Node2Vec Stability: Preliminary Study to Ensure the Compatibility of Embeddings with Incremental Data Alignment

Oumaima El Haddadi, Max Chevalier, Bernard Dousset, Ahmad El Allaoui, Anass El Haddadi, Olivier Teste

Node2Vec stability: preliminary study to ensure the compatibility of embeddings with incremental data alignment

Oumaima El Haddadi[1,2], Max Chevalier[1], Bernard Dousset[1], Ahmad El Allaoui[2], Anass El Haddadi[2], and Olivier Teste[1]

[1] IRIT, SIG, Université de Toulouse, CNRS, France
lastname.firstname@irit.fr
[2] LSA, SDIC, ENSAH, Abdelmalek Essaadi University, Tetouan, Morocco
lastname.firstname@uae.ac.ma

Abstract. In dynamic information systems, data alignment addresses challenges like data heterogeneity, integration, and interoperability by connecting diverse datasets. To ensure the stability and effectiveness of these alignments over time, an incremental process may be required, allowing the alignments to be updated as the data evolves. While embedding-based methods are valuable for handling incremental data in the graph learning field, they are underexplored in data alignment. However, before implementing such an approach, it is essential to verify the stability of the embeddings in order to guarantee their reliability and temporal consistency. So, we study the most promising model (i.e. Node2Vec) that exhibits favourable stability in embeddings, particularly with respect to the stability of node embeddings. Despite potential variability in pairwise similarities, the idea of an incremental approach remains reliable, especially with a fixed model. Implementing such an approach can efficiently manage data dynamics in information systems with reduced resource needs. By applying this incremental process to data alignment, it will be possible to efficiently manage heterogeneous data in dynamic information system environments, while minimising resource requirements.

Keywords: Heterogeneous Data · Incremental Data Alignment · Embedding Stability · Dynamic Environment.

1 Introduction

In today's dynamic information systems context, data alignment addresses the challenges of integrating diverse data sources coherently [1]. This approach identifies alignments (matches) between different data sources, including schemas and instances, to manage their heterogeneity effectively. Among the various alignment methodologies, embeddings offer a powerful way to represent and compare data from diverse sources, revealing underlying similarities and relationships that might be elusive with other methods [2]. The alignment results are used in both requirements-centered storage approaches, such as data integration or schema

mapping [3], and non-requirements-centered approaches, such as data lakes [4] or ontologies matching [5].

However, dynamics in data sources requires maintaining and updating alignments as sources evolve, incurring high computational costs. To optimize resource use and ensure long-term alignment efficiency, we aim to propose an illustrative solution for incremental data alignment based on embeddings. This approach enables alignment updates as sources evolve. It can serve as a solution for managing dynamic data, eliminating the necessity of totally recalculating alignments. This incremental process should support a wide range of data changes, including data additions, modifications, and deletions.

Prior to apply incremental embedding approach to achieve incremental data alignment, it's crucial to assess the stability of obtained embeddings to ensure alignment reliability and temporal consistency. The study of embedding stability involves evaluating their sensitivity to data and learning condition variations, ensuring robust alignment maintenance over time. In this paper, our focus is solely on the learning conditions. This assessment determines the reliability of incremental data alignment in dynamic information systems, eliminating the need for recalculating alignments from scratch. To achieve this, we address in our research two research questions: 'How can we study embedding stability?' and 'What is the impact of unstable embeddings on the incremental alignment process?'

In this article, we focus on embedding stability in the following structure: Section 2 defines embedding stability. Section 3 provides a review. Section 4 studies the stability of the most promising embedding model for data-alignment: Node2Vec.

2 Definition of embedding stability

According to [7], stability in this context is observed when the embedding method consistently manages to generate similar representations for the same data across multiple runs. Furthermore, within the scope of this research, [8] aims to pinpoint the key factors responsible for the instability of the embedding method, drawing from training data. By merging these two objectives, the study of embedding stability can generally be defined as the exploration of the coherence and robustness of numerical vector representations of data over time, in response to data variations and learning conditions.

3 How is embedding stability studied in the literature?

To examine the compatibility and stability of the embedding method within the context of incremental data alignment, it is crucial to consider the concept of stability of embeddings. Several studies have explored the stability of embedding methods in various domains. For example, [7] investigated the sensitivity of embeddings obtained from corpora (documents) in relation to classification. The study found that distances between nearest neighbors are highly sensitive

to minor changes in the corpus, particularly for smaller corpora. As a recommendation, they suggest employing multiple embedding methods and averaging the distances for corpus classification (it is unclear whether this method is computationally expensive in terms of volume and processing time). Similarly, [6] examined the stability of word embedding methods (specifically Word2Vec, fastText, and GloVe) for clustering. The results indicated that fastText exhibited greater stability compared to the other two methods. However, it was observed that the stability of Word2Vec improved with multiple executions.

As highlighted in the article by [8], it is important to note that for each execution approach, if reapplied, the constructed embeddings differ even for the same initial data. However, after a series of executions, the model stabilizes. In the context of a dynamic graph, aligning the embeddings between new and previous executions becomes essential. Therefore, considering the insights from these studies, it is necessary to investigate the stability and sensitivity of the chosen embedding method in the context of incremental data alignment. This will help determine the appropriate alignment approach and ensure reliable and consistent results when dealing with evolving data sources.

Nonetheless, it is crucial to underline that the stability of the Node2Vec method (a graph embedding technique we intend to support our incremental alignment approach) has not been extensively explored in prior studies. Furthermore, these studies exclusively relied on cosine similarity to compute the likeness between vector representations. In our specific case study, executing the full Node2Vec-based method may not be imperative. However, a thorough stability analysis of this method remains indispensable to assess any potential risks when deploying it in the context of evolving data sources. By carrying out a stability analysis, we can glean valuable insights into the robustness and reliability of the Node2Vec method itself, as well as its effectiveness when processing dynamic data sources. This contribute to a deeper comprehension of the constraints and potential challenges associated with using the model on evolving datasets. Aligned with the provided definition, we are confronted with two research objectives:

- Examine the stability of the method regarding the condition learning, for example, the selection of hyperparameters (see Section 4. 1 for details).
- Investigate the stability of the method when data changes.

Note that this paper only focuses on the first objective. In the following section we introduce our experiments.

4 Experiments: studying embedding stability of Node2Vec model

In the context of incremental data alignment, we have opted for the incremental embedding approach [9]. The main objective of this approach is to learn representations from temporal graphs. It involves generating embeddings for the new

nodes using the embedding matrix computed for the initial graph and updating the embeddings of influenced nodes (usually neighbors nodes). The method follows these steps:

- Compute the embedding matrix for the existing graph using a standard graph embedding technique (Node2Vec, as proposed by [10]).
- When new nodes are added, generate embeddings for these nodes and update the unfluenced nodes using Algorithm 1 and Algorithm 2 introduced by [9].

At present, the selected approach only takes into account data additions and modifications. We are actively looking for methods to update alignments in the case of data deletions. Before delving into further development, our first step is to study the stability of Node2Vec, as outlined in the following description section.

4.1 Description and Setup of our experiments

Node2vec is unsupervised learning method aims to learn node representations in graphs. It aims to capture the structural and contextual similarity between graph nodes by assigning dense vectors in a reduced-dimensional space. The technique is based on Word2Vec techniques, which is commonly used for word embedding, but adapted to the graph domain.

To investigate the stability of Node2Vec, we construct a graph using the methodology proposed by [2]. The graph is generated from two distinct sources defined by [11]: Authors 1 (with attributes: Authors, Title, DOI, Year, EID, Source Title, and Cited By) and Authors 2 (including attributes: EID, Authors, Cited By, Country, Document Type, City, Access Type, and Aggregation Type). By merging data from these sources, we create a graph that encompasses both instances and schema elements. The resulting graph (Figure 1) comprises 219 nodes and 870 edges, illustrating relationships between different elements.

Node embeddings are generated using the Node2Vec method, implemented in the Python library [12]. The model is executed 10 times on the graph, employing the following parameters:

$$\text{DIMENSION}=100,\ \text{WALK\_LENGTH}=30,\ \text{NUM\_WALKS}=10,\ \text{WINDOW}=10$$

In both datasets, we identified three true alignments of schema elements: EID *(Authors1.EID–Authors2.EID)*, Authors *(Authors1.Authors–Authors2.Authors)*, and Cited By *(Authors1.Cited By–Authors2.Cited By)*. To assess the model's performance, we computed the Root Mean Square Error (RMSE) metric for the 10 scores alignments obtained. According to Table 1, the RMSE values are quite low, indicating that the predictions are close to the actual values.

To address the first research question on how to study stability, we define two scenarios: one focusing on the dynamics of random walks and the other involving stable random walks.

- **Scenario 1:** we perform random walks independently for each of the 10 runs.

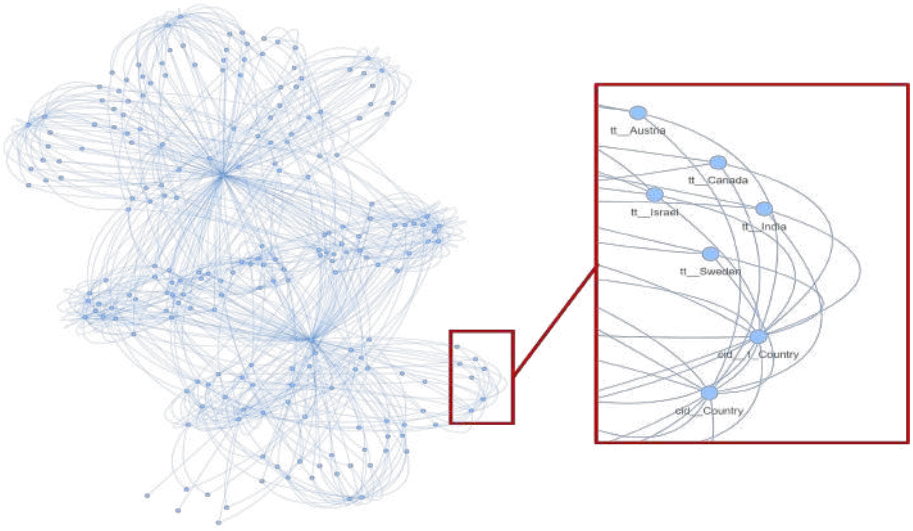

Fig. 1. The left image shows the entire graph obtained. The right image shows a zoomed-in view

– **Scenario 2:** we keep the same walks generated during the initial run for the subsequent 9 runs.

These two scenarios allow us to study the stability of Node2Vec independently from different perspectives, without considering the source changes.

Table 1. Model performance

	EID	**Authors**	**Cited by**
RMSE	0.055	0.063	0.052

To examine the stability of the embeddings in the both scenarios, we defined two strategies. Firstly, we compute the similarity of each node to the centroid of the embedding obtained from the 10 runs for the same node. Secondly, we compute the pairwise similarity between nodes within each run and compare these similarities. These strategies enable us to assess the consistency and stability of the embeddings in different contexts.

To compute similarity, we use the following metrics. Unlike some other researchers, we do not solely rely on cosine similarity (Equation 1), which yields results ranging from -1 to 1. Instead, we employ three other metrics that produce similarity scores within the range of 0 to 1. To achieve this, we normalize the cosine similarity as shown in Equation 2. Additionally, we have defined two other metrics (Equations 3 and 4). According to Figure 2 (the figure presents three similarity metrics within the range [0,1]), the normalized cosine curve remains

relatively flat between strong similarity scores of 0.95 to 1 and weak similarity scores of 0 to 0.5, indicating minimal differences. Conversely, the opposite similarity is more sensitive within this range. On the other hand, the normalized cosine is more sensitive to scores ranging from 0.5 to 0.95.

Cosine similarity (cosine defined by [13]):

$$cos(\Theta) = u \cdot v / \|u\| * \|v\| \tag{1}$$

Normalized cosine similarity in the range [0:1]:

$$S_{cos_{norm}} = (cos(\Theta) + 1)/2 \tag{2}$$

Linear similarity:

$$S_{lin} = 1 - \Theta/\Pi \; ; \; where: \quad \Theta = arcos(cos_{norm}) \tag{3}$$

Opposite similarity:

$$S_{opp} = 2 * S_{lin} - S_{cos_{norm}} \tag{4}$$

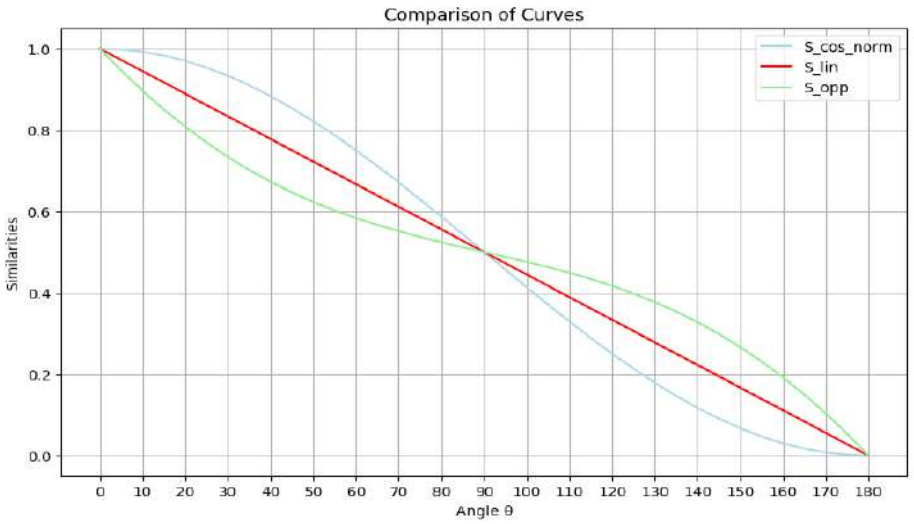

Fig. 2. The figure reveals that cosine similarity (blue curve) tends to be more optimistic than linear similarity (red curve). In contrast, opposite similarity (green curve) demonstrates the opposite effect of cosine similarity compared to linear similarity.

To compare the variation in the obtained results, we compute the dispersion value. Dispersion refers to the distribution of values within a dataset, measuring

the degree of separation between values. In the context of embeddings or similarity scores, dispersion can provide insights into the diversity or similarity of node pairs in a particular graph. In our case, dispersion is the average of the standard deviations obtained for all node embeddings across the 10 runs.

$$Dispersion = 1/n \sum_{i=1}^{n} std_i \; ; \tag{5}$$

where:

- *n is the total number of node embeddings*
- *std_i is the standard deviation of similarity scores for the i_{th} node embeddings across the 10 runs*

4.2 Results and discussion

We obtained the following results after evaluating the different scenarios:

- **Scenario 1 (random walk):** We conducted 10 runs of the model, generating 10 embedding matrices (each 219x100). For each node, we computed similarity to the centroid of the 10 embeddings for that node and the similarity between pairs of nodes in each matrix, resulting in a 23762x10 pairwise similarity matrix. The dispersion for both cases using four similarity metrics is shown in Table 2. The dispersion of similarity values between nodes and the centroid is relatively small (0.004 to 0.008), indicating stable embeddings. However, normalized cosine similarity between pairs of nodes is less stable (0.0013).
  To gain a deeper understanding, we conducted another experiment.we focused on pairs with normalized cosine similarity values greater than 0.95, aiming to ensure a higher level of similarity between the nodes. The results revealed that only a very low percentage of such pairs 0.63% of such pairs (out of 23762) met the criteria in the first run (Figure 3). According to the figure, we observe a slight difference between the different results of each execution. When we display the alignment results obtained during the 10 executions and compare them with the three true alignments, we consistently obtain the three expected alignments.
- **Scenario 2 (walk fixed):** In this scenario, we also executed the model 10 times, resulting in 10 embedding matrices of size (219x100). Similar to Scenario 1, we calculated the similarity between each node and the centroid of each embedding node.
  Throughout our observations, we consistently found that the similarity between each node's embedding and the centroid, as well as the similarity between pairs of nodes, remained consistently perfect, with a dispersion value of 0.0. This indicates that the embeddings remain stable when using the

same walks generated in the first run for subsequent runs. The high consistency in similarity values demonstrates the reliability and robustness of the embeddings in capturing the underlying patterns and relationships in the data across multiple runs.

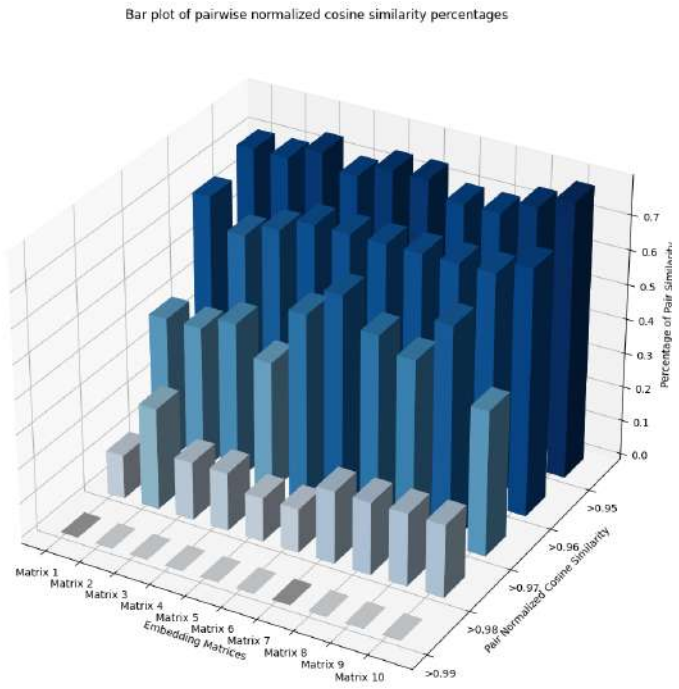

Fig. 3. Bar Plot of the Percentage of pairwise normalized cosine Similarities greater than [0.95, 0.96, 0.97, 0.98, 0.99]

To resume, the embeddings obtained in this study demonstrate desirable stability in capturing node embedding stability and pairwise similarities. The analysis of dispersion of node-to-centroid similarities showed relatively small values, indicating that the embeddings remained stable across the 10 runs. However, pairwise similarities exhibited slightly higher dispersion values, suggesting some variability in pairwise relationships. Despite this, the embeddings remained consistently perfect in both cases, with a dispersion value of 0.0, emphasizing their reliability when using the same walks for subsequent runs. Additionally, the experiment focusing on pairs with high similarity values further reinforced the embeddings' stability. Furthermore, the study highlights the importance of considering the efficiency of fixing the walks in the Node2Vec model.

Table 2. Minimum, Maximum and dispersion of embeddings in both scenarios

			Cosine Similarity Dispersion	Cosine Normalized Dispersion	Lin Similarity Dispersion	Opposite Similarity Dispersion
Random Walk	Centroid	min	0.004	0.002	0.002	0.003
		max	0.016	0.008	0.010	0.012
		dis	**0.008**	**0.004**	0.006	**0.008**
	Pairwise	min	0.004	0.002	0.001	0.0
		max	0.055	0.027	0.014	0.011
		dis	0.026	**0.013**	0.005	0.002

These findings highlight the importance of verifying the stability of embeddings before implementing any approach to ensure their reliability and temporal consistency, especially in dynamic information systems. Striking a balance between stability and efficiency is crucial in developing an optimal data alignment solution for dynamic information systems. Further research and experimentation are necessary to determine the most effective configuration and parameters for the Node2Vec model to ensure both stability and computational efficiency in the incremental data alignment process.

5 Conclusion and outlook

Incremental alignment methods can play a crucial role in resource optimization within dynamic information systems, reducing computational costs associated with recalculating alignments as data sources evolve. In line with this, our study focuses on examining the stability of Node2Vec, a graph embedding model that we intend to use in the context of incremental data alignment. Evaluation results reveal a favorable embedding stability, especially with fixed generated walks. Despite this result, the existing embeddings were still effective in supporting incremental data alignment without the need for frequent recomputations. However, if frequent recomputations were necessary, it could impact the incremental alignment process as the embeddings might become unstable with other data sources, thus affecting the alignment quality. To improve Node2Vec's stability in dynamic data scenarios, future research and method refinement, such as exploring alternative models like fastText as demonstrated by [6], could improve performance.

In light of these promising results, it is imperative to direct our focus towards advancing the incremental alignment process itself. For future work, it is important to study the stability according to the second research objective (data dynamism), as well as to consider the quality of the base model and the scalability of the process in handling larger datasets and real-world scenarios.

This entails evaluating the computational efficiency and memory requirements when dealing with substantial data sources, while also addressing potential challenges related to scalability. By addressing these crucial aspects, the incremental data alignment process can be further optimized, offering a robust and efficient solution for managing evolving information systems with diverse and extensive datasets.

References

1. Sutanta, E., Wardoyo, R., Mustofa, K., Winarko, E. : Survey: Models and Prototypes of Schema Matching , IJECE, vol. 6, no 3, p. 1011, juin 2016, https://doi.org/10. 11591/ijece. v6i3. 9789.
2. Cappuzzo, R., Papotti, P., Thirumuruganathan, S. : Local Embeddings for Relational Data Integration , Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, june 2020, p. 1335-1349. https://doi.org/10. 1145/3318464. 3389742.
3. Miller, RJ., Haas, LM., Hernandez, MA. : Schema Mapping as Query Discovery. In: Very Large DataBase conference (VLDB), pp. 77—88. (2000)
4. Alserafi, A., Abelló, A., Romero, O., Calders T. : Keeping the Data Lake in Form: Proximity Mining for Pre-Filtering Schema Matching. ACM Trans. Inf. Syst. **2**(38), 3 (2020)
5. Aumueller, D., Do, H-H., Massmann, S., Rahm, E. : Schema and ontology matching with COMA++. In: ACM international conference on Management of data (SIGMOD '05). Association for Computing Machinery, New York, NY, USA, pp. 906–908 (2005)
6. Borah, A., Barman, M. P., Awekar,A. : Are Word Embedding Methods Stable and Should We Care About It? , Proceedings of the 32st ACM Conference on Hypertext and Social Media, Virtual Event USA: ACM, august 2021, p. 45-55. https://doi.org/10. 1145/3465336. 3475098
7. Antoniak, M., Mimno, D. : Evaluating the Stability of Embedding-based Word Similarities , TACL, vol. 6, p. 107-119, déc. 2018, https://doi.org/10. 1162/tacl_a_00008.
8. Tagowski, K., Bielak, P., Kajdanowicz, T. : Embedding Alignment Methods in Dynamic Networks , in Computational Science – ICCS 2021, Éd., in Lecture Notes in Computer Science, vol. 12742. Cham: Springer International Publishing, 2021, p. 599-613. https://doi.org/10. 1007/978-3-030-77961-0_48.
9. Liu, X., Hsieh, P. -C., Duffield, N., Chen, R., Xie, M., Wen, X. : Real-Time Streaming Graph Embedding Through Local Actions , in Companion Proceedings of The 2019 World Wide Web Conference, San Francisco USA: ACM, mai 2019, p. 285-293. doi: 10. 1145/3308560. 3316585.
10. StreamNode2Vec,https://github. com/husterzxh/StreamNode2Vec. Last accessed 31 June 2023
11. Koutras, C., Siachamis, G., Ionescu, A., Psarakis, K., Brons, J., Fragkoulis, M., Lofi, C., Bonifati, A., Katsifodimos, A. : Valentine: Evaluating Matching Techniques for Dataset Discovery. In: IEEE 37th International Conference on Data Engineering (ICDE), (2021)
12. Node2vec python https://github. com/eliorc/node2vec. Last accessed 10 July 2023
13. Cosine https://scikit-learn. org/stable/modules/generated/sklearn.metrics.pairwise .cosine_similarity.html. Last accessed 18 July 2023