Print ISSN: 0976-2876 Online ISSN: 2250-0138

Available online at: http://www.ijsr.in

INDIAN JOURNAL OF SCIENTIFIC RESEARCH

DOI:10.32606/IJSR.V12.I2.00025

Received: 10-10-2021 Accepted: 13-12-2021 Publication: 31-01-2022

Indian J.Sci.Res. 12 (2): 151-156 Original Research Article

CLUSTER BASED ANALYSIS OF GENE- DISEASE ASSOCIATION USING AUTOENCODERES

ARUN KUMAR[a] AND VISHAL VERMA[b1]

[ab]Department of Computer Science & Applications, CRSU, Jind, Haryana, India

ABSTRACT

The study of gene-disease associations is a critical area of research in genetics and medicine. It involves understanding the relationship between specific genes and the development of various diseases, providing insights into their causes, mechanisms, and potential treatments. In this study, researchers have done a Clustering analysis of gene-disease associations using autoencoders. Autoencoders are a type of neural network that can learn to encode and decode data, allowing them to identify complex patterns and relationships within large datasets. By applying autoencoders to gene-disease association data, researchers aimed to identify clusters of genes that are associated with specific diseases, providing insights into the underlying mechanisms and potential treatment strategies.

KEYWORD: Gene-Disease, Autoencoderes, Network, cluster

Gene-disease association studies have numerous benefits, including improved diagnosis, personalized medicine approaches, early detection and prevention, and drug development. However, there are also challenges, such as complex genetic interactions, sample size and data availability, and ethical considerations. Recent advancements in genome-wide association studies, next-generation sequencing, and functional genomics have revolutionized gene-disease association studies, providing deeper insights into the functional consequences of gene variants. Although there exist many clustering algorithms but researchers opted autoencoders for this study because of their ability to learn data representations and perform clustering in high-dimensional spaces, as well as their capacity to handle noisy data.

Autoencoders, while not a traditional clustering technique, can be advantageous in certain scenarios like Non-linear Dimensionality Reduction. Research papers support the effectiveness of autoencoders for clustering. For instance, "Deep Clustering Based on a Mixture of Autoencoders"; proposes a Deep Autoencoder Mixture Clustering (DAMIC) algorithm, which shows significant improvement over state-of-the-art methods in image and text corpora. Another paper, "Clustering of LMS Use Strategies with Autoencoders," (Verdú *et al.*, 2023) demonstrates the performance boost of autoencoder-based clustering over traditional methods such as K-means. Additionally, "DAC: Deep Autoencoder-based Clustering, a General Deep Learning Framework of Representation Learning"; (Lu and Li, 2021) presents a generalized data-driven framework to learn clustering representations using deep neural networks, showing the effectiveness of this approach in boosting the performance of K-Means. These papers provide evidence of the advantages of autoencoders (Chazan *et al.*, 2018) for clustering compared to traditional techniques (Bindra and Mishra, 2018).

LITERATURE REVIEW

In paper titled 'Prediction of Human Disease-Related Gene Clusters by Clustering Analysis' (Gang Sun *et al.*, 2011) researchers have implemented clustering algorithms such as the Markov cluster algorithm (MCL), Molecular complex detection (MCODE), and Clique percolation method (CPM) to partition the human protein-protein interaction (PPI) network into cohesive clusters that could potentially be associated with diseases. Subsequently, a log likelihood model was developed to evaluate and rank these dense clusters by considering multiple biological evidences. Ultimately, disease-related clusters were identified based on their higher scores within these dense clusters. Next in paper titled 'Heterogeneity among patients with Parkinson disease: Cluster analysis and genetic association' (Ma *et al.*, 2015) researchers employed cluster analysis (CA) to identify distinct subtypes of Parkinson;s disease (PD) and investigate the potential correlation between these subtypes and specific polymorphisms in the LRRK2 (G2385R and R1628P) and GBA (L444P) genes. A comprehensive k-means CA was conducted on a cohort of 1,510 Chinese PD patients obtained from the Chinese National Consortium on Neurodegenerative Diseases,

[1]Corresponding author

taking into account various factors such as demographics, disease progression, and both motor and non-motor symptoms. This paper (Chen *et al.*, 2018) introduces a innovative computational model called Bipartite Network Projection for MiRNA-Disease Association (BNPMDA) prediction. The model leverages existing knowledge of miRNA-disease associations, integrated miRNA similarity, and integrated disease similarity to make accurate predictions. This review paper (Kabekkodu *et al.*, 2018) focuses on the latest developments in miRNA cluster research and explores their regulation and biological functions in various pathological conditions. In this paper (Yuan *et al.*, 2020), researchers propose a novel method for predicting potential lncRNA-disease associations was proposed by constructing a bipartite network and calculating cluster association scores to evaluate the strength of relationships between disease and gene clusters. The implementation of Autoencoders for clustering tasks (Verdú *et al.*, 2023; Lu and Li, 2021; Chazan *et al.*, 2018) have shown tremendous performance over traditional clustering methods (van Dam *et al.*, 2018; Xu *et al.*, 2019; Qumsiyeh *et al.*, 2022; Sanjak *et al.*, 2023; Bose *et al.*, 2023; Richer *et al.*, 2023).

METHODOLOGY

Dataset

In this study, researchers utilized the DISGENET (Piñero *et al.*, 2020) dataset to do clustering using the effectiveness of autoencoders to plot and understand gene disease associations. DISGENET, a reputable organization, specializes in conducting research and providing high-quality datasets pertaining to disease gene associations. By leveraging the comprehensive and reliable information offered by DISGENET, the researchers. The dataset has shape (84038, 16) which means 84038 rows and 16 columns. The detailed information is given below (Figure 1).

```
assoc.shape
(84038, 16)

[ ] assoc.isnull().sum()
geneId                       0
geneSymbol                   0
DSI                         40
DPI                         59
diseaseId                    0
diseaseName                  0
diseaseType                  0
diseaseClass              3637
diseaseSemanticType          0
score                        0
EI                        4383
YearInitial               4383
YearFinal                 4383
NofPmids                     0
NofSnps                      0
source                       0
dtype: int64
```

```
assoc.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 84038 entries, 0 to 84037
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   geneId                84038 non-null  int64
 1   geneSymbol            84038 non-null  object
 2   DSI                   83998 non-null  float64
 3   DPI                   83979 non-null  float64
 4   diseaseId             84038 non-null  object
 5   diseaseName           84038 non-null  object
 6   diseaseType           84038 non-null  object
 7   diseaseClass          80401 non-null  object
 8   diseaseSemanticType   84038 non-null  object
 9   score                 84038 non-null  float64
 10  EI                    79655 non-null  float64
 11  YearInitial           79655 non-null  float64
 12  YearFinal             79655 non-null  float64
 13  NofPmids              84038 non-null  int64
 14  NofSnps               84038 non-null  int64
 15  source                84038 non-null  object
dtypes: float64(6), int64(3), object(7)
memory usage: 10.3+ MB
```

Figure 1: DataSet

For preparing data for autoencoders input, dataset has been cleaned as per needs i.e. removed not needed columns as per requirements. Also several categorical features have been encoded to numeral representatives for calculations. Also dataset values have been scaled and normalized so that model achieves better performance, stability, and interpretability. Here's what dataset looked like after data preprocessing (Figure 2)

```
# Scaling the data to bring all the attributes to a comparable level
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Normalizing the data so that the data
# approximately follows a Gaussian distribution
X_normalized = normalize(X_scaled)

# Converting the numpy array into a pandas DataFrame
X_normalized = pd.DataFrame(X_normalized)

# Renaming the columns
X_normalized.columns = X.columns

X_normalized.head()
```

	DSI	DPI	score	EI	NofPmids	NofSnps	diseaseIdEncoded	geneSymbolEncoded
0	0.503217	-0.310506	-0.192390	0.067367	-0.093687	-0.026002	-0.374867	-0.677368
1	0.520701	-0.321293	-0.199075	0.069708	-0.096942	-0.026906	-0.282430	-0.700902
2	-0.009809	0.067388	0.224518	-0.874914	0.116175	-0.012860	-0.231594	-0.334934
3	-0.023500	0.161436	-0.189655	0.079817	-0.111001	-0.030808	-0.523421	-0.802375
4	-0.023500	0.161438	-0.227949	0.079819	-0.111003	-0.030808	-0.507894	-0.802388

Figure 2: DataSet after Preproccessing

METHODS

The training process of an autoencoder involves two main steps: encoding and decoding. During the encoding step, the input data is passed through an encoder network, typically consisting of multiple layers, which progressively reduces the dimensionality of the data. The final layer of the encoder network represents the learned latent representation. Once the encoding step is complete, the decoding step begins. The encoded representation is passed through a decoder network, which aims to reconstruct the original input data. The decoder network has a symmetric structure to the encoder network, with each layer attempting to reconstruct the data at a higher level of detail until the final output layer. During training, the autoencoder aims to minimize the reconstruction error, which is the difference between the original input data and the reconstructed output. By minimizing this error, the autoencoder learns to capture the most salient features of the input data in the latent representation.

In the context of clustering, the learned latent representation can be used as a basis for grouping similar data points together. Each data point latent representation can be compared with others using distance metrics, such as Euclidean distance or cosine similarity. Data points with similar latent representations are likely to be clustered together, indicating similar characteristics or patterns in the original input data.

Based upon the requirements of dimensionality reduction and reconstruction of input data, model structure for an autoencoder, here is a breakdown of the model structure (Figure 3)

```
autoencoder.summary()

Model: "model_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 8)]               0

 dense_8 (Dense)             (None, 7)                 63

 dense_9 (Dense)             (None, 500)               4000

 dense_10 (Dense)            (None, 500)               250500

 dense_11 (Dense)            (None, 2000)              1002000

 dense_12 (Dense)            (None, 10)                20010

 dense_13 (Dense)            (None, 2000)              22000

 dense_14 (Dense)            (None, 500)               1000500

 dense_15 (Dense)            (None, 8)                 4008

=================================================================
Total params: 2303081 (8.79 MB)
Trainable params: 2303081 (8.79 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 3: Structure of Model

1. Input Layer: The model starts with an input layer that takes in data with a shape of (8,), indicating that it expects input samples with 8 features.
2. Encoder Layers: The input is passed through a series of dense (fully connected) layers to perform dimensionality reduction. The encoding_dim variable is set to 7, which means the output of the encoder layers will have a dimensionality of 7. The activation function used in the encoder layers is ReLU.
3. Decoder Layers: After the encoding layers, the model goes through a series of dense layers to reconstruct the original input. The decoder layers mirror the architecture of the encoder layers, with the same number of units in each layer. The output layer has 8 units, representing the reconstructed input.
4. Autoencoder Model: The autoencoder model is created by defining the input layer and the output layer (decoded layer).
5. Encoder Model: The encoder model is created by defining the input layer and the output layer (encoded layer), which represents the compressed representation of the input data.
6. Model Compilation: The autoencoder model is compiled using the Adam optimizer and the mean squared error loss function.

RESULTS

The model is trained for 25 epochs with a batch size of 128. The model is trained to reconstruct the input data itself as the target data. During the training process, the model aims to minimize the mean squared error (MSE) between the reconstructed output and the original input. As the epochs progress, the model learns to encode and decode the input data, gradually improving its ability to reconstruct the original data. Lower loss values indicate better reconstruction performance. Silhouette Score is 0.19014674016110625 .The image ahead represents the Reconstruction Error (Figure 4 & 5).

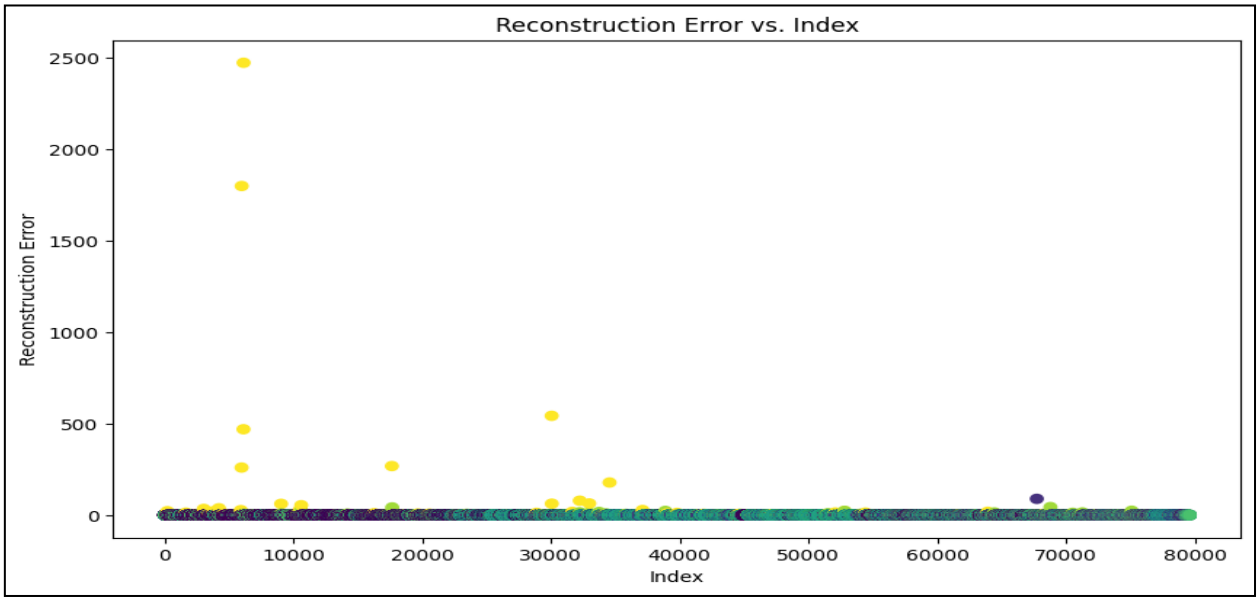

Figure 4: Reconstruction Error

For further plotting the data, researchers reduced dimensions to plot the clustered data by autoencoders. Here's the plot.

Clustered Data

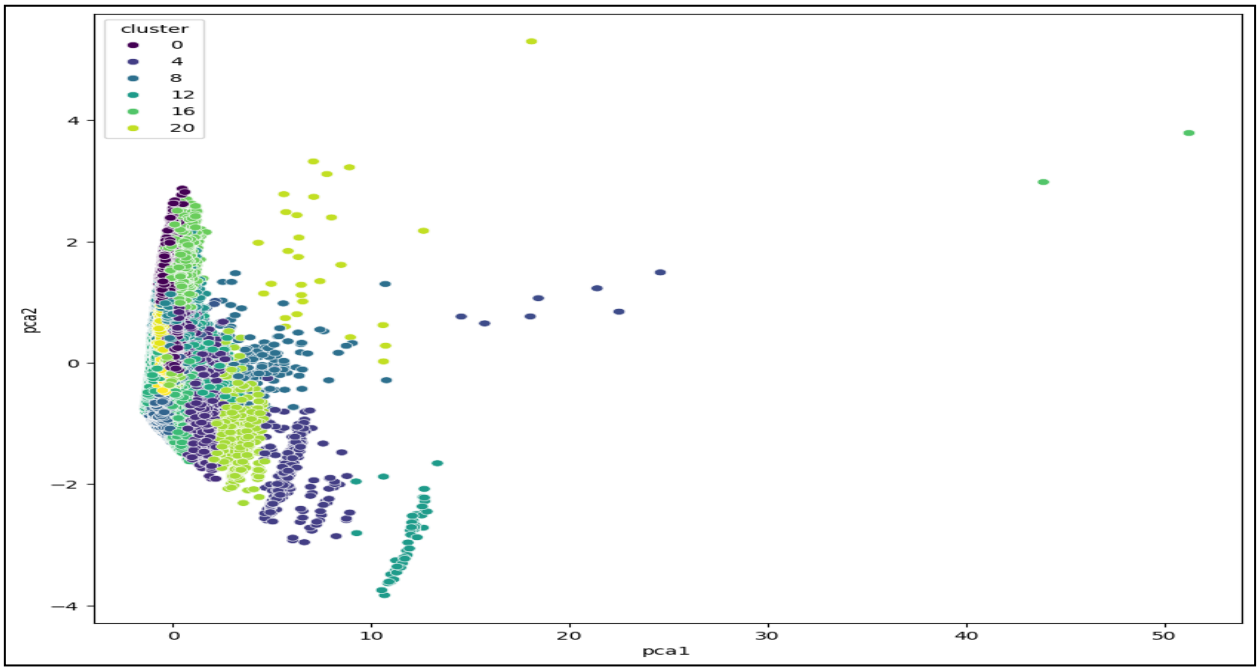

Figure 5: Clustered Data (Output of Model)

Future Study

For future study, research can be explored using more versatile methodologies for example - graph representation.

REFERENCES

Bindra K. and Mishra A., 2018. A Detailed Study of Clustering Algorithms, International Conference on Current Trends in Computer, Electrical, Electronics and Communication, CTCEEC 2017, pp. 752–757, doi: 10.1109/CTCEEC.2017.8454973.

Bose A., Burch M., Chowdhury A., Paschou P. and Drineas P., 2023. Structure-informed clustering for population stratification in association

studies. BMC Bioinformatics, **24**(1), doi: 10.1186/s12859-023-05511-w.

Chazan S.E., Gannot S. and Goldberger J., 2018. Deep Clustering Based on a Mixture of Autoencoders. (Online). Available: http://arxiv.org/abs/1812.06535.

Chen X., Xie D., Wang L., Zhao Q., You Z.H. and Liu H., 2018. BNPMDA: Bipartite network projection for MiRNA–Disease association prediction. Bioinformatics, **34**(18): 3178–3186, doi: 10.1093/bioinformatics/bty333.

Gang Sun P., Gao L. and Han S., 2011. Prediction of Human Disease-Related Gene Clusters by Clustering Analysis, (Online). Available: http://www.biolsci.org61

Kabekkodu S.P., Shukla V., Varghese V.K., D'Souza J., Chakrabarty S. and Satyamoorthy K., 2018. Clustered miRNAs and their role in biological functions and diseases. Biological Reviews, **93**(4): 1955–1986, doi: 10.1111/brv.12428.

Lu S. and Li R., 2021. DAC: Deep Autoencoder-based Clustering, a General Deep Learning Framework of Representation Learning. (Online). Available: http://arxiv.org/abs/2102.07472

Ma L.Y., Chan P., Gu Z.Q., Li F.F. and Feng T., 2015. Heterogeneity among patients with Parkinson's disease: Cluster analysis and genetic association. J. Neurol. Sci., **351**(1–2): 41–45, doi: 10.1016/j.jns.2015.02.029.

Piñero J., Ramírez-Anguita J.M., Saüch-Pitarch J., Ronzano F., Centeno E., Sanz F. and Furlong L.I., 2020. The DisGeNET knowledge platform for disease genomics: 2019 update. Nucleic Acids Res., **48**(D1): D845–D855, doi: 10.1093/nar/gkz1021.

Qumsiyeh E., Showe L. and Yousef M., 2022. GediNET for discovering gene associations across diseases using knowledge based machine learning approach. Sci. Rep., **12**(1), doi: 10.1038/s41598-022-24421-0.

Richer S., Tian Y., Schoenfelder S., Hurst L., Murrell A. and Pisignano G., 2023. Widespread allele-specific topological domains in the human genome are not confined to imprinted gene clusters. Genome Biol., **24**(1), doi: 10.1186/s13059-023-02876-2.

Sanjak J., Binder J., Yadaw A.S., Zhu Q. and Mathé E.A., 2023. Clustering rare diseases within an ontology-enriched knowledge graph. Journal of the American Medical Informatics Association, **31**(1): 154–164, doi: 10.1093/jamia/ocad186.

van Dam S., Võsa U., van der Graaf A., Franke L. and de Magalhães J.P., 2018. Gene co-expression analysis for functional classification and gene-disease predictions. Brief Bioinform, **19**(4): 575–592, doi: 10.1093/bib/bbw139.

Verdú M.J., Regueras L.M., de Castro J.P. and Verdú E., 2023. Clustering of LMS Use Strategies with Autoencoders. Applied Sciences (Switzerland), **13**(12), doi: 10.3390/app13127334.

Xu Y., Xing L., Su J., Zhang X. and Qiu W., 2019. Model-based clustering for identifying disease-associated SNPs in case-control genome-wide association studies. Sci. Rep., **9**(1), doi: 10.1038/s41598-019-50229-6.

Yuan Q., Guo X., Ren Y., Wen X. and Gao L., 2020. Cluster correlation based method for lncRNA-disease association prediction. BMC Bioinformatics, **21**(1): 1, DUMM, doi: 10.1186/s12859-020-3496-8.