Results in Nonlinear Analysis 6 (2023) No. 2, 1–8
https://doi.org/10.31838/rna/2023.06.02.001
Available online at www.nonlinear-analysis.com

Results in Nonlinear Analysis
ISSN 2636-7556

Peer Reviewed Scientific Journal

Prediction nullity of graph using data mining

Mehdi Alaeiyan[a]*, Karrar Khudhair Obayes[a], Mohammadhadi Alaeiyan[b]

[a]*School of Mathematics, Iran University of Science and Technology, Narmak, Tehran 16846, Iran;* [b]*Faculty of Computer Engineering, K. N. Toosi University of Technology, Seyed Khandan, Shariati Ave, 16317-14191 Tehran, Iran*

Abstract

Nullity computation is widely used to determine the stability of a chemical molecule. Mainly, a molecule is presented as a graph, and the graph nullity value clarifies the strength of the molecule. Some formulas for specific graphs help us compute the nullity value, but it is challenging to remember the formula of each particular graph. However, another formula for calculating the nullity value is based on the graph rank. Nevertheless, processing time would be increased by the growth of the number of vertices of graphs. This paper suggests machine learning methods for computing the nullity value of a given graph. We leveraged random graph generation methods to collect many graph instances. Then, the experimental results on the collected dataset offer accuracy of 97.0878% for binary classification and 94.56% for value prediction.

Keywords: Nullity of a graph, Machine learning, Graph, Eigenvalue

2020 MSC: 05C50, 15A18

1. Introduction

A graph is $G=(V,E)$ defined on V as a set of vertices and E as a set of edges. G would be a simple graph if avoids self-loop and multiple-edge. Also, $A(G)$ is the adjacency matrix of the graph. The nullity of a graph, shown as $\eta(G)$, is the presence of zero as an eigenvalue in the spectrum of G where graph spectrum is the set of eigenvalues of the graph including with their repetition. Moreover, a λ is an eigenvalue of G if $\det(A(G)-\lambda I)=0$ where I is identity matrix.

Knowing the number of nullities has many applications in various fields, including chemical sciences. For instance, if the nullity of a graph extracted from a molecule is equal to zero, this indicates

Email addresses: alaeiyan@iust.ac.ir (Mehdi Alaeiyan)*; karrar.obayes@gmail.com (Karrar Khudhair Obayes); m.alaeiyan@kntu.ac.ir (Mohammadhadi Alaeiyan)

Received March 21, 2023; Accepted April 3, 2023; Online May 8, 2023

that the molecule has the minimum attraction with other molecules with low chemical activity. In contrast, if the nullity of a graph is greater than zero, the molecule would be highly reactive and unstable [1]. Also, this computation would be leveraged in the calculation of the energy of graphs [2].

There are two ways to compute $\eta(G)$. First, by calculating the eigenvalues of G and counting its multiplicity of the zero eigenvalues that its time complexity is $O(n^2)$ [3], [4]. Second, by computing the $\eta(G) = n - rank(G)$ where, $n = |V(G)|$ and $rank(G)$ is to the number of nonzero rows in its row echelon form [5]. Since the time complexity $rank(G)$ is $O(n^{2.3})$ [6], the higher the growth of n, the higher the computation time.

However, prior works categorize and parametrize some special cases [7], [8]. For instance, the nullity value of complete graphs, path graphs for an even number of vertices, and cycle graphs with $n\%4 \neq 0$ where n is the number of vertices, is zero [7]. Also, the nullity value of path graphs for an odd number of vertices is one, cycle graphs with $n\%4 = 0$ where n is the number of vertices is two, complete bipartite graphs $k_{m,n}$ is $m + n - 2$, and Wheel graphs for even number of vertices are two and for an odd number of vertices is three [7]. Consequently, a large number of studies and categorizations are required.

Therefore, computation of some graph properties as condition number [9], s expensive in terms of memory cost CPU usage. Consequently, we present a new prediction method based on machine learning to facilitate this process. In this case, we propose a large number of graphs with their nullity number and offer a set of features that can be used to generate a machine learning model.

In this paper, we leveraged machine learning techniques to classify and predict the nullity of graphs. We listed 14 features that can be used for classification to segregate the graphs with $\eta(G) \geq 1$ from those $\eta(G) = 0$. Next, we used regression models to predict the nullity value of the graph. In this respect, we collected many graphs and their relevant nullity values to be leveraged at model generation. Then, the experimental results on the collected dataset offer accuracy of 97.0878% for binary classification and 94.56% for value prediction.

Contributions: The contributions of this manuscript are highlighted here:

- Listing a set of properties leveraged for the prediction of the nullity value.
- Proposing a model to segregate the graphs with $\eta(G) \geq 1$ from those $\eta(G) = 0$.
- Proposing a model to predict $\eta(G) \geq 1$.

Organization: The remaining parts of this paper are organized as follows: Section 2 presents the related works; Section 3 offers the feature sets for prediction and 2 suggests a new approach to binary classification and nullity value prediction. Our proposed method is evaluated, and the results are presented in Section 4. Also, detailed discussions about the pearls and pitfalls of the proposed method are given in this section. Finally, the concluding remarks are described in Section 5.

2. Related Works

The classification of particular types of graphs is proposed in prior works [7]–[10]. Some of the outcomes achieved exact values, and some previous studies stated the nullity value if some classes of graphs are a member of a set of consequences [11]–[15]. Also, [16]–[19] defined the bound of the nullity value based on some properties as the maximum degree of the graph and the number of pendent vertices. On the other hand, some papers classified the rank values [20]–[23] that help us to compute the nullity values. Moreover, L. Wang [10] proposed that $\eta(G) \leq \rho(G)$ for a connected graph including a cycle or a clique, where $\rho(G)$ is the minimum size of the path cover of G. However, the minimum path problem is NP-hard [5].

In some cases, the $\eta(G)$ of unicyclic graphs of order $n \geq 5$is a member of the interval of $[0, n-4]$ [11]. The $\eta(G)$ set of bicyclic graphs of order n is $[0, n-2]$ [12]. The $\eta(G)$ set of tricyclic graphs of order $n \geq 8$ is $[0, n-4]$ [13]. The $\eta(G)$ set of bipartite graphs of order n is $\{n - 2m: m = 1, 2, 3, \ldots, n/2\}$ [14], [15].

$\eta(T)$, where T is a tree, is equivalent to $|T|$, not belonging to a maximal matching of T [16], [24]. Also, if G is a reduced bipartite graph and the graphs attaining equality are characterized, $\eta(G) \leq |G| - 2 - 2\ln 2(\Delta(G))$ [25].

According to all the above-mentioned methods, the nullity value of graphs is a complex task. In this paper, we leverage machine learning methods to see the ability of these methods to compute the exact nullity value of a graph. Also, according to our studies, this is the first work that presents the computation of the nullity value of a graph using machine learning approaches.

3. Proposed Method

This paper presents a new machine learning method for computing the nullity value of graphs. Figure 1 illustrates the block diagram of our proposed method. For the generation of a dataset, we propose an algorithm discussed in Section 3.1, and their nullity value is computed based $\eta(G) = n - rank(G)$ formula to provide the ground trust values of the randomly generated graphs. For each graph, the rank value is subtracted from the number of vertices. Section 3.2 clarifies the feature set and represents how to extract attributes. We have conducted two types of experiments, a prediction on the nullity values based on the collected dataset and another prediction on the nullity values of instances classified as YN by using the binary classifier. The binary classification clarifies whether a given graph has the nullity value (YN) or not (NN). Thus, while the nullity value of the graph is greater than zero, then the graph has a nullity value.

3.1. Random Graph generation

To provide a dataset including a large number of various types of graphs, we present the Random Graph Generator proposed in Algorithm 1. This algorithm accepts the number of vertices and a threshold as inputs to output an adjacent matrix. This algorithm is called as much as required to provide a significant and influential dataset. The threshold is a value between the range [0, 1] that specifies the generation type to be a sparse or dense graph. There are three steps; step 1 randomly sets the value of the lower triangular of the adjacent matrix. In step 2, the matrix's main diagonal is set to zero, and in Step 3, the values of the lower triangular are copied to the upper triangular. Thus, a symmetric matrix is achieved.

3.2. Feature Extraction

Features are attributes that represent the characteristics of a given graph. We introduced fourteen numerical attributes listed in Table 1 to predict the nullity value of the given graph.

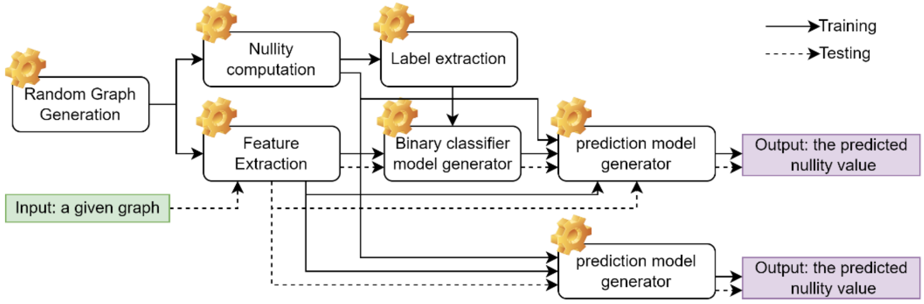
Training
Testing
Random Graph Generation
Nullity computation
Label extraction
Feature Extraction
Binary classifier model generator
prediction model generator
Output: the predicted nullity value
Input: a given graph
prediction model generator
Output: the predicted nullity value

Figure 1: Block diagram of the proposed method for nullity value prediction. Lines show the training phase process, and dash lines illustrate the testing phase process.

Table 1: Feature set.

	Symbol	Discerption
1	*n*	The number of columns in the matrix.
2	*nnz*	The number of nonzero entries in the matrix.
3	*avg_nnz*	The average value of the nonzero entries column.
4	*tril_nnz*	The number of nonzero entries in the lower triangular part
5	*nc*	The number of nonzero columns
6	*r*	The fraction of the number of matrix elements by the number of nonzero entries in the matrix.
7	*F*	The Frobenius norm of the matrix.
8	*lowband*	The lower bandwidth of the matrix.
9	*upband*	The upper bandwidth of the matrix.
10	*maxband*	The maximum bandwidth of the matrix.
11	*S*	The sparsity rate
12	*l*	The lower nonzero rate
13	*max_S*	The one norm (the maximum column summation)
14	*min_S*	The minimum column summation

As described in Table 1, F is the Frobenius norm of the matrix that it is the square root of the sum of the absolute squares of its elements.

$$A_F \equiv \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|a_{ij}\right|^2}$$

```
Algorithm 1: RandomGraphGenerator
  Data: $n$: the number of vertices, $\tau$ a given threshold
  Result: $A$: the adjacent matrix of a generated graph.
  begin
      //Step 1: Set the value of lower triangular.
      for row =1 to n do
          for col=(row+1): to n do
              $ranValue$ = randomInteger();
              if $ranValue \geq \tau$ then
                  $A(row, col) = 1$;
              else
                  $A(row, col) = 0$;
      //Step 2: Set the value on the main diagonal of the matrix to zero.
      for index =1 to n do
          A(index, index) = 0;
      //Step 3: Set the value on the upper triangular.
      for row =1 to n do
          for col=(row+1) to n do
              $A(row, col) = A(col, row)$;
      return $A$;
```

where n is the number of columns/rows in the matrix, and A is the adjacent matrix of a given graph. Also, $lowband$ is the upper bandwidth of a matrix that is defined as the largest value of $i-j$, where a_{ij} is nonzero. Moreover, $upband$ is the upper bandwidth of the matrix that is defined as the largest value of $j-i$, where a_{ij} is nonzero. In addition, $maxband$ is the maximum bandwidth of the matrix that is defined as the $max(max(i)-min(j))$, where a_{ij} is nonzero. Furthermore, the sparsity rate is the number of nonzero values in the matrix divided by the total number of elements in the matrix or $S=\dfrac{nnz}{n^2}$. Besides, the lower nonzero rate, named l, is the number of nonzero entries in the lower triangular part divided by the total number of elements in the matrix.

4. Evaluation

This section provides extensive details on the experiments conducted. The study investigated three different aspects (a) Time analysis, (b) prediction of the nullity value by various production methods, (c) classification, and again the prediction of the nullity value.

4.1. Dataset Preparation

We have extracted a large number of random graphs dataset with the number of vertices in the range [4, 1500]. The number of instances of the range [4, 50] is shown in Figure 2. In addition, there are 90 more instances for each value in the range [51, 1500]. Then, there are 226155 instances, including 202454 and 23701 instances with zero and greater than zero nullity values, respectively. Then, the dataset is imbalanced for binary classification.

4.2. Experimental Setup

We run our experiments on a PC with Lenovo Ideapad 3 laptop with Intel Core i7-CPU @ 3.3 GHz and 8GB of physical memory running Windows 11. We used Weka 3.8.6 [26] and MATLAB [27] version R2022a (9.12.0).

4.3. Evaluation Metrics

We used metrics, which are listed in Table 2, to estimate the performance of classification algorithms. The binary classification clarifies whether a given graph has the nullity value (YN) or not (NN). True-Positive (TP) denotes the number of true identifications of YN. True-Negative (TN) specifies the total number of NN identified appropriately. False-Positive (FP) specifies the number of misclassified YN, and finally, False-Negative (FN) designates wrongly identified NN.

Root mean squared error clarifies the difference between prediction and truth for each instance with the actual value of x_i estimated as $\hat{x}_i$. Also, N is the number of instances.

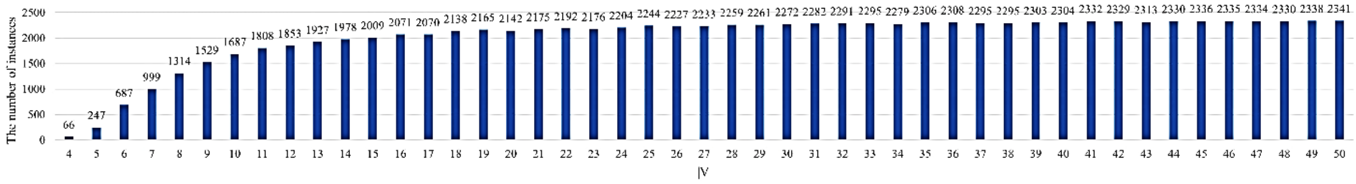

Figure 2: The number of samples in range [4-50].

Table 2: Performance metrics used in the experiments.

Metric	Formula
Accuracy	$\dfrac{TP+TN}{TP+FP+TN+FN}$
Precision	$\dfrac{TP}{TP+FP}$
Recall	$\dfrac{TP}{TP+FN}$
MCC	$\dfrac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$
F measure	$\dfrac{TP}{TP+\frac{1}{2}(FP+FN)}$
Root mean squared error	$\sqrt{\dfrac{\sum_{i=1}^{N}(x_i-\hat{x}_i)^2}{N}}$

4.4. Time Improvement

The time performance of our proposed machine learning-based method and the nullity computation formula are illustrated in Figure 3. Axes demonstrate the number of vertices and time that is the average time spent for nullity computation. As the number of vertices grows, the nullity calculation processing time increases. Thus, this figure clarifies the advantage of our proposed method for nullity computation based on machine learning.

4.5. Performance Analysis

We applied prediction and binary classification methods to compare the performance of machine learning for the collected dataset. Table 3 shows the performance of binary classifications. J48 provides better performance compared with other binary classifiers.

Next, we compared the prediction performance of those instances where their nullity value is greater than zero. Table 4 presents the prediction performance of prediction methods for the mentioned samples. The M5P tree has better performance and has a lower RMS error value.

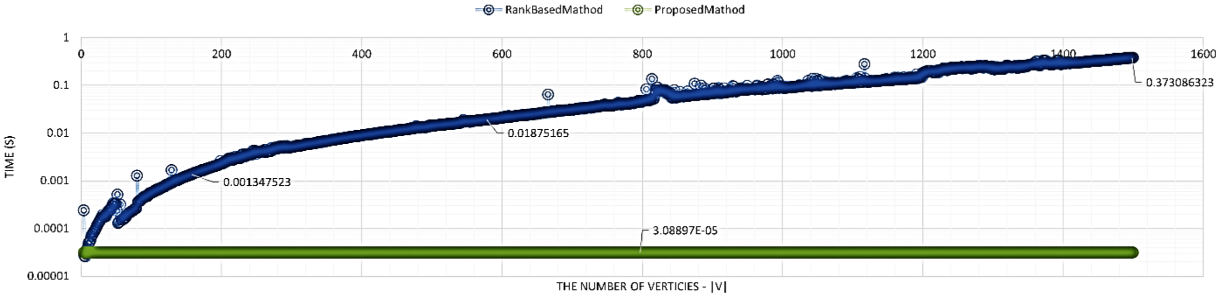

Figure 3: Time performance of the proposed method.

Table 3: Performance of binary classification.

Decision Tree	Correctly Classified Instance	Root Mean Squared Error	Precision	Recall	F measure	MCC	ROC Area
Naive Bayes	77.2793%	0.4742	0.925	0.773	0.815	0.474	0.965
Random Tree	95.6999%	0.2009	0.957	0.957	0.957	0.769	0.903
REP Tree	97.0436%	0.1429	0.970	0.970	0.970	0.837	0.986
Decision Stump	96.2787%	0.189	0.964	0.963	0.959	0.787	0.819
J48 -C 0.25	**97.0878%**	0.1443	0.970	0.971	0.970	0.841	0.981
IBk (KNN)	95.6322%	0.2021	0.956	0.956	0.956	0.765	0.899
Random Forest	96.2446%	0.1627	0.962	0.962	0.962	0.797	0.984

Table 4: Prediction performance for those samples that their nullity value is greater than zero.

Decision Tree	Correctly Classified Instance	Root Mean Squared Error	Relative Absolute Error
Decision Stump	0.7316	1.3449	64.1542%
Random Tree	0.7898	1.2806	51.2895%
REP Tree	0.8778	0.9459	42.4209%
Simple linear Regression	0.8329	1.0916	52.6387%
M5P Tree	**0.8917**	**0.8928**	**41.5148%**

The prediction performance for all dataset instances is shown in Table 5, which explains the advantage of the M5P Tree performance with a lower RMS error value compared with other methods.

5. Conclusion

Instead of computing the nullity value of graphs by leveraging the formula, for the fist time, we proposed fourteen features to explore the machine learning model with better performance and a lower computational time. We offer a random graph generation method to collect many graph instances to show the graph type is not a constraint. The features and the nullity values of graphs are extracted, and the machine learning model is achieved. The collected dataset is classified and predicted with some model generation algorithms. In comparison, the J48 algorithm, with an accuracy of 97.0878% and the M5P algorithm, with an accuracy of binary 94.56%, had better performances for binary classification and value prediction, respectively.

Table 5: Prediction performance.

Decision Tree	Correctly Classified Instance	Root Mean Squared Error	Relative Absolute Error
Decision Stump	0.8183	0.6068	50.9094%
Random Tree	0.8928	0.4894	25.3344%
REP Tree	0.9392	0.3626	20.9181%
Simple linear Regression	0.898	0.4644	36.72%
M5P Tree	**0.9456**	**0.3433**	**20.302%**

References

[1] Cvetkovic, D. M., *Applications of graph spectra: An introduction to the literature*, Application Graph Spectra, 13 (21), (2009), 7–31.

[2] Ahmad, Z., Mufti, Z. S., Nadeem, M. F., Shaker, H., and Siddiqui, H. M. A., *Theoretical study of energy, inertia and nullity of phenylene and anthracene*, Open Chemistry, 19, (1), (2021), 541–547.

[3] Chu M. T., *A note on the homotopy method for linear algebraic eigenvalue problems*, Linear Algebra and its Applications, 105, (1988), 225–236.

[4] Dhillon I. S., Parlett B. N., and Vomel C., *The design and implementation of the MRRR algorithm*, ACM Transactions on Mathematical Software (TOMS), 32 (4), (2006), 533–560.

[5] Wang L., *Nullity of a graph in terms of path cover number*, Linear and Multilinear Algebra, 69 (10), (2021), 1902–1908.

[6] Von Zur Gathen J., and Gerhard J., Modern computer algebra, Cambridge university press, 2013.

[7] Naresh R., and Sharma U., *Nullity of corona of a path with Smith graphs*, European Journal of Pure and Applied Mathematics, 10 (5), (2017), 1050–1057.

[8] Brouwer A. E., *Some simple graph spectra*, Eindhoven, (2021).

[9] Han D., and Zhang J., *A comparison of two algorithms for predicting the condition number*, in Sixth International Conference on Machine Learning and Applications (ICMLA 2007), (2007).

[10] Wang L., *Nullity of a graph in terms of path cover number*, Linear and Multilinear Algebra, 69 (10), (2021), 1902–1908.

[11] Xuezhong T., and Liu B., *On the nullity of unicyclic graphs*, Linear Algebra and its Applications, 408, (2005), 212–220.

[12] Hu S., Xuezhong T., and Liu B., *On the nullity of bicyclic graphs*, Linear Algebra and its Applications, 429, (2008), 1387–1391.

[13] Cheng B., and Liu B., *On the nullity of tricyclic graphs*, Linear algebra and its applications, 434, (2011), 1799–1810.

[14] Fan Y.-Z., and Qian K.-S., *On the nullity of bipartite graphs*, Linear algebra and its applications, 430, (2009), 2943–2949.

[15] Omidi G., *On the nullity of bipartite graphs*, Graphs and Combinatorics, 25 (1), (2009), 111–114.

[16] Cvetkovic D. M., and Gutman I. M., *The algebraic multiplicity of the number zero in the spectrum of a bipartite graph*, Matematicki vesnik, 9 (56), (1972), 141–150.

[17] Guo J.-M., Yan W., and Yeh Y.-N., *On the nullity and the matching number of unicyclic graphs*, Linear Algebra and its Applications, 431 (8), (2009), 1293–1301.

[18] Wang L., and Wong D., *Bounds for the matching number, the edge chromatic number and the independence number of a graph in terms of rank*, Discrete Applied Mathematics, 166, (2014), 276–281.

[19] Ma X., Wong D., and Tian F., *Nullity of a graph in terms of the dimension of cycle space and the number of pendant vertices*, Discrete Applied Mathematics, 215, (2016), 171–176.

[20] Ma H., and Liu X., *A Characterization of Graphs with Rank No More Than 5*, Applied Mathematics, 8 (1), (2017), 26–34.

[21] Cheng B., and Liu B., *On the nullity of graphs*, The Electronic Journal of Linear Algebra, 16, (2007), 60–67.

[22] Chang G. J., Huang L.-H., and Yeh H.-G., *A characterization of graphs with rank 4*, Linear Algebra and its Applications, 434 (8), (2011), 1793–1798.

[23] Chang G. J., Huang L.-H., and Yeh H.-G., *A characterization of graphs with rank 5*, Linear Algebra and its Applications, 436 (11), (2012), 4241–4250.

[24] Cvetkovic D. M., DOOB B., and SACHS H., *Spectra of graphs. Theory and application*, Academic Press, 87, (1980), 1–368.

[25] Song Y.-z., Song X.-q., and Zhang M., *An upper bound for the nullity of a bipartite graph in terms of its maximum degree*, Linear and Multilinear Algebra, 64 (6), (2016), 1107–1112.

[26] Witten I. H., Frank E., Hall M. A., and Pal C. J., Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, (2016).

[27] MATLAB, version 9.12.0 (R2022a), Natick, Massachusetts: The MathWorks Inc., (2022).