Deep Tensor: Eliciting New Insights from Graph Data that Express Relationships between People and Things

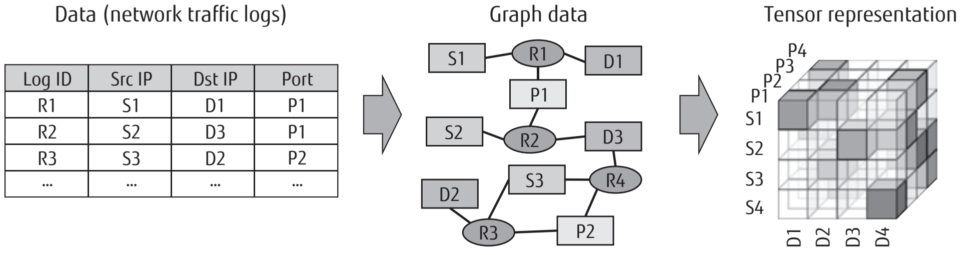
● Koji Maruhashi

An important problem in information and communications technology (ICT) is classifying graph data that expresses the relationships between people and things. For example, how can cyberattacks be detected by using network traffic logs showing the relationships between the source IP address and the destination IP addresses and ports, and how can fraudulent activities be detected by using banking transactions showing the relationships between senders and receivers and bank branches? When classifying large volumes of graph data, however, there are many yet-to-be-expressed features in the partial graphs used in conventional graph learning methods, so there are limits to achieving accurate classification. We propose using a novel tensor decomposition method called "Deep Tensor" for leveraging a deep neural network to enable it to automatically extract these features of graph data. Experiments in three different domains demonstrated that use of this decomposition method results in high accuracy for various types of graph data, enabling interpretation based on the activity of the neural network.

1. Introduction

Massive amounts of data have been accumulated thanks to the development of high-speed communication and the Internet of Things (IoT), including many types of data that can be analyzed as graph data that express relationships between people and things. Typical examples are network traffic logs showing relationships between source IPs (src IPs), destination IPs (dst IPs), and ports.

An example graph representation of a network traffic log is shown in the center panel in **Figure 1**. Records, represented as elliptical nodes, are connected to elements, represented as rectangular nodes. The corresponding tensor representation is shown in the right panel. Connections between elements are expressed as non-zeros. The problem is to learn predictive model μ that can predict class y of graph data g as accurately as possible on the basis of the given set of training data. The training data contains a set of pairs of graph data g_i and their class y_i (**Figure 2**). This problem is strongly related to the problems addressed by 'FUJITSU Human Centric AI Zinrai' technology, including intrusion detection based on the connectivity between IPs and ports and detection of online banking fraud based on the relationships between senders, receivers, and bank branches. Conventional methods use a



Log ID	Src IP	Dst IP	Port
R1	S1	D1	P1
R2	S2	D3	P1
R3	S3	D2	P2
...	...	...	...

Figure 1
Graph data and tensor representation.

set of sub-graphs designed by specialists knowledgeable about the data. However, these methods cannot achieve high classification accuracy because many important features are missed, especially in large-scale graph data with a highly complicated structure. For example, a set of network traffic logs is graph data with a complicated structure even though they are observed within only a few minutes (**Figure 3**).

We have developed a novel tensor decomposition method called "Deep Tensor (DT)" for learning and classifying graph data by using a deep neural network. DT automatically learns how to extract the features of graph data by utilizing a tensor decomposition technique.

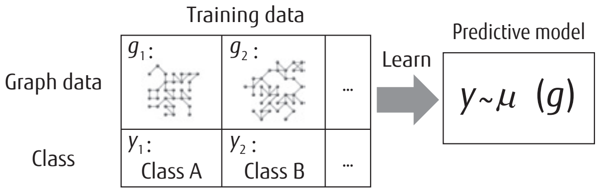


Figure 2
Learning graph data.

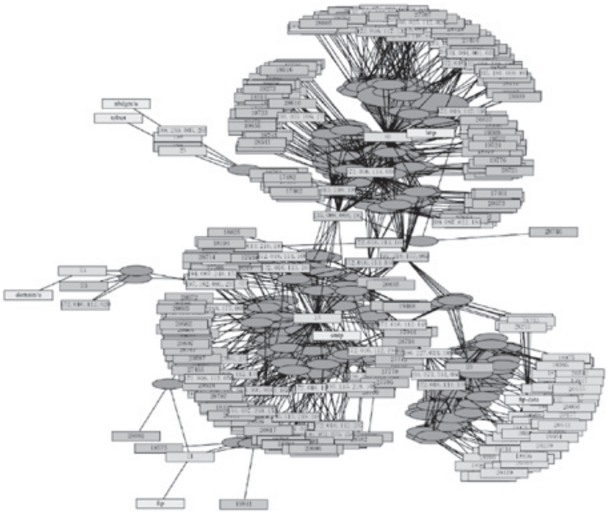


Figure 3
Example of graph data expressing network traffic logs.

In this paper we report the experimental results for three different domains: intrusion detection, peer-to-peer (P2P) lending, and drug discovery. The results show that our method can achieve high accuracy for many types of graph data and thus enable further inspection of the results along with the activity of neurons.

2. Feature extraction of graph data

Previous work related to feature extraction of graph data can be categorized as designing sub-graphs, learning sub-graphs using deep learning, and extracting features using tensor decomposition.

2.1 Designing sub-graphs

As mentioned above, conventional graph learning methods use a set of sub-graphs designed by specialists knowledgeable about the graph data (**Figure 4**). Support vector machines (SVMs) using graph kernels[1)]are based on the same idea even though they do not explicitly enumerate all the sub-graphs. However, they cannot achieve high classification accuracy because it is hard for pre-designed sub-graphs to express all the important features of many types of graph data.

2.2 Learning sub-graphs using deep learning

Recent advances in deep learning using deep neural networks have enabled us to automatically generate useful features for many types of data in speech recognition and image recognition.[2)] A convolutional

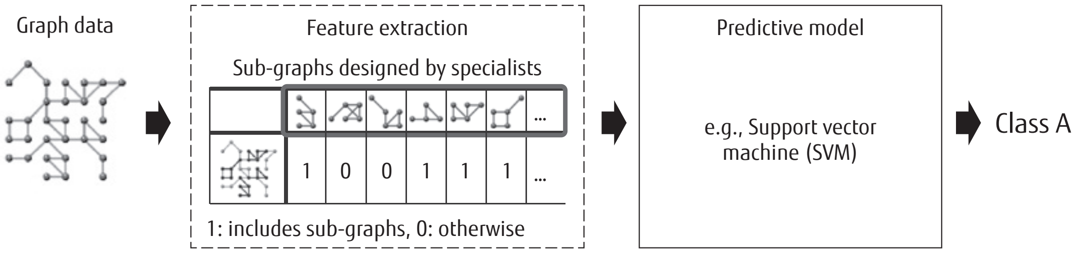

Graph data
Feature extraction
Sub-graphs designed by specialists
...
1
0
0
1
1
1
...
1: includes sub-graphs, 0: otherwise
Predictive model
e.g., Support vector machine (SVM)
Class A

Figure 4
Conventional methods based on pre-designed sub-graphs.

neural network can be used to learn sub-structures of graph data from matrices that express the connectivity of the data.[3] However, it is not trivial to find the optimal alignments of elements in the adjacency matrices.

2.3 Extracting features using tensor decomposition

As shown in Figure 1, graph data can be expressed as a tensor, and the tensor can be approximated as a core tensor multiplied by factor matrices (**Figure 5**).[4] The core tensor expresses the major connectivity structure of the graph data.[5] However, the alignments of the elements in the core tensor can be decided freely, and it is not trivial to find the optimal alignments for classification. Unlike conventional tensor decomposition, DT can learn how to extract the optimal core tensor so that high classification accuracy can be achieved.

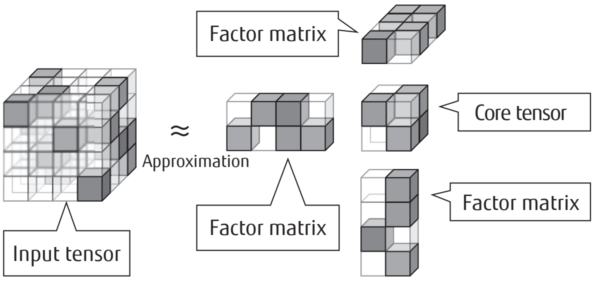


Figure 5
Conventional tensor decomposition.

3. Deep Tensor

An overview of DT is shown in **Figure 6**. Given a tensor expressing graph data, DT calculates a core tensor and factor matrices by using structure restricted tensor decomposition (SRTD) and uses the core tensor as input to a neural network. SRTD calculates the core tensor as closely as possible to a target core tensor expressing key sub-structures for classification. DT optimizes the target core tensor in order to maximize the classification accuracy by using an extension of the backpropagation algorithm used in training conventional neural networks.

3.1 Structure restricted tensor decomposition

Conventional tensor decomposition does not necessarily locate key sub-structures for classification in similar indices of core tensors calculated from every graph data instance. We have developed an SRTD method that not only approximates the input tensor but also calculates the core tensor as closely as possible to the target core tensor so that the key sub-structures should be located in similar indices of the core tensor. We can accurately classify graph data by using these core tensors via neural networks trained by using these tensors. SRTD is calculated using a two-step optimization (**Figure 7**). In the first step, factor matrices are optimized so that the target core tensor multiplied by the factor matrices approximates the input tensor as closely as possible. In the second step, the core tensor is optimized so that the core tensor multiplied by the optimized factor matrices approximates the input tensor as closely as possible.

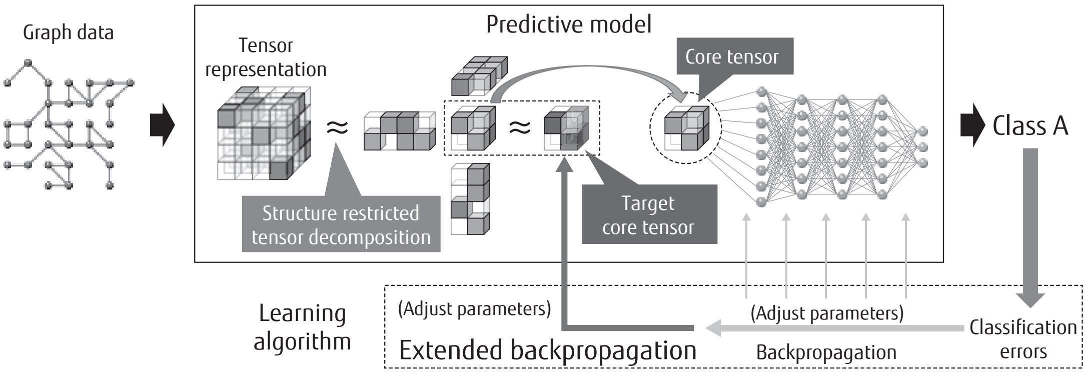


Figure 6
Overview of Deep Tensor.

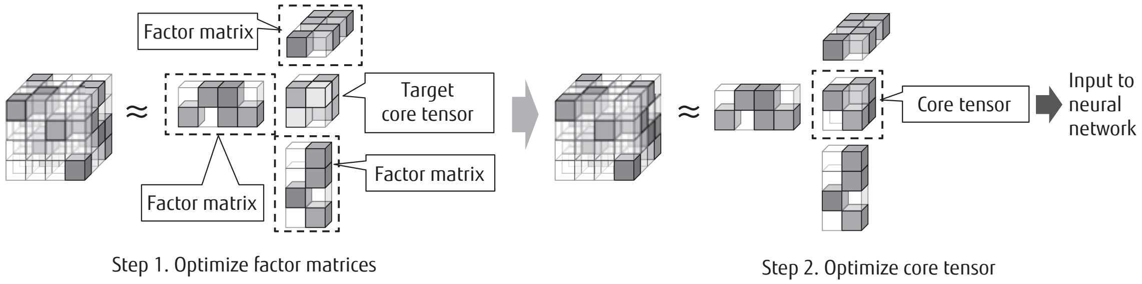


Figure 7
Structure restricted tensor decomposition.

3.2 Extended backpropagation (EBP)

We have developed an EBP method that optimizes the target core tensor in order to achieve better classification accuracy. The conventional backpropagation algorithm calculates the parameter adjustments needed to minimize the classification errors by propagating the classification errors toward the lower layers of the neural network. EBP extends the algorithm so that the errors are propagated to the target core tensor and calculates the degree to which the target core tensor should be adjusted. It updates the target core tensor along with the neural network parameters by using the stochastic gradient descent (SGD) algorithm.

4. Empirical results

As mentioned, we evaluated our method by using datasets from three different domains: intrusion detection, P2P lending, and drug discovery.

4.1 Datasets

1) Intrusion detection

We used an intrusion detection dataset[6] containing data logs for nine weeks: seven weeks of the data were used as training data and the other two weeks of data were used as test data. We used source IP (src IP), destination IP (dst IP), source port (src Port), and destination port (dst Port) and used service descriptions such as "http" and "ftp" as the labels for dst Port. We applied our method to this dataset on a per-10-minute basis to predict whether any attacks occurred in each 10-minute period. The model used for prediction was learned separately for each category.

2) P2P lending

P2P lending services[7] enable people to borrow and lend money by matching lenders directly with borrowers. We regarded loans with interest rates higher than 10% as high-risk loans and set the problem as the problem of predicting them by using a set of transactions that includes the relationships between lender, borrower, and lending ID. We used the locations of lenders and borrowers as their labels. We used transactions for 2012, with those before December used as training data and those in December used as evaluation data. The ability to identify high-risk loans without using personal information would expand the possibilities for new FinTech services.

3) Drug discovery

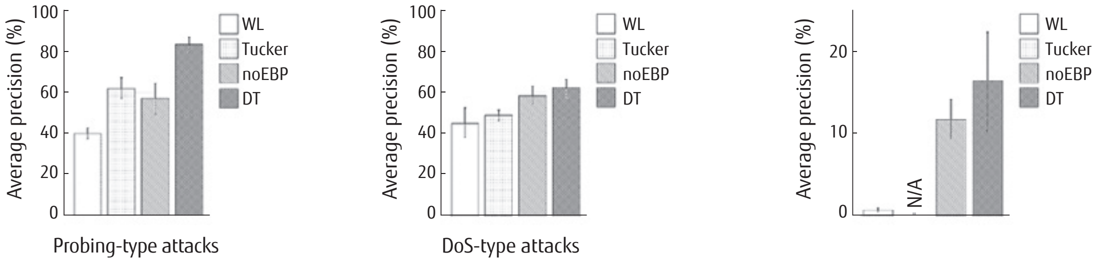
For drug discovery, we set the problem as the problem of predicting the activity or toxicity of chemical compounds by using a graph structure expressing the relationships between atoms labeled by species.[8]

4.2 Compared methods

We compared DT with SVM using several graph kernels, graphlet kernels (GK), shortest path kernels (SP), and Weisfeiler-Lehman subtree kernels (WL). We also compared DT with a method using conventional tensor decomposition (Tucker) instead of SRTD. Furthermore, we evaluated DT without EBP, in which SGD was not used to update the target core tensor (noEBP).

4.3 Prediction accuracy

The results for intrusion detection and P2P lending are shown in **Figure 8**. For intrusion detection, we show the results for probing-type and denial of service (DoS)-type attacks, which had the largest number of log entries among the various types of attack. Note that GK and SP are impractical for these data types because they need enormous calculation time and



(a) Intrusion detection

(b) P2P lending

Figure 8
Classification accuracy for graph data with relationships between more than two elements.
(error bars indicate standard deviation)

memory space due to the huge number of elements and relationships. Moreover, Tucker is impractical for detecting DoS-type attacks for the same reason. We used average precision as the measure of accuracy, i.e., the average precision values (number of true positives divided by number of predicted positives) for the thresholds for the predicted attack and high-risk loan probabilities. The average precisions for both DT and noEBP were better than for the other methods, and the accuracy for DT was higher than for noEBP.

These results suggest that SRTD works better than conventional methods and that EBP is effective in many cases. DT predicted probing-type attacks with practical accuracy (almost 80% of the predicted positives were true positives), whereas more than half of the predicted positives were false positives for WL and Tucker. Although the accuracy of DT for P2P lending might seem to be very low (about 16%), there were about 10 high-risk loans out of more than 1,600 loans in the test data (less than 1%), and DT detected more than half of them.

Evaluation using the datasets for drug discovery were done using 10-fold cross validation. WL achieved the highest accuracy for most of the datasets, suggesting that the Weisfeiler-Lehman graph kernels are well-suited for classifying chemical compounds. The accuracies of DT and noEBP were generally higher than the average accuracies of the graph kernels for all the datasets, suggesting that our method is also effective for chemical compounds.

4.4 Learning large-scale data

We also evaluated DT using data from the PubChem BioAssay database,[9] an open database of chemical compounds. We used the dataset containing the largest number of compounds in this database. The results of 10-fold cross validation showed that our method achieved about 75% accuracy after only 20 training epochs using SGD, whereas the accuracy was about 66% after training using 1,286 compounds for 200 epochs. These results suggest that our method can achieve higher accuracy with very large-scale datasets, as with other deep neural networks.

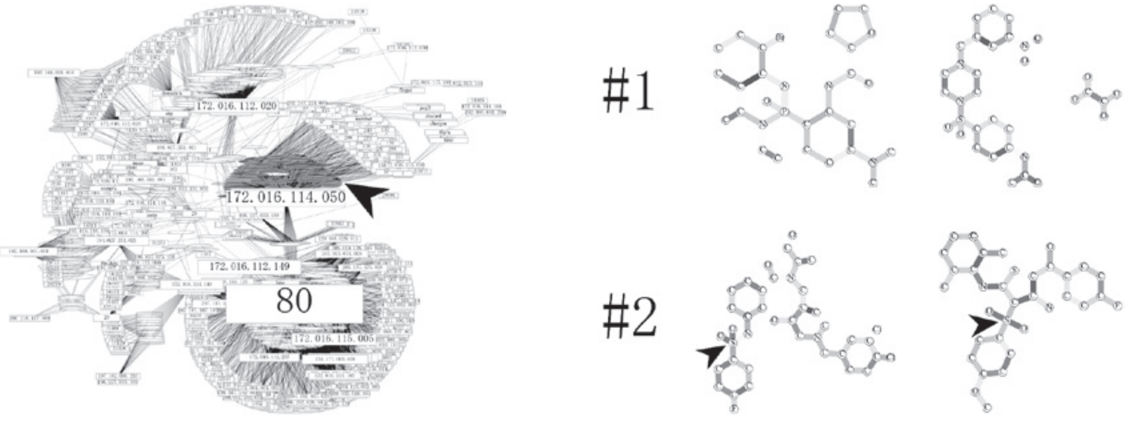
4.5 Interpreting prediction results

We further analyzed the results in order to explain the prediction results along with neuron activity. **Figure 9** depicts the data that had the greatest effect on neurons whose activities are most correlated with the classification probabilities. In the detection of probing-type attacks for intrusion detection, the logs that most contributed to neuron activity [arrowheads in Figure 9 (a)] indicated that one of many HTTP servers was attacked through various ports, and these logs have successfully spotted a port scanning included in the true attacks.

The two compounds with the highest activities for the top two neurons in the dataset taken from the PubChem BioAssay database are shown in Figure 9(b). Of the compounds most affecting neuron 2, the bonds between sulfur (S) and oxygen (O) [arrowheads in Figure 9(b)] greatly contributed to neuron activity, suggesting that these bonds are important for classifying compounds in this dataset.

5. Conclusion

The tensor decomposition method we developed, "Deep Tensor," uses a deep neural network to



(a) Intrusion detection (probing-type attacks)

(b) Drug discovery (PubChem BioAssay)

Figure 9
Graph data with greatest effect on specific neurons.

automatically extract features of graph data. It utilizes a novel tensor decomposition, structure restricted tensor decomposition, to extract the important features of graph data by using a target core tensor. We also developed an extended backpropagation method that optimizes the target core tensor along with the parameters of the neural network. Experiments on three different domains showed that these methods achieve high accuracy, enabling further inspection of the results along with the activity of neurons.

Future work includes evaluation using a wider variety of predictive models, that is, models using larger neural networks, models combined with other models such as convolutional neural networks, and models using core tensors with more complicated structures. Further studies in which these methods are applied to real world problems are needed to expand the possibilities and clarify the limitations of these methods.

These methods will be used as part of FUJITSU Human Centric AI Zinrai, Fujitsu's AI technology.

References

1) N. Shervashidze et al.: Weisfeiler-Lehman Graph Kernels. Journal of Machine Learning Research, Vol. 12, pp. 2539–2561 (2011).

2) A. Krizhevsky et al.: ImageNet Classification with Deep Convolutional Neural Networks. Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12), pp. 1097–1105 (2012).

3) M. Niepert et al.: Learning Convolutional Neural Networks for Graphs. Proceedings of the 33th International Conference on Machine Learning (ICML'16), pp. 2014–2023 (2016).

4) T. G. Kolda et al.: Tensor Decompositions and Applications. SIAM Rev., Vol. 51, No. 3, 455–500 (2009).

5) Y-R. Lin et al.: MetaFac: community discovery via relational hypergraph factorization. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'09), pp. 527–536 (2009).

6) DARPA Intrusion Detection Data Sets.
http://www.ll.mit.edu/ideval/data/

7) Show Me The Money.
http://smtm.labs.theodi.org/

8) ETH zurich.
https://www.bsse.ethz.ch/mlcb/research/machine-learning/graph-kernels/weisfeiler-lehman-graph-kernels.html

9) PubChem BioAssay.
https://www.ncbi.nlm.nih.gov/pcassay

Koji Maruhashi
Fujitsu Laboratories Ltd.
Dr. Maruhashi is currently engaged in research on machine learning, data mining of graph data, and anomaly detection.