Int. J. Nonlinear Anal. Appl. In Press, 1–17
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.28936.4027

Correlation estimation between samples based on covariance, graph theory and graph neural network

Ebrahim Khalili, Razieh Malekhosseini*, S. Hadi Yaghoubian, Hamid Parvin, Karamollah Bagherifard

Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran

(Communicated by Saman Babaie-Kafaki)

Abstract

One of the standard criteria for expressing the relationship between two random variables is the correlation coefficient. Correlation between variables shows that changing the value of one variable leads to changing another variable in a certain direction. It is also possible to use the value of one variable to predict the value of another. In statistics, the correlation coefficient measures the direction and strength of the tendency to change. In machine learning, the correlation coefficient is known as a measure of classification quality. In fact, as a starting step for classification, the correlation between different samples should be estimated using a specific method. There are various methods to estimate the correlation of different data types, which have disadvantages such as low accuracy or high computational time. One of the methods that can overcome these problems, due to its high capability in modeling correlation between samples is graphical modeling. In this research, a new covariance model based on graph theory and graph neural network for estimating the correlation between samples is presented. The results show the improvement of the proposed model in accuracy, sensitivity, precision, F-Micro, F-Macro and statistical tests compared to Pearson and cosine methods.

Keywords: Correlation estimation, Covariance, Graph theory, GNN
2020 MSC: 05Cxx, 92B20

1 Introduction

In statistical multivariate analysis, there are different computational methods for measuring the dependence or relationship between two random variables. Correlation between two variables means the ability to predict the value of one in relation to the other [1]. For example, supply and demand are two interdependent phenomena. One way to show the relationship between the two variables is to calculate "covariance" and "correlation coefficient" between them. The larger value of these two indicators shows the greater relationship or dependence between the two variables. For example, there is a strong correlation between the two variables of power consumption and air temperature. With increasing temperature, the use of cooling devices also increases and increases power consumption. As a result, there is a wide correlation between the two variables. There are several types of correlation coefficients, each with its own definition, scope, and characteristics. The range of all of them is defined from -1 to +1. So that ± 1 represents the strongest possible agreement and 0 represents the strongest possible difference [28].

*Corresponding author
Email addresses: khalili.ebrahim.edu@gmail.com (Ebrahim Khalili), malekhoseini.r@gmail.com (Razieh Malekhosseini), yaghoobian.h@gmail.com (S. Hadi Yaghoubian), parvinhamid@gmail.com (Hamid Parvin), karam.bagherifard@gmail.com (Karamollah Bagherifarda)

Received: November 2022 *Accepted:* May 2023

Selecting a suitable criterion for calculating the similarity between samples has a great impact on the performance of the correlation estimation algorithm. There are different factors for calculating the similarity between samples such as the amount of variability in the data, differences in the shapes of the distributions, lack of linearity, the presence of one or more "outliers," characteristics of the sample, and measurement error, each of them has different results [11]. For this reason, it is especially important to select an appropriate criterion for calculating the similarity between samples.

The parameter used in this study to estimate the correlation is to calculate the distance between two samples in the problem space. Distance is a criterion for showing heterogeneity. It helps to move in the sample space and calculate the final correlation. Accordingly, first two samples will be defined as two vectors in the problem space and then the distance between them will be calculated. Now, if the distance criterion is provided for two feature vectors, the similarity between the two vectors (samples) can be calculated. After calculating the distance between two samples, their correlation is determined and they are placed in a category accordingly. Each distance measure introduced to calculate the distance between two samples in the problem space must be clear and have a series of properties. These features include:

1. The distance between both samples is greater or equal to zero.
2. The distance of each sample with itself is equal to zero.
3. The distance of sample x with y is equal to the distance of sample y with x.
4. The triangle theorem must be true of the distance between three properties. That is, the sample distance x to z plus the distance between z to y must be greater than the distance between x to y.

These axioms can be summarized as the following relations [4]:

$$d(x, y) > 0. \tag{1.1}$$
$$d(x, y) = 0 \quad \text{if} \quad x = y. \tag{1.2}$$
$$d(x, y) = d(y, x). \tag{1.3}$$
$$d(x, y) < d(x, z) + d(z, y) \tag{1.4}$$

To estimate the correlation between samples, there are various methods that have problems such as high time complexity, insufficient accuracy, and the impossibility of using different data types. To overcome these issues, we use a covariance model based on graph theory and graph neural network (GNN) to solve the problem of approximation of centrality criteria. GNN is a type of neural network architecture that uses graph structure and node/edge feature information to learn node or graph representation [22]. The general principle of GNNs is the node feature aggregation scheme along the edges of the graph. In a multilayer GNN model, each node aggregates the features of its neighbors along all paths that start or end at the node in question. By repeated aggregation, the resulting node representation acquires the structural information of its neighborhood. One of the important applications of graphs is to classify input data based on their structure. For example, in the field of software engineering, software is displayed as a program flow diagram, and the classification of diagrams is used to distinguish correct and defective software [9]. Relying on the ability of GNN to learn the graph structure, we propose a new selective feature aggregation scheme based on the covariance model and shortest paths in the graph. Reducing computational complexity, increasing performance factors such as accuracy, sensitivity, etc., and the ability to use different types of data are considered as assumptions of the proposed model.

This research is organized as follows: section 2 investigated advantages and disadvantages of various mathematical functions to calculating the distance. The proposed method explained in section 3. Then, evaluation and comparison results are shown in section 4. Finally, section 5 concluded the research

2 Related works

Classification is the taxonomy of structured or non-structured data sets into categories. In statistics, classification is the placement of new observations into a set of groups based on the previous data set trained. In machine learning, classification is mentioned as one of the supervised learning cases. In other words, learning in which well-defined training sets are available. The main purpose of classification is to determine in which category new data should be placed. The first step in the classification process is to estimate the correlation between the samples and the first step in the approximation of the correlation is to calculate the distance between the samples, which is defined as numerical vectors. Atomic vectors, which are the most important, have six types: logical, integer, binary, character, complex

and raw. Integer and binary vectors are collectively known as numeric vectors. Atomic vector was first presented by Kwan et al. [34].

There are various criteria for measuring the distance between numeric vectors, the most common and widely used of which is the Euclidean distance. In a general classification, distance measurement criteria can be divided into two general categories of Euclidean and non-Euclidean criteria. Euclidean criteria include Manhattan, Minkowski, City block and Chebyshev distance. Non-Euclidean distance criteria include Jaccard, Mahalanobis, Edit, cosine and Pearson distance. In the following, we will have an overview of these criteria.

2.1 Euclidean criteria

2.1.1 Manhattan Distance

The Manhattan distance between two points x = $(x_1, x_2, \ldots, x_n)$ and y = $(y_1, y_2, \ldots, y_n)$ in n-dimensional space is the sum of the distances in each dimension [3]. The formula is given in Equation (2.1)

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|. \tag{2.1}$$

2.1.2 Minkowski Distance

This criterion is calculated according to the formula 2.2 [3]:

$$d(S_i, S_j) = \left(\sum_{t=1}^{D} (S_{it} - S_{jt})^p \right)^{\frac{1}{p}}. \tag{2.2}$$

In formula (2.2), S_i and S_j are two properties in the D dimension space. This criterion is one of the most well-known and general criteria for calculating distance.

2.1.3 City Block Distance

In relation (2.2), when p = 1, It is called the city block. The city block distance is generally calculated between the coordinates of the two paired objects. This sum is the absolute difference between the two coordinates.

2.1.4 Chebyshev distance

Also, in formula (2.2), when p =∞, It is called Chebyshev distance [7]. In mathematics, Chebyshev distance (or chebychev), is a metric defined on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension [13].

2.2 Non-Euclidean criteria

2.2.1 Jaccard Distance

The Jaccard distance, commonly referred to as the Jaccard similarity coefficient, is defined by Paul Jaccard to calculate the distance between different samples. The Jaccard similarity coefficient can be defined as the size of the commonality of two samples on the community of the two. The formula for this definition is as follows [24]:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{2.3}$$

Formula (2.3) obtains the similarity between the two set. While to obtain the dissimilarity of two sets, the following relation is used.

$$J_\delta(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}. \tag{2.4}$$

2.2.2 Mahalanobis Distance

Mahalanobis distance criterion is a statistical criterion for calculating the degree of similarity between two different features. This measure was introduced in 1936 [32]. This criterion is mostly used to calculate the distance of a known sample from an unknown sample. This criterion is somewhat different from the Euclidean distance criterion, which considered data correlation and was a fixed-scale criterion.

Since the linear correlation between the samples can change the distance criterion to some extent, by using a transfer and applying this distance criterion, its effect can be reduced. This distance criterion can be summarized as follows

$$d(S_i, S_j) = (S_i - S_j)C^{-1}(S_i - S_j)^T, \tag{2.5}$$

where C^{-1} is the inverse covariance matrix of independent variables.

2.2.3 Edit Distance

This distance criterion is mostly used to calculate the distance between different strings. This string criterion shows the distance between two sequences or two words and states how many characters must be changed to convert one of the two sequences to another. This criterion, also known as the Levenshtein distance, was introduced in 1965 [19]. For example, the distance between the two strings "kitten" and "sitten" is one because by changing "s" to "k" these two sequences become one. Also, the distance between the two "kitten" and "sitting" sequences is 3. The formula for calculating this distance criterion can be expressed as follows.

$$d_{Lev}(S_i, S_j) = \left\{ \begin{array}{c} Max(S_i, S_j) \\ \min \left\{ \begin{array}{l} d_{Lev}(S_{i-1}, S_j) + 1 \\ d_{Lev}(S_i, S_{j-1}) + 1 \\ d_{Lev}(S_{i-1}, S_{j-1}) + [a_{S_i} \neq b_{S_j}] \end{array} \right\} \end{array} \right\} \tag{2.6}$$

where S_i, S_j are two strings to be compared.

2.2.4 Cosine similarity

Cosine similarity is introduced to calculate the proximity of two samples using the cosine of the angle between them. The cosine of a zero-degree angle is one, and every other angle has a cosine similarity of less than one. In fact, when the angle between two samples is zero, it is a case where the two samples are completely similar [17]. In this case, the similarity criterion has its maximum value. Also, two 90-degree angles have a cosine of zero. This means that when the vectors are perpendicular to each other, the similarity of the samples is zero. The formula for calculating cosine similarity can be shown as follows:

$$SimCos(S_i, S_j) = \frac{S_i.S_j}{||S_i|| \quad ||S_j||} = \sum_{t=1}^{D} \frac{(S_{it} \times S_{jt})}{\sqrt{\sum_{t=1}^{D}(S_{it})^2} \times \sqrt{\sum_{t=1}^{D}(S_{jt})^2}} \tag{2.7}$$

2.2.5 Pearson similarity

Pearson correlation coefficient between two samples F_i and F_j is calculated according to Equation (2.8) [18].

$$P_{ij} = \left| \frac{\sum_p (x_i - \overline{x}_i)(x_j - \overline{x}_j)}{\sqrt{\sum_p (x_i - \overline{x}_i)^2} \sqrt{\sum_p (x_j - \overline{x}_j)^2}} \right| \tag{2.8}$$

where x_i and x_j represent the vector samples of F_i and F_j, respectively. Also, the variables $\overline{x}_i$ and $\overline{x_j}$ represent the mean values for the vector x_i and x_j between p attributes, respectively. According to Equation (2.8), it can be understood that the amount of similarity between two samples that are completely similar is equal to one and the amount of similarity between two samples that are completely dissimilar is equal to zero [18].

In many cases it is possible that the calculated Pearson similarity values for the different properties are close to each other. To solve this problem, in order to normalize the calculated similarity values, the nonlinear scaling technique is used. Using this technique, all calculated similarity values are normalized in the range of zero to one. Normalization of similarity between features is done using Equation (2.9).

$$\hat{w}_{ij} = \frac{1}{1 + \exp(\frac{P_{ij} - \overline{S}}{\sigma})} \tag{2.9}$$

where P_{ij} is Pearson similarity between samples F_i and F_j, $\hat{w}$ and σ show the mean and standard deviation for all calculated similarities between all samples, respectively. In the following table, comparing the similarity between the two vectors is discussed and the advantages and disadvantages of these methods have been investigated. Table 1 shows the comparison between Euclidean and non-Euclidean criteria.

Table 1: Comparison between Euclidean and non-Euclidean criteria

Euclidean				
Criterion	**Structure space**	**Application example**	**Advantages**	**Disadvantages**
Manhattan	**Vector/Matrix**	**Regression analysis/ compressed sensing/ Frequency distribution/ Measures of distances in chess**	**Usability in high-dimensional data/ High speed**	**All data must be available**
Minkowski	**vector**	**Fuzzy Clustering [33]/ Measuring Quality of Service Infrastructure for Mobile Ad Hoc Networks [2]**	**High accuracy/ Flexibility and generality [8]**	**Need to normalizing the continuous features**
City block	**Vector**	**Calculate the distance between two data points in a grid-like path**	**Fast and low complexity**	**Not compatible with many standard multivariate analyses**
Chebyshev	**Vector**	**Chess/ Warehouse logistics**	**Easy implementation**	**Only usable in vector feature**

Non- Euclidean				
Criterion	**Structure**	**Application**	**Advantages**	**Disadvantages**
Jaccard	**finite sample sets**	**Duplicates detection**	**High speed/ Usability in continuous and categorical variables [12]**	**Only usable in collections**
Mahalanobis	**Vector/Matrix**	**For detecting outliers during calibration or prediction, or for detecting extrapolation of the model during analyses [21]**	**Mahalanobis is a data-driven measure that can ease the distance distortion caused by a linear combination of attributes**	**Complex implementation and It can be expensive in terms of computation [8]**
Edit	**String**	**Correction of spelling mistakes or OCR errors**	**Fast and simple**	**Only usable in string mode**
cosine	**Vector/Matrix**	**Measure document similarity in text analysis [12]**	**Usability in data with null values**	**High computational complexity/ Insufficiency in nominal data [12]**
Pearson	**Vector/Matrix**	**Computes the similarity of two lists of numbers**	**High accuracy/ Usability in large scale data**	**High computational complexity/ Sensitive to outliers [12]**

Some of the most important researches conducted in line with the method of estimating the degree of closeness are as follows. van der Grinten et al. first proposed a different approximation algorithm [30]. This method is up to two times faster and more accurate in practice. They take advantage of the strong correlation between uniformly spanning trees and forest distances by adapting and extending recent approximation algorithms for related single-vertex problems. This leads to an almost linear time algorithm with an absolute probable error guarantee. Investigations show that in cut graphs, group forest closeness performs better than existing centrality criteria in the context of semi-supervised

vertex classification. Saxena and co-workers in an article propose a heuristic method to quickly estimate the proximity rank of a node in O (α.m) time complexity [27]. They also propose an improved method developed using a uniform sampling technique. This method estimates the rank better and its time complexity is O(α.m). Borrego et al. provide an approach based on machine learning to define two models based on linear and polynomial regression to estimate the future values of node centrality [5]. Node centrality estimation is then used in a messaging technique called "Linear and Polynomial Regression Based" (LAPSE). Using simulations and through the use of real mobility traces, they show that the selection of forwarding nodes through estimated centrality values allows to obtain better performance than traditional approaches based on the overall centrality of the selected node. Jin et al. state in their research that they use forest distance to evaluate the importance of nodes in a graph, whether connected or disconnected [16]. For a node in a graph, its forest distance is defined as the sum of forest distances from the node to all other nodes in the graph. To demonstrate the discriminating power of forest distance, we first calculate the exact forest distances for all nodes in the path graph and show that the order of importance of nodes with forest distance is in perfect agreement with intuition. Then it shows that forest distance centrality has better discriminating power than alternative measures such as betweens, harmonic centrality, eigenvector centrality and page rank. Inariba et al. focused on a family of centrality measures, including harmonic centrality and its variants, and addressed their computational problem in very large graphs by presenting a new estimation algorithm called the Random Radius Ball (RRB) method [15]. The RRB method is easy to implement, and a theoretical analysis, including time complexity and error bounds, is also presented. The effectiveness of the RRB method over existing algorithms has been demonstrated through experiments on real-world networks. In the next section, we present the proposed model.

3 Proposed method

Similarity measurement is an instance of supervised machine learning in artificial intelligence, which is closely related to distance, regression, and classification measures, but the purpose of the similarity function is to measure the similarity of two samples and check which samples in the set are important and how they affect the overall data structure.

Betweenness centrality and closeness centrality are two examples ranking criteria that are usually used to find influential examples in graphs in terms of information dissemination and connectivity. Both of these are considered as shortest-path-based metrics because the calculations require the assumption that information flows between nodes via shortest paths. However, the exact computation of these centrality measures is computationally expensive and prohibitive, especially for large graphs.

There are several methods for calculating similarity that have issues such as low accuracy or high computational complexity. For example, one of the problems with the cosine similarity method is that calculating similarity in this method requires vector multiplication between all features, which increases the complexity of calculating similarity. Pearson method are also very costly in collections with a large number of samples. In fact, when the number of samples in our data set is large, to calculate the similarity between two samples, it is necessary to calculate the difference between the mean of each sample and the characteristics of that sample alone, that process is very time consuming and not applicable at an acceptable time.

To solve this problem, this research offers a new criterion that is both more accurate and less computationally complex. In fact, in this similarity criterion, an attempt has been made to improve the problem of high computational complexity in the cosine similarity coefficient resulting from the multiplication of properties and the problem of calculating the mean difference and samples of each property in the Pearson similarity coefficient. So, in this study, a new criterion based on the covariance vector is presented. This equation is introduced as follows:

$$Sim(S_i, S_j) = \frac{Cov(S_i)Cov(S_j)}{\sqrt{Var(S_i)Var(S_j)}}. \tag{3.1}$$

In the equation (3.1), Cov(S_i) represents the covariance of the sample vector x_i and also Var(S_i) represents the calculation of variance for the sample vector x_i. As it is known in this equation, if two samples are exactly the same, in this case the degree of similarity is equal to 1 or -1, and if two samples that are completely independent of each other, their degree of similarity will be equal to zero. Given that the value of this criterion will always be equal to 1 or -1 in the highest similarity and zero in the lowest similarity [23]. So contrary to Pearson's criterion, it does not need to be normalized and only by taking the absolute value of the similarity obtained, the weight of the corresponding edge is specified in the graph. Due to the omission of the normalization step, the proposed similarity calculation method will be much more efficient than the previous methods.

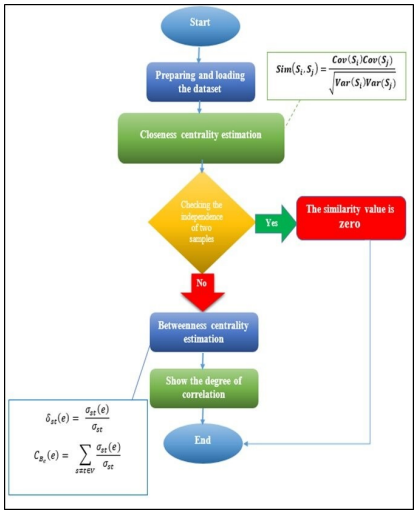

Figure 1: Flowchart of the proposed method

After calculating the closeness centrality of the samples, the betweenness centrality of the samples is estimated using the graph neural network. The concept of centrality is defined for the edges of a graph, and from a historical point of view, the first approach to calculating the center of the edge was proposed in 1971 by Antonius. In this approach, the centrality of the edge is interpreted as the flow of centrality. To define, let us consider a graph G = (V, E) and let s,t ∈ V be a fixed pair of nodes. The Rush index [20] with pair (s, t) and edge e ∈ E is defined as:

$$\delta_{st}(e) = \frac{\sigma_{st}(e)}{\sigma_{st}} \tag{3.2}$$

where σ_{st} is the number of shortest connection paths from s to t and $\sigma_{st}(e)$ is the number of shortest connection paths from s to t through the edge 'e'. If there is no way to join from S to t, then σ_{st} (e) = 0. The Rush index for edge e is defined in the range 0 (if 'e' does not belong to any of the shortest joining paths S and t) to 1 (if 'e' belongs to all the shortest connecting paths S and t). Thus, the highest value of σ_{st} is related to the contribution of e in the transfer of a current unit from S to t.

In 2002, Grivan and Newman [10] proposed a definition of the intermediate center of the edge that is very similar to the definition proposed by Antonius but differs from Antonius' theory because the source node s and the target node t must be different. Various margins of intermediate centrality have been proposed by Brands including intermediate edge and group, and central tension and load [6]. According to the symbol introduced above, the intermediate center of the edge e ∈ E is defined as follows:

$$C_{B_e}(e) = \sum_{s \neq t \in V} \frac{\sigma_{st}(e)}{\sigma_{st}}. \tag{3.3}$$

Figure 1 shows the flowchart of the proposed model.

As shown in Figure 1, the process of preparing and loading the dataset is done first. The dataset preparation process can include a series of operations such as removing null values, removing duplicates, etc. In the next step, first, the closeness centrality of the samples is calculated using the new formula based on covariance (Eq. (3.1)). If the output of this relationship is equal to zero, it means that the samples are completely independent and there is no similarity or correlation between them. So in this case, the operation is terminated and we enter the "end" phase. Otherwise (the presence of similarity between the samples), it is time to estimate the betweenness centrality using the graph neural network. Betweens centrality is a measure of the centrality of an edge in a network based on the number of shortest paths passing through it. Thus, it identifies edges in the network that are critical for information flow. Therefore, this step is very important. The pseudo-code of the Closeness centrality estimation stage named CCE algorithm is as follows: Based on the pseudo-code above, first the correlation coefficient of the input matrix is calculated. This value is then placed in W. Then, using the equation of line 3, the auxiliary vector called p is calculated, which is based on the absolute value of each W column. In the next step, the value of W is in column i and is calculated based on the value of columns Xi and pi. Finally, the highest value of W is calculated as the initial correlation matrix called WIC. In the next step, it is checked that if the WIC matrix values are less than

Algorithm 1. Closeness Centrality Estimation

Input: Samples matrix X, Flag
Output: Final Correlation Matrix W_{FC}
1. Initialize: W = 0.
2. do
3. $p_i = \frac{1}{2*||W(i)||}$ // Auxiliary vector p of Matrix W Initial Correlation [m1, n1]
4. $W_{IC} = Inverse(X^T * X + \lambda * p_i * I) * X^T * X_i$ // correlation matrix W
5. After process:
for i=1 to m1
for j=1 to n1
If $W_{IC}(i,j) < 0$
$W_{IC}(i,j) = 0$
else
W=W_{IC}
6. While converged do
7. N ← size(W)
8. B ← Repeat Matrix (mean(W), n,1)
9. W_{center}=W-B
10. C = $(W_{center})^T * (\frac{W_{center}}{n})$ // Covariance Matrix
11. end while

zero, it is replaced with zero value. Otherwise, the values themselves remain in effect. In the last step of Algorithm 3, the operation of calculating the covariance matrix is performed. This is the output matrix of Algorithm 3. The pseudo-code of the betweenness centrality estimation stage named BCE algorithm is as follows:

Algorithm 2. Betweenness Centrality Estimation

Input: G: graph. A NetworkX graph
Output: edges: dictionary. Dictionary of edges with betweenness centrality as the value.
1. Initialize: K: int, optional (default=None)
If k is not None use k node samples to estimate betweenness. The value of k $\leq$ n
where n is the number of nodes in the graph. Higher values give better approximation.
2. Normalized: bool, optional
If True the betweenness values are normalized by $\frac{2}{n(n-1)}$
for graphs, and $\frac{1}{n(n-1)}$ for directed graphs where n is the number of nodes in G
3. Weight: None or string, optional (default=None)
If None, all edge weights are considered equal. Otherwise holds the name of
the edge attribute used as weight. Weights are used to calculate weighted shortest paths, so they
are interpreted as distances.
4. Seed: integer, random-state, or None (default)
Indicator of random number generation state. See Randomness. Note that this
is only used if k is not None.

4 Evaluation

In this section, first, the datasets and evaluation metrics are introduced. Then the results and discussion are presented.

4.1 Dataset

The data sets that used are:

4.1.1 20-Newsgroups [26]

This dataset consists of 20 news groups, each of them consists of approximately 1,000 news items. The specifications of this dataset are as shown in Table 3. To create an incremental collection, the dataset divided into 20 incremental categories each training category containing 480 training data except for the last category, which includes 478 training

Table 3: Features of 20-Newsgroups dataset

#	Newsgroup Name	Number of samples	#	Newsgroup Name	Number of samples
1	Alt.atheism	1000	11	Rec.sport.hockey	1000
2	Comp.graphics	1000	12	Sci.crypt	1000
3	Comp.os.ms-windows.misc	1000	13	Sci.electronics	1000
4	Comp.sys.ibm.pc.hardware	1000	14	Sci.med	1000
5	Comp.sys.mac.hardware	1000	15	Sci.space	1000
6	Comp.windows.x	1000	16	Soc.religion.christian	997
7	Misc.forsale	1000	17	Talk.politics.guns	1000
8	Rec.autos	1000	18	Talk.politics.mideast	1000
9	Rec.motorcycles	1000	19	Talk.politics.misc	1000
10	Rec.sport.baseball	1000	20	Talk.religion.misc	1000

data, which there are about 24 training data from each class. For each training category, there is a validation class of 220 cases (There are 11 data from each class) except for the last category, which is 219, and for the final test, there is a set of 6001 test data.

4.1.2 Web KB [29]

The texts in the Web KB (Web Knowledge Base) dataset are web pages of the Internet provided by a group at CMU University. This data set was collected from the Department of Computer Science of various universities in 1997. The specifications of this data set are as shown in Table 4. To create an incremental set, this dataset is divided

Table 4: Web KB Database Specifications

#	Newsgroup Name	Number of samples
1	Course	930
2	Faculty	1124
3	Project	504
4	Student	1641

into 16 training categories, each containing 128 data, so that there are all four categories in this subdivision except the last category, which contains 112 data. Also, for each training group, a validation group of 57 is considered, which includes all classes except the last set, which contains 50 data. For the final test, the set contains 1262 test data.

4.1.3 Image Net [25]

It is a large-scale data set organized according to the WorldNet hierarchy and each node is represented by hundreds and thousands of images. The Image Net dataset has 1000 classes. In experiments, a subset of Image Net that includes 200,000 images for training and 100,000 images for testing is used. To create an incremental set, it divided into 7 training categories each contains 30,000 cases. There are cases from each data class in these sets, except for the last set which contains 20,000 data. For each training group, a set of 15,000 from each data class in these sets and for the final test, a set of 10,000 test data is considered.

In all experiments, 66% of the data were considered as training data and 34% as testing data. In the training data set, 50% of the data were considered as labeled data and remain as unlabeled data. In other word, 33% of the data is labeled during training and 33% of the unlabeled data is used for training. Also, 34% of the data were used to test and evaluate the proposed method. In all datasets, work starts with training data and the data is called based on the specified strategy and the desired strategy will be performed depending on whether the data is labeled or not.

4.2 Evaluation metrics

To evaluate the proposed method, compare it with other methods and show the improvement, two categories of evaluation, one micro and macro criteria to classification quality assessment and the other, Friedman statistical test to investigate the relationship between the hypotheses and the data set and also for a repeated measures type of experiment to determine if a particular factor has an effect or not, have been used.

4.2.1 Micro metrics

Micro metrics assign the same weight to all texts, regardless of the number of classes to which they belong. These metrics are calculated according to formulas (19) to (21) from Table 5. In the above relations TP_i is equal to the

Table 5: Micro metrics

Metric	Formula	
$\text{Precision}_{\text{Micro}}$	$\text{Precision}_{\text{Micro}} = \sum_{i=1}^{\|c\|} \frac{TP_i}{\sum_{i=1}^{\|c\|}(TP_i+FP_i)}$	(19)
$\text{Recall}_{\text{Micro}}$	$\text{Recall}_{\text{Micro}} = \sum_{i=1}^{\|c\|} \frac{TP_i}{\sum_{i=1}^{\|c\|}(TP_i+FN_i)}$	(20)
$F1_{\text{Micro}}$	$F1_{Micro} = \frac{2*\text{Precision}_{\text{Micro}}*\text{recall}_{\text{Micro}}}{\text{Precision}_{\text{Micro}}+\text{Recall}_{\text{Micro}}}$	(21)

number of texts that are correctly categorized in category c_i, FP_i is equal to the number of texts incorrectly classified in category c_i, FN_i is equal to the number of texts incorrectly categorized in other category and TN_i equals the number of texts that are correctly placed in other categories.

4.2.2 Macro metrics

Macro metrics assign the same weight to all classes, regardless of how much text belongs to them. These metrics are calculated according to formulas (22) to (28) from Table 7.

Table 6: Please write your table caption here

Metric	Formula	
Precision_i	$\text{Precision}_i = \frac{TP_i}{TP_i+FP_i}$	(22)
Recall_i	$\text{Recall}_i = \frac{TP_i}{TP_i+FN_i}$	(23)
$F1_i$	$F1_i = \frac{2*\text{Precision}_i*recall_i}{\text{Precision}_i+\text{Recall}_i}$	(24)
$\text{Precision}_{\text{Macro}}$	$\text{Precision}_{\text{Macro}} = \sum_{i=1}^{\|c\|} \frac{\text{Precision}_i}{\|c\|}$	(25)
$\text{Recall}_{\text{Macro}}$	$\text{Recall}_{\text{Macro}} = \sum_{i=1}^{\|c\|} \frac{\text{Recall}_i}{\|c\|}$	(26)
$F1_{\text{Macro}}$	$F1_{\text{Macro}} = \frac{2*\text{Precision}_{\text{Macro}}*\text{recall}_{\text{Macro}}}{\text{Precision}_{\text{Macro}}+\text{Recall}_{\text{Macro}}}$	(27)
Accuracy_i	$\text{Accuracy}_i = \frac{TP_i+TN_i}{TP_i+FN_i+TN_i+FP_i}$	(28)

4.2.3 Friedman test

In the last part of the evaluation of the proposed method, we examine it using the Friedman test. The Friedman test is a non-parametric statistical test [14]. This test, known as the two-way analysis of variance test, is one of the statistical tests used to compare several groups and to determine the average rank of groups, whether these groups

can be from one community or not? Friedman's test determines whether the rank totals for each condition differ significantly from the values which would be expected by chance. The Friedman test formula is as follows:

$$M = \frac{12}{nK(K+1)} \sum_{k=1}^{K} R_k^2 - 3n(K+1)(29) \tag{4.1}$$

where
K=number of columns (treatments)
n= number of rows (blocks)
R= sum of ranks.

Under the null hypothesis, as n tends to infinity, this statistic M has an asymptotic Chi-square distribution with $K-1$ degrees of freedom.

4.2.4 Execution time

Training time is the time taken by a model to train on a dataset, and the execution time represents the total time taken for computations, including data splitting, data preprocessing, and model evaluation.

4.2.5 Results and discussion

In this section, the performance of different methods for estimating the correlation between samples is examined. In this research, to simulate the proposed method and compare it with other methods, MATLAB software has been used because with this software, datasets can be defined in the form of matrices and the relationships between them can be well modeled. In this research, MATLAB version 2021 as been used for programming. Also, the hardware specifications of the computer used are given in Table 7. Tables 8 to 10 show the superiority of the proposed method in

Table 7: Hardware specifications of simulation system

Specification	Hardware
CPU	Intel Ci7, 12 Cores, 15 Meg Cache
RAM	16 Giga Byte DDR4
H.D.D	1T.B

comparison to different correlation estimation techniques. As the tables show, in all metrics, the proposed covariance-based correlation estimation method has the best performance.

Table 8: Comparison of correlation estimation methods in 20 Newsgroup datasets

Method	**F1-Macro(%)**	**F1-Micro(%)**	**Precision (%)**	**Recall (%)**	**Accuracy (%)**	**Execution time(s)**
Pearson	52.67	81.97	91.98	.72	.39	1346
Cosine	49.37	84.13	86.43	34.56	41.76	1253
Proposed method	89.00	96.72	95.14	84.37	85.51	1151

Despite the high diversity of data samples in the 20 Newsgroup dataset and the need for accurate classification, the proposed method, due to its reliance on the covariance index, succeeded in better classification in comparison to the Pearson and Cosine and methods. In this dataset, the highest rate of increase is related to F1-Macro. In comparison to Pearson and Cosines methods, our method has 36.33% and 39.63% increase, respectively. Considering that the nature of the dataset is text, the proposed method has been able to use the useful feature of covariance in examining the relationship and dependence between texts, and on the other hand, the success of covariance in reducing data dimensions and extracting effective features has shown improvement in this dataset. Also, the implementation time of the proposed method is less compared to the other two methods. The different nature of data (such as text, images,

Table 9: Comparison of correlation estimation methods in Web KB dataset

Method	F1-Macro(%)	F1-Micro(%)	Precision (%)	Recall (%)	Accuracy (%)	Execution time(s)
Pearson	58.87	82.62	92.38	56.71	57.42	413
Cosine	51.47	84.13	82.15	53.52	53.61	316
Proposed method	84.32	92.71	93.65	81.61	79.65	283

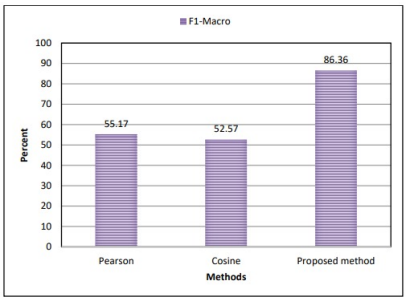

Figure 2: Comparison of correlation estimation methods in F1-Macro factor

videos, etc.) in the WebKB dataset, and their conversion to matrices, complicates data separation and classification operations. The proposed method in this diverse data set, in addition to the F1-Macro metric, was able to show better performance in terms of Recall and Accuracy in comparison to the other two methods. In this data set, due to the large volume of data dimensions and the existence of more random variables, the proposed method based on covariance has succeeded in reducing the data dimensions by establishing the relationship between random variables and finally extracting useful features. Modeling the important features and establishing the relationship between them has led to accurate classification of data and ultimately increased accuracy. Due to the smaller number of data in this dataset compared to the 20 newsgroup dataset, we see a shorter execution time. But in this classification as well, the proposed method has taken less execution time. In the ImageNet database, due to the large number and variety of images,

Table 10: Comparison of correlation estimation methods in Image Net dataset

Method	F1-Macro(%)	F1-Micro(%)	Precision (%)	Recall (%)	Accuracy (%)	Execution time(s)
Pearson	53.97	80.78	82.62	63.41	64.98	5366
Cosine	56.87	83.62	81.64	55.72	56.83	6211
Proposed method	85.78	90.84	92.09	82.58	83.19	4526

we also face the problem of computational complexity and long execution time. In such cases that the number of samples is large, to calculate the similarity between the two samples, the difference between the mean of each sample and the characteristics of the dataset must be calculated. This process complicates the computation and increases execution time. As mentioned earlier, the proposed method has less computational complexity due to the lack of need for normalization in the edge weight determination step in the graph. In addition to this advantage, in the ImageNet dataset, the proposed method was able to show significant growth in all metrics. The highest increase occurred in F1-Macro, Recall and Accuracy metrics in comparison to Pearson and Cosine methods, respectively. The average performance of the proposed method in different data sets in comparison to other methods in terms of F1-macro, F1-micro, Precision, Recall and Accuracy factors are shown in Figures 2 to 6, respectively.

From the examination and analysis of Figure 2, it is clear that the proposed method has grown by 56% and 64%, respectively, in the F1-Macro factor compared to the Pearson and Cosine methods. The reason for the difference is that the covariance-based method has been successful in successive measurements of a value, and checking how close the measured values are to each other. Also, according to Figure 3, we see a growth of 14 and 11 percent in the

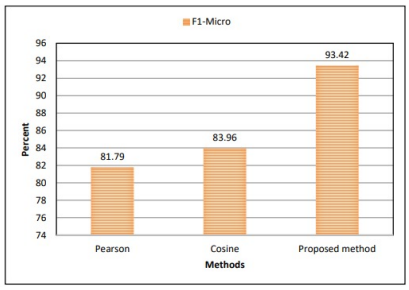

Figure 3: Comparison of correlation estimation methods in F1-micro factor

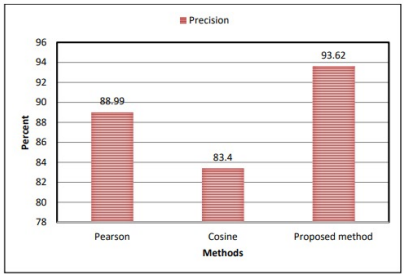

Figure 4: Comparison of correlation estimation methods in Precision factor

F1-Micro factor of the proposed method compared to the other two methods. The presence of multiple labels in the mentioned data set and on the other hand the ability of the proposed method in extracting useful features is the reason for the success of this method.

Precision is a description of random errors and a measure of statistical variability. In simpler terms, in a data set, with repeated measurements of a value, the set can be said to be accurate if their mean is close to the true value of the measured value. In terms of Precision factor, the proposed method has provided 0.05 and 12% growth compared to Pearson and Cosine methods. The reason for this growth is repeated measurements using modeling and data analysis by covariance.

According to Figure 5, the method presented in the Recall factor has been faced with an increase of 42 and 72 percent compared to the Pearson and Cosine methods. The reason for the increase of this index is the capability of the proposed method in the number of data detection and their correct classification by using the covariance capability in the detailed examination of the data and reducing their dimensions. In the Accuracy factor, the proposed method has provided an increase of 39 and 63 percent compared to the Pearson and Cosine methods. In a set of data measurements, the accuracy of the measurements is considered to be close to a certain value. The proposed method has been able to reduce the amount of features by using the covariance feature and select a more effective and useful number for classification. Reducing the number of features causes less confusion in the machine learning pattern and better classification.

Friedman's non-parametric hypothesis test is used to investigate the difference between groups (three or more paired

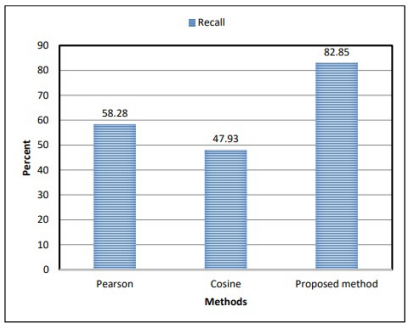

Figure 5: Comparison of correlation estimation methods in Recall factor

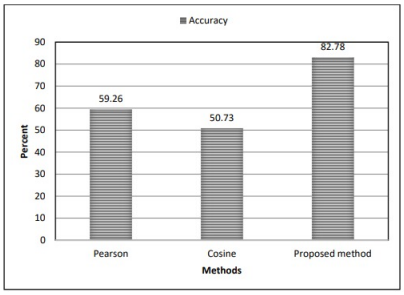

Figure 6: Comparison of correlation estimation methods in Accuracy factor

groups) when the dependent variable is at least ordinal. This test is preferred in comparison to other non-parametric tests in situations where the same parameter is measured in different conditions in the same subject. The Friedman test is similar to the Kruskal-Wallis test and also an extension of the sign test. This test is the best statistic that is used to test repeated measurements to determine whether a certain factor has an effect on the data classification or not. We applied this test to prove the efficiency and success rate of the proposed method.

The results of Friedman test on Pearson, cosine and the proposed method are shown in Tables 11 to 13, respectively.

Table 11: Friedman test results on Pearson method

Test Statistics	
N	3
Chi-Square	10.400
df	4
Asymp. Sig.	.034

Table 12: Friedman test results on Cosine method

Test Statistics	
N	3
Chi-Square	9.867
df	4
Asymp. Sig.	.043

In each of the above three tables, the value N represents the number of datasets. The Friedman test was applied to all three datasets for each of the Pearson, Cosine, and proposed methods. Also, all F-Macro, F-Micro, Precision, Recall and Accuracy metrics are considered in this test. A positive result from a chi-squared test indicates that there is some kind of relationship between variables but we do not know what sort of relationship it is. We need to use summary statistics to discuss what the relationship is. There is significant evidence of an association in proposed method in comparison with other, ($\chi 2$= 10.933, p ¡ 0.05). Statistical significance is often referred to as the p-value (short for "probability value") or simply p in research papers. A small p-value basically means that your data are unlikely under some null hypothesis. A somewhat arbitrary convention is to reject the null hypothesis if p ¡ 0.05. A lower p-value indicates more confidence in the relationship between the samples. In other words, less Asymp. Sig. (p-value) in the proposed method (0.027) in comparison with the values of the other two methods, shows more confidence in the reality of the observed relationships between the samples.

Comparing Tables 11 to 13, it is clear that the proposed method has 5% and 10% improvement in Chi-Square factor in comparison to Pearson and Cosine methods, respectively. Also, the proposed method in the Asymp. Sig. achieved 25% and 59% growth in comparison to Pearson and Cosine methods.

Table 13: Friedman test results on proposed method

Test Statistics	
N	3
Chi-Square	10.933
df	4
Asymp. Sig.	.027

As the proposed method was more successful in comparison to two other methods in terms of macro and micro metrics, in Friedman's statistical test also shows a better performance. The reason for the result of Friedman test is the better performance of the proposed method in the process of detecting and estimating the correlation between data sets. In fact, because the proposed method was more successful in dissociation, correlation estimation, and finally classification, the result of the Friedman test, which showed a significant difference between the classifications of the data set under study, was more acceptable than other methods.

5 Conclusion

In today's world, the rapid growth of technology and its subsequent production of large amounts of different data is not hidden from anyone. Therefore, in order to separate and categorize diverse data based on their specific and different applications, the need for correlation estimation methods is felt. There are different ways to do this, each with its own advantages and disadvantages. Computational complexity and execution time are usually the most important criteria for comparing the efficiency of methods. In this study, a new model based on covariance, graph theory and graph neural network is presented to estimate the correlation between samples.

The purpose of this method is to investigate the direct similarity between the samples. In our approach, the sample space is represented as a graph, each sample forming a node of the graph and a function that examines the degree of similarity between the two samples. The simulation results show that the proposed method has both high accuracy and less computational complexity in comparison to Pearson and Cosine methods.

In Table 14, the percentage of improvement obtained by the proposed method in different factors compared to the Pearson and Cosine methods is displayed.

Table 14: The percentage of improvement achieved by the proposed method in different factors

Compared to the method	Accuracy	Recall	Precision	F1-Micro	F1-Macro
Pearson	39%	24%	0.05%	14%	56%
Cosine	63%	72%	12%	11%	44%

References

[1] H. Akoglu, *User's guide to correlation coefficients*, Turk. J. Emergency Med. **18** (2018), no. 3, 91–93.

[2] Y. Al-Sbou, *Minkowski distance as a quality of service assessment tool*, Preprint.

[3] R.C. Amorim and B. Mirkin, *Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering*, Pattern Recog. **45** (2012), no. 3, 1061–1075.

[4] A.V. Arkhangel'skii and L.S. Pontryagin, *General Topology I: Basic Concepts and Constructions Dimension Theory*, Encyclopaedia of Mathematical Sciences, 1990.

[5] C. Borrego, E. Hemández-Orallo, P. Manzoni, and A.M. Vegni, *LAPSE: A machine learning message forwarding approach based on node centrality estimation in sparse dynamic networks*, Wireless Days (WD). IEEE, 2021, pp. 1–6.

[6] U. Brandes, *On variants of shortest-path between centrality and their generic computation*, Soc. Networks **30** (2008), no. 2, 136–145.

[7] H.B. Colakoglu, *A generalization of the Minkowski distance and a new definition of the ellipse*, Turk. J. Math. 44 (2020), no. 1, 319–333.

[8] M.C. Delfour, *Topological derivative: A semidifferential via the Minkowski content*, J. Convex Anal. **25** (2018), no. 3, 957–982.

[9] F. Errica, M. Podda, D. Bacciu, and A. Micheli, *A fair comparison of graph neural networks for graph classification*, arXiv preprint arXiv:1912.09893 (2019).

[10] M. Girvan and M.E. Newman, *Community structure in social and biological networks*, Proc. Nat. Acad. Sci. **99** (2002), no. 12, 7821-7826.

[11] L. Goodwin, D. Leech, and L. Nancy, *Understanding Correlation: Factors that Affect the Size of r*, J. Exper. Educ. **74** (2006), no. 3, 251–266.

[12] M. Goswami, A. Babu, and B.S Purkayastha, *A comparative analysis of similarity measures to find coherent documents*, Int. J. Manag. **8** (2018), no. 11, 2249-7455.

[13] S. Gultom, S. Sriadhi, M. Martiano, and J. Simarmata, *Comparison analysis of K-means and K-Medoid with Ecluidience Distance Algorithm, Canberra Distance, and Chebyshev Distance for big data clustering*, IOP Conf. Ser.: Mater. Sci. Engin., vol. 420, 2nd Nommensen International Conference on Technology and Engineering, 2018, pp. 19–20.

[14] M. Hanafy and R. Ming, *Classification of the insureds using integrated machine learning algorithms: A comparative study*, Appl. Artific. Intell. **36** (2022), no. 1, 2020489.

[15] W. Inariba, T. Akiba, and Y. Yoshida, *Random-radius ball method for estimating Closeness centrality*, Proc. AAAI Conf. Artific. Intell., 2017.

[16] Y. Jin, Q. Bao, and Z. Zhang, *Forest distance closeness centrality in disconnected graphs*, IEEE Int. Conf. Data Min. (ICDM), 2019, pp. 339–348.

[17] H. Kalhori, M.M. Alamdari, and L. Ye, *Automated algorithm for impact force identification using cosine similarity searching*, Measurement **122** (2018), 648-657.

[18] Kent State University, *SPSS Tutorials: Pearson Correlation*, Available: https://libguides.library.kent.edu/SPSS/PearsonCorr.

[19] J.M. List, *Beyond edit distances: Comparing linguistic reconstruction systems*, Theor. Linguist. **45** (2019), no. 3-4, 247-258.

[20] C. Liu, F. Zhu, X. Chang, X. Liang, Z. Ge, and Yi-Dong Shen, *Vision-language navigation with random environmental mixup*, Proc. IEEE/CVF Int. Conf. Comput. Vision, 2021, pp. 1644–1654.

[21] H. Mark and J. WorkmanJr, *Chemometrics in Spectroscopy*, Second Edition, Elsevier, 2018.

[22] S.K. Maurya and X. Liu, *Tsuyoshi Murata, graph neural networks for fast node ranking approximation*, ACM Trans. Knowledge Discov. Data **1** (2021), 1—32.

[23] R. Pascual-Marqui, D. Lehmann, K. Kochi, T. Kinoshita, and N. Yamada, *A measure of association between vectors based on similarity covariance*, arXiv preprint arXiv:1301.4291 (2013).

[24] M. Pervaiz, A. Jalal, and K. Kim, *Hybrid algorithm for multi people counting and tracking for smart surveillance*, Int. Bhurban Conf. Appli. Sci. Technol., 2021, pp. 530–535.

[25] M. Pintor, D. Angioni, A. Sotgiu, L. Demetrio, A. Demontis, B. Biggio, and F. Roli, *ImageNet-Patch: A dataset for benchmarking machine learning robustness against adversarial patches*, Pattern Recog. **134** (2023), 109064.

[26] A. Raj and S. Susan, *Clustering Analysis for Newsgroup Classification*, Data Engineering and Intelligent Computing, Lecture Notes in Networks and Systems, 2022.

[27] A. Saxena, R. Gera, and S.R.S Iyengar, *A faster method to estimate closeness centrality ranking*, arXiv preprint arXiv:1706.02083 (2017).

[28] J.R. Taylor, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, Sausalito, CA: University Science Books, 1997.

[29] K. Thirumoorthy and K. Muneeswaran, *Feature selection for text classification using machine learning approaches*, Nat. Acad. Sci. Lett. **45** (2022), 51—56.

[30] A. van der Grinten, E. Angriman, M. Predari, and H. Meyerhenke, *New approximation algorithms for forest closeness centrality–for individual vertices and vertex groups*, Proc. SIAM Int. Conf. Data Min. (SDM), Soc. Ind. Appl. Math., 2021, pp. 136–144.

[31] A. Verm and V. Ranga, *Machine learning-based intrusion detection systems for IoT applications*, Wireless Pers Commun. **11** (2020), 2287–2310.

[32] S. Zhang and X. Pan, *A novel text classification based on Mahalanobis distance*, Int. Conf. Comput. Res. Dev., 2011, pp. 156–158.

[33] K. Zhao, Y. Dai, Z. Jia, and Y. Ji, *General fuzzy C-means clustering algorithm using Minkowski metric*, Signal Process. **188** (2021), 108161.

[34] Q. Zhou, P. Tang, S. Liu, J. Pan, Q. Yan, and S.-C Zhan, *Learning atoms for materials discovery*, Proc. Nat. Acad. Sci. **115** (2018), no. 28, 6411–6417.