; Li, D.; Hao, J. Dual-Process Graph Neural Network for Diversified Recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, Birmingham, UK, 21–25 October 2023; pp. 2126–2135. [CrossRef]
24. Erdös, P.; Renyi, A. On the Strength of Connectedness of a Random Graph. *Acta Math. Hung.* **1961**, *12*, 261–267. [CrossRef]
25. Abbe, E. Community Detection and Stochastic Block Models: Recent Developments. *J. Mach. Learn. Res.* **2018**, *18*, 1–86.
26. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
27. Gómez, A.M.E.; Paynabar, K.; Pacella, M. Functional directed graphical models and applications in root-cause analysis and diagnosis. *J. Qual. Technol.* **2021**, *53*, 421–437. [CrossRef]
28. Xing, Z.; Zhao, S.; Guo, W.; Meng, F.; Guo, X.; Wang, S.; He, H. Coal resources under carbon peak: Segmentation of massive laser point clouds for coal mining in underground dusty environments using integrated graph deep learning model. *Energy* **2023**, *285*, 128771. [CrossRef]
29. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. *arXiv* **2018**, arXiv:1706.02216.
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
31. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**. arXiv:1710.10903.
32. De Nicolò, F.; Monaco, A.; Ambrosio, G.; Bellantuono, L.; Cilli, R.; Pantaleo, E.; Tangaro, S.; Zandonai, F.; Amoroso, N.; Bellotti, R. Territorial Development as an Innovation Driver: A Complex Network Approach. *Appl. Sci.* **2022**, *12*, 9069. [CrossRef]
33. Tajoli, L.; Piccardi, C.; Hoang, V.P. The Structural Change of World Trade from 1996 to 2019. A Network Approach. Available online: https://fondazionemasi.it/public/masi/files/ITSG/Salerno2022/TajoliPiccardiHoang.pdf (accessed on 18 February 2024).
34. Korniyenko, Y.; Pinat, M.; Dew, B. Assessing the Fragility of Global Trade. *IMF Work. Pap.* **2017**, *2017*, 38. [CrossRef]
35. Chessa, M.; Persenda, A.; Torre, D. Brexit and Canadadvent: An application of graphs and hypergraphs to recent international trade agreements. *Int. Econ.* **2023**, *175*, 1–12. [CrossRef]
36. Zhang, L.; Priestley, J.; DeMaio, J.; Ni, S.; Tian, X. Measuring Customer Similarity and Identifying Cross-Selling Products by Community Detection. *Big Data* **2021**, *9*, 132–143. . [CrossRef]
37. Kafkas, K.; Perdahçı, Z.N.; Aydın, M.N. Discovering Customer Purchase Patterns in Product Communities: An Empirical Study on Co-Purchase Behavior in an Online Marketplace. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 2965–2980. [CrossRef]
38. Feng, X.; Rutherford, A. The Dynamic Resilience of Urban Labour Networks. *arXiv* **2022**, arXiv:2202.12856.
39. Lü, L.; Pan, L.; Zhou, T.; Zhang, Y.C.; Stanley, H.E. Toward link predictability of complex networks. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 2325–2330. [CrossRef] [PubMed]
40. Mungo, L.; Lafond, F.; Astudillo-Estevez, P.; Farmer, J.D. Reconstructing production networks using machine learning. *J. Econ. Dyn. Control* **2023**, *148*, 104607. [CrossRef]
41. Wu, D.; Wang, Q.; Olson, D.L. Industry classification based on supply chain network information using Graph Neural Networks. *Appl. Soft Comput.* **2023**, *132*, 109849. [CrossRef]
42. Benita, F.; Sarica, S.; Bansal, G. Testing the static and dynamic performance of statistical methods for the detection of national industrial clusters. *Pap. Reg. Sci.* **2020**, *99*, 1137–1157. [CrossRef]

43. Combe, D.; Largeron, C.; Géry, M.; Egyed-Zsigmond, E. I-Louvain: An Attributed Graph Clustering Method. In *Advances in Intelligent Data Analysis XIV*; Fromont, E., De Bie, T., van Leeuwen, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switerland, 2015; Volume 9385, pp. 181–192. [CrossRef]
44. Traag, V.A.; Waltman, L.; Van Eck, N.J. From Louvain to Leiden: Guaranteeing well-connected communities. *Sci. Rep.* **2019**, *9*, 5233. [CrossRef]
45. Hamilton, W.L.; Ying, R.; Leskovec, J. Representation Learning on Graphs: Methods and Applications. *arXiv* **2018**, arXiv: 1709.05584.
46. Jiang, S.; Luo, J. Technology Fitness Landscape for Design Innovation: A Deep Neural Embedding Approach Based on Patent Data. *J. Eng. Des.* **2022**, *33*, 716–727. [CrossRef]
47. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. *arXiv* **2016**, arXiv:1607.00653.
48. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-Means Clustering Algorithm. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1979**, *28*, 100–108. [CrossRef]
49. Yuan, C.; Yang, H. Research on K-Value Selection Method of K-Means Clustering Algorithm. *J* **2019**, *2*, 226–235. [CrossRef]
50. Mccallum, A.K. Automating the Construction of Internet Portals with Machine Learning. *Inf. Retr.* **2000**, *3*, 127–163. [CrossRef]
51. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
52. van Capelleveen, G.; Amrit, C.; Zijm, H.; Yazan, D.M.; Abdi, A. Toward building recommender systems for the circular economy: Exploring the perils of the European Waste Catalogue. *J. Environ. Manag.* **2021**, *277*, 111430. [CrossRef] [PubMed]
53. Swoboda, T.; Hemmje, M.; Dascalu, M.; Trausan-Matu, S. Combining Taxonomies using Word2vec. In Proceedings of the 2016 ACM Symposium on Document Engineering, Vienna, Austria, 13–16 September 2016; pp. 131–134. [CrossRef]
54. Fruchterman, T.M.J.; Reingold, E.M. Graph drawing by force-directed placement. *Soft. Pract. Exp.* **1991**, *21*, 1129–1164. [CrossRef]
55. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [CrossRef]
56. Strehl, A.; Ghosh, J. Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
57. Rosenberg, A.; Hirschberg, J. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.