Similarity based Deep Neural Networks

Seungyeon Lee^{1*}, Eunji Jo^{1*}, Sangheum Hwang², Gyeong Bok Jung³, and Dohyun Kim¹

¹Department of Industrial and Management Engineering, Myongji University, Yongin, Korea

²Department of Industrial and Information Systems Engineering, Seoul National University of Science and Technology, Seoul, Korea

³Department of Physics Education, Chosun University, Gwangju, Korea

Ijfis

Abstract

Deep neural networks (DNNs) have recently attracted attention in various areas. Their hierarchical architecture is used to model complex nonlinear relationships in high-dimensional data. DNNs generally require large numbers of data to train millions of parameters. However, the training of a DNN with a small number of high-dimensional data can result in an overfitting. To alleviate this problem, we propose a similarity-based DNN that can effectively reduce the dimensionality of the data. The proposed method utilizes a kernel function to calculate pairwise similarities of observations as input, and the nonlinearity based on the similarities is then explored using a DNN. Experiment results show that the proposed method performs effectively regardless of the dataset used, implying that it can be applied as an alternative when learning a small number of high-dimensional data.

Keywords: Deep neural networks (DNNs), Kernel method, Feature extraction, High-dimensional data

1. Introduction

In recent years, deep neural networks (DNNs) have attracted attention in the field of machine learning. DNNs consist of a hierarchical structure with more than one hidden layer between an input layer and an output layer, through which they can model complex nonlinear relationships in high-dimensional data. However, DNNs are often affected by the dimensionality of the features. In particular, when training DNNs with a small number of high-dimensional data, such as in a sequencing or spectrum analysis, an overfitting, low prediction performance, and high computing cost can arise. Dimensionality-reduction algorithms have been proposed to solve these problems. The algorithms can be classified into two categories: feature selection and feature extraction [1].

With feature selection, a subset of features is directly selected without altering the original representation of the features. These algorithms can be broadly classified into similarity-, information-theory-, statistical-, and sparse-learning-based methods [2]. In the case of similarity-based methods such as the Laplacian score and a spectral analysis based feature selection using expanded Laplacian and Fisher scores, the importance of the features is assessed based on the ability to preserve similarities between observations. In addition, information-theory-based methods such as information gain and minimum redundancy maximum relevance use different heuristic filter criteria to measure the importance of the features [2]. These

Received: Feb. 10 2021

Accepted: Sep. 2, 2021

Correspondence to: Dohyun Kim
 (ftgog@mju.ac.kr)

*These authors contributed equally to this work.
 ©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

methods maximize the feature relevance and minimize the feature redundancy. A statistical-based method relies on various statistical measures instead of learning algorithms to evaluate the relevance of the features. In this case, the features with high redundancy are removed. Finally, sparse learning-based methods, such as l_p -norm regularization, can minimize the fitting error along with the sparse regularization term. The sparse regularizer can eliminate features by making the coefficients of the features zero or a value of near zero.

Feature extraction algorithms combine the original features to create new features. These methods project the original high-dimensional features into a new feature space with low dimensionality [1]. Typical examples of feature extraction algorithms include a principal component analysis (PCA) and autoencoders. A PCA generates new features represented by linear combinations of the original features. A PCA attempts to create new features in a linear subspace by maximizing the variance of the data [3]. The dimensionality of the features can be reduced by selecting from among the newly extracted features those that properly describe the data. However, a standard PCA is not helpful when the data cannot be expressed well in a linear subspace. Fortunately, a standard PCA can be generalized into a nonlinear form using kernel functions [4]. An autoencoder is a deep learning method that reconstructs the original features of the input in its output to create new nonlinear features. The autoencoder extracts nonlinear features from the middle hidden layer, the dimensions of which are smaller than the input features. The model is learned by reconstructing the input data in the output layer, and thus it can obtain features that reflect the important characteristics of the data.

Before analyzing a small number of high-dimensional data with a DNN, a PCA or an autoencoder is used to reduce the dimensionality. These approaches can decrease the complexity of the DNN by directly reducing the features of the data. However, because the features are extracted without considering the target, the models can exhibit a low prediction performance.

In this study, we propose a similarity-based deep neural network for small high-dimensional data, which simplifies the model by utilizing the similarities between observations rather than directly reducing the dimensionality of the features. The proposed method can mitigate the problems of existing DNNs, such as the model complexity and run time. The following are the contributions of the proposed method. 1) Using the similarities of the observations, the method can predict labels without direct access to the features for a small number of high-dimensional data. 2) Through the kernel function, the raw data

are projected in the feature space, and the nonlinearity is explored using the DNNs. 3) The proposed method effectively reduces the complexity of the model without directly reducing the dimensionality of the features. 4) Using only the similarities between observations, the proposed method shows a good performance.

The remainder of this paper is organized as follows: Section 2 describes the similarity-based method. Section 3 describes the process of the proposed method in detail. In Section 4, the proposed method and the basic DNN model are compared and the results are presented. Finally, Section 5 summarizes the contributions of this study and discusses issues for further research.

2. Similarity Based Approach

Machine learning algorithms often deal with high-dimensional datasets and complex nonlinear models, which require an optimization problem of high dimensionality. One of the common approaches to solving this is the primal-dual method, which can also be applied to a variety of tasks, including a regression problem. By solving the problem through primal-dual methods, the computational efficiency significantly increases in many cases, and specific characteristics can be used more efficiently [5]. A support vector machine (SVM) is a representative method based on primal and dual methods. A primal SVM is a feature-based approach that directly models the data features, whereas the dual SVM is a similarity-based approach that models the data similarities.

In this section, we deal with a feature-based approach (primal problem) using features directly, and a similarity-based approach (dual problem) utilizing data similarities. We then describe the differences between the two approaches.

A training dataset $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ consists of observations $\mathbf{x}_i, i = 1, \dots, N$, and its corresponding label y_i , where N denotes the number of data. A linear SVM model $f(\mathbf{x})$ that predicts the label y can be expressed as follows:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}^T \mathbf{x} = \sum_{i=1}^P w_i x_i, \quad (1)$$

where $\mathbf{x} = \{x_1, x_2, \dots, x_P\}^T$ is a P -dimensional input vector, and $\mathbf{w} = \{w_1, w_2, \dots, w_P\}^T$ is the coefficient vector of the input vector. The discriminant function is as follows:

$$y = \begin{cases} 0, & \text{if } \mathbf{w}^T \mathbf{x} + b < 0, \\ 1, & \text{if } \mathbf{w}^T \mathbf{x} + b \geq 0. \end{cases} \quad (2)$$

In Equations (1) and (2), the data features are directly entered into the model and utilized to predict the label. This formulation is considered a primary problem.

The discriminant function can be represented as a dual problem using the dual formulation of the optimization problem. The dual problem focuses only on the inner product between observations and not on the features. In the dual SVM, model $f(\mathbf{x})$ is expressed as

$$f(\mathbf{x}) = \sum_{i=1}^N a_i y_i \mathbf{x}_i^T \mathbf{x} + \hat{b}. \quad (3)$$

The primal problem models the features, whereas the dual problem models the linear combination of the inner product using all of the data. Therefore, when the number of training data is smaller than the number of features, the dual problem is faster than the primal problem. In addition, the dual representation can express the nonlinearity of the data using a kernel function, unlike with the primal problem.

For real-world problems, it is often necessary to model more complex representations beyond the linear dependence of the data. A kernel function captures the nonlinear data representations and can measure the similarities between observations in a high-dimensional feature space, eliminating the need to map observations directly to this space [6]. The kernel function can be applied in many different ways, such as a polynomial kernel and radial basis function (RBF) kernel. First, a linear kernel is calculated using the inner product and is expressed as follows:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j. \quad (4)$$

Second, the Gaussian or RBF kernel is in the form of a radial basis function and is expressed as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left\{ -\gamma \| \mathbf{x}_i - \mathbf{x}_j \|^2 \right\}. \quad (5)$$

The RBF kernel is confirmed to be a valid kernel by expanding the square term and allows the feature vector corresponding to the RBF kernel to have an infinite number of dimensionalities [7].

The kernel SVM is a model that applies the kernel function to the dual SVM to measure the similarity in a high-dimensional feature space [8]. The dual model $f(\mathbf{x})$ is expressed as follows:

$$f(\mathbf{x}) = \sum_{i=1}^N a_i y_i \mathbf{x}_i^T \mathbf{x} + \hat{b} = \sum_{i=1}^N a_i y_i k(\mathbf{x}_i, \mathbf{x}) + \hat{b}. \quad (6)$$

Using the dual formulation, the parameter a_i is greater than or equal to zero. Therefore, the observation $\mathbf{x}_i$ with $a_i = 0$ does not affect the prediction function $f(\mathbf{x})$. The prediction is applied only on $\mathbf{x}_i$ with $a_i \neq 0$, which is a support vector. As a result, the kernel SVM predicts y through similarities between the input data $\mathbf{x}$ and the support vectors $\mathbf{x}_i$.

In the dual problem, through the kernel function, y corresponding to the input data $\mathbf{x}$ can be predicted based on similarities with the training data in a high-dimensional feature space. Thus, unlike the primal problem, the dual problem can be viewed as a similarity-based approach because it uses similarities between points rather than features. When the primal and dual methods are applied to the SVM, the primal problem should obtain P parameters, whereas the dual problem should obtain N parameters. Therefore, a dual formulation is useful in terms of the number of model parameters when using a small number of high-dimensional data ($N < P$). In a DNN, the difference in the model parameters is much greater because of the complex architecture. However, because most DNNs use a primal approach, if $N \ll P$, the model is more complicated than when using a dual approach. To take advantage of a dual approach, we propose a method for applying such a DNN approach to effectively reduce the dimensionality and model complexity.

3. Similarity based Deep Neural Network

In this study, we propose a similarity-based DNN. The proposed method can effectively reduce the dimensionality of the data by adopting a similarity-based approach to a DNN when the number of features is much larger than the number of data.

With the proposed method, the dimensionality can be reduced using the similarities between the data as input. Each observation is newly configured as similarities with the entire training data and similarities enter into input nodes. Figure 1 shows the input layer of the proposed method and a basic DNN model. The proposed method uses the similarities between data as input, whereas the basic DNN uses P features as input. To obtain the pairwise similarity in a high-dimensional feature space, we utilize the linear and RBF kernel functions, which are denoted as $k(\cdot)$. The nonlinearity of the data was explored using a DNN. Figure 2 shows a flowchart of the entire process.

The notation for the proposed method is expressed as follows:

Figure 1. Input layer of proposed method and basic DNN.

Figure 2. Flowchart of proposed method.

Here, $\mathbf{X} \in R^{N \times P}$ is a raw dataset with rows $\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_N^T$, where $\mathbf{x}_i^T$ is a vector consisting of $\{x_{i1}, x_{i2}, \dots, x_{iP}\}^T$, N is the number of observations, and P is the number of features. The kernel function for measuring the similarity can be given as $\mathbf{K} \in R^{N \times N}$, where the elements of $\mathbf{K}$ are $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Because P is much larger than N , the dimension is reduced

from $R^{N \times P}$ to $R^{N \times N}$. Here, each row of $\mathbf{K}$ represents the observations, which are newly composed as similarities.

The proposed method conducts a prediction based on the similarity in the high-dimensional feature space obtained using the kernel function. Thus, when applying a small sample of high-dimensional data (i.e., $N \ll P$), the number of input nodes in the proposed DNN is reduced from P (the number of features) to N (the number of observations).

This proposed method can effectively reduce the number of input nodes in a DNN by adopting a similarity-based approach. The pairwise similarities of the observations are calculated using the kernel function, and the nonlinearity of the data is then explored through the application of the DNN with the similarities obtained as input. As a result, the proposed method can effectively simplify the model when using a small number of samples of high-dimensional data. In addition, the proposed method is advantageous in terms of computational efficiency and running time compared to conventional dimension reduction methods such as a PCA and an autoencoder. A PCA must compute $O(P^3)$ to obtain the principal components. By contrast, the autoencoder should be trained to reduce the dimensionality, which increases the running time. However, the proposed method only requires calculating a similarity matrix whose number of computations is $O(P^2)$, leading to reduced computations and learning time compared to those of our an autoencoder and a PCA.

3.1 Selection of Important Observations

The proposed method reduces the dimensionality through the similarities. However, the similarities with all observations may be unimportant. Accordingly, we utilize a variant of a rectified linear unit (ReLU) based sparse autoencoder [9] to consider

Figure 3. Selection of important observations.

only similarities with important observations. As the only difference between the proposed and existing methods, the existing method is applied to the autoencoder, whereas the proposed method is applied to a basic DNN. This can be thought of as the same process as finding support vectors in an SVM. The ReLU-based sparse autoencoder was proposed to select important features by making the input nodes sparse. The algorithm adds a hidden layer of the same number of dimensions as the input layer and connects the nodes in a one-to-one manner (Figure 3). These hidden nodes use the ReLU activation function, $f(x) = \max(0, x)$ [10]; therefore, the negative input of ReLU does not affect the next layer, that is, features with positive input are selected by the ReLU. When all input data are positive, the hidden node returns to zero if its weight is negative. As a result, features are selected according to the weights trained during the DNN training process. It is therefore essential to convert the input data into positive values to select the important features. In this study, we applied this algorithm to the proposed method. Using only similarities with important observations reduces the complexity of the model and obtains a generalized model.

4. Experiment

4.1 Experiment Setup

We evaluated the proposed methods, combined with the two kernels (linear and RBF), by comparing them with the existing method using features as the input. The proposed sparse method with an additional step for selecting important observations was also compared. Because the input data types of the proposed and existing methods differ, the two models consist of different structures. Therefore, to minimize the sum of the margin losses, we defined the experimental protocols that are commonly used in the experiments using both methods. In the experiments, we applied classification tasks on two datasets, and all experiments

were implemented in PyTorch. We used the Adam optimizer and initialized parameters with a normal distribution $N(0, 0.1^2)$. Both methods consist of fully connected hidden layers and a ReLU activation function. The output of the last hidden layer is fed to a softmax layer with three classes for classification. We utilized a three-fold cross-validation for a performance measurement. We provided the results using two performance measures: the accuracy and the number of parameters. The accuracy is the rate at which the labels of the validation set are accurately predicted and averaged over the three validation sets. To alleviate an overfitting, we used a ridge regularization, and lambda was the penalty parameter applied for regularization. In all experiments, each model had the same node for all hidden layers. The numbers of hidden layers and nodes, lambda, and the learning rate were tuned through a grid search.

4.2 Datasets

Computational experiments were conducted using two spectral datasets. Both datasets are high-dimensional with a small number of samples (Table 1). The detailed dataset descriptions and statistics are as follows.

The Brain dataset was collected using Raman spectroscopy, which is widely applied to characterize the biology organization. The dataset shows biochemical changes in the hippocampal region of the brain. The vector normalization method was used to correct the baseline of the Raman spectra, and the spectral data of the brain tissue were calculated as the average of 10 measured samples. The dataset consists of 30 observations with 10 measurements per class (i.e., normal, ischemic, and neuronal nitric oxide synthase [nNOS] inhalation) with 2,501 frequencies ($500\text{--}1,750\text{ cm}^{-1}$) [11].

The BV dataset represents biochemical changes at the molecular level in human MDA-MB-231 breast cancer cells following exposure to bee venom (BV). It consists of Raman spectral data measured at various concentrations. The dataset has 30 observations: 10 observations for each concentration (i.e., 0.7, 1.5, and $3.0\text{ }\mu\text{g/mL}$). Raman intensities were measured at 2,301 frequencies ($600\text{--}1,750\text{ cm}^{-1}$) [12].

Table 1. Experimental data

Dataset	Observations	Features	Classes
Brain	30	2,501	3
BV	30	2,301	3

4.3 Results

After finding the optimal hyper-parameters using the three-fold cross-validation, we obtained the average mean and standard deviation of the accuracy over 100 runs. The results are summarized in Table 2 and Figure 4. In Table 2, the proposed method using the RBF kernel is highly competitive. The proposed method achieves a similar or higher accuracy than the existing DNN approach. In the BV dataset, the proposed method using similarity as the input showed a performance improvement of 2.41% and a reduction in the number of parameters of 99.4%, compared to the existing DNN method using features as input. In the Brain dataset, although the proposed method slightly improved the performance by approximately 0.5%, the standard deviation of the proposed method was lower than that of the existing model. This implies that the proposed method is more stable. These performance improvements are thought to be due to the reduced dimensionality of the input data (i.e., the model parameters), which effectively reduces the model complexity and is therefore less affected by an overfitting.

Experiment results show that the type of kernel used to measure the similarity of the data is important. As listed in Table 2, the performance was improved only when using the RBF kernel, whereas using the linear kernel degraded the performance. The linear kernel measures the similarities among the data linearly using the inner product, whereas the RBF kernel computes the similarities implicitly based on infinite-dimensional features. These characteristics of the RBF kernel can effectively reflect the complicated relations between observations. Therefore, the proposed method using the RBF kernel can be a useful candidate for modeling complicated high-dimensional data.

We conducted additional experiments to select the important observations, which correspond to the proposed sparse method

Figure 4. Plot of prediction accuracy.

in Table 2. The proposed sparse method is less accurate than the proposed method. However, this method has the advantage of detecting important observations. Notedly, it showed a competitive performance despite the similarities (i.e., input features) being removed by approximately 15% or more. This method can be a useful alternative when applying similarities with all observations is neither a possible nor practical approach.

5. Conclusion

In this study, a new method was proposed for solving problems when using a small number of high-dimensional data samples. The training of a DNN with high-dimensional data can result in model complexity, a decreased learning accuracy, and compu-

Table 2. Classification results for each dataset

Dataset	Method	Accuracy	Number of parameters	Number of input nodes removed
Brain	DNNs	0.979 ± 0.064	50,880	0
	Proposed method (linear kernel)	0.967 ± 0.000	2,490	0
	Proposed method (RBF kernel)	0.985 ± 0.017	22,300	0
	Proposed sparse method (RBF kernel)	0.912 ± 0.059	1,260	4 (20%)
BV	DNNs	0.868 ± 0.030	4,304,000	0
	Proposed method (linear kernel)	0.767 ± 0.020	96,900	0
	Proposed method (RBF kernel)	0.889 ± 0.025	24,470	0
	Proposed sparse method (RBF kernel)	0.866 ± 0.025	3,670	3 (15%)

tational costs. The proposed similarity-based approach can be used to solve such problems. First, the similarities between observations are measured in a high-dimensional feature space by the kernel function, and the nonlinearity of the data is then explored using a DNN. The proposed method reduces the number of input nodes in the DNN and simplifies the model. Experimental results show that the proposed method can be a useful alternative when training a DNN using a small number of samples of high-dimensional data. This implies that the proposed method can be used in a variety of machine learning algorithms for small samples of high-dimensional data. In addition, we conducted additional experiments by applying a ReLU-based sparse autoencoder to the proposed method. The model can be further simplified using only the similarities with important observations. The proposed method is a suitable alternative when modeling the data by considering only those similarities with important observations, such as the support vectors of an SVM. As a future study, it would be interesting to investigate the relationships between observations and data modeling using such relationships. Applying these findings to regression tasks may also be a fruitful area for future research.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. NRF-2017R1E1A1A01077375).

References

- [1] K. Z. Mao, "Identifying critical variables of principal components for unsupervised feature selection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 2, pp. 339-344, 2005. <https://doi.org/10.1109/TSMCB.2004.843269>
- [2] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: a data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1-45, 2017. <https://doi.org/10.1145/3136625>
- [3] Q. Guo, W. Wu, D. L. Massart, C. Boucon, and S. De Jong, "Feature selection in principal component analysis of analytical data," *Chemometrics and Intelligent Laboratory Systems*, vol. 61, no. 1-2, pp. 123-132, 2002. [https://doi.org/10.1016/S0169-7439\(01\)00203-9](https://doi.org/10.1016/S0169-7439(01)00203-9)
- [4] B. Scholkopf, A. Smola, and K. R. Muller, "Kernel principal component analysis," in *Artificial Neural Networks – ICANN'97*. Heidelberg, Germany: Springer, 1997, pp. 583-588. <https://doi.org/10.1007/BFb0020217>
- [5] N. Komodakis and J. C. Pesquet, "Playing with duality: an overview of recent primal? Dual approaches for solving large-scale optimization problems," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 31-54, 2015. <https://doi.org/10.1109/MSP.2014.2377273>
- [6] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210-219, 2017. <https://doi.org/10.1016/j.neucom.2017.06.005>
- [7] R. Scholkopf, C. J. C. Burges, and A. J. Smola, *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT Press, 1999.
- [8] B. Scholkopf, "The kernel trick for distances," *Advances in Neural Information Processing Systems*, vol. 13, pp. 301-307, 2001.
- [9] S. Kim, S. Hwang, D. Yoon, and D. Kim, "Unsupervised feature selection for autoencoder," in *Proceedings of 2019 Spring Conference on Korean Institute of Industrial Engineers*, Gwangju, Korea, 2019, pp. 1330-1356.
- [10] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014 [Online]. Available: <https://arxiv.org/abs/1412.6980>.
- [11] G. B. Jung, S. W. Kang, G. J. Lee, and D. Kim, "Biochemical characterization of the brain hippocampal areas after cerebral ischemia-reperfusion using Raman spectroscopy," *Applied Spectroscopy*, vol. 72, no. 10, pp. 1479-1486, 2018. <https://doi.org/10.1177/0003702818776627>
- [12] G. B. Jung, J. E. Huh, H. J. Lee, D. Kim, G. J. Lee, H. K. Park, and J. D. Lee, "Anti-cancer effect of bee venom on human MDA-MB-231 breast cancer cells using Raman spectroscopy," *Biomedical Optics Express*, vol. 9, no. 11, pp. 5703-5718, 2018. <https://doi.org/10.1364/BOE.9.005703>

Seungyeon Lee received her B.S. and M.S. degrees in industrial and management engineering from Myongji University, Korea, in 2018 and 2020, respectively. Her research interests include statistical data mining, deep learning, recommender systems, and graph data analysis.

E-mail: sylee1@mju.ac.kr

Eunji Jo received her B.S. and M.S. degrees in industrial and management engineering from Myongji University, Korea, in 2018 and 2020, respectively. Her research interests include statistical data mining, machine learning, deep learning, and series data analysis.

E-mail: goodji@mju.ac.kr

Sangheum Hwang received his Ph.D. degree in industrial and systems engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2012. He is currently an assistant professor at the Department of Industrial and Information

Systems Engineering, Seoul National University of Science and Technology. His research interests are in the areas of statistical learning methods, kernel machines, and deep learning.

E-mail: shwang@seoultech.ac.kr

Gyeong Bok Jung received her Ph.D. degree from the Department of Applied Physics at the School of Engineering at the University of Tokyo, Japan. She is currently an assistant professor with the Department of Physics Education at Chosun University in

Korea. Her research interests include surface-enhanced Raman scattering and biomedical applications.

E-mail: gbyung@Chosun.ac.kr

Dohyun Kim received his M.S. and Ph.D. degrees in industrial engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2002 and 2007, respectively. He is currently an associate professor with the Department of Industrial and Management Engineering, Myongji University. His research interests include statistical data mining, deep learning, and graph data analysis.

E-mail: ftgog@mju.ac.kr