A Scheme of Extracting Network Configuration Template Blocks Based on Tucker Decomposition

Daisuke HANAMITSU[†a)], ***Nonmember***, **Kimihiro MIZUTANI**[†,††], ***Member***, **Satoru KOBAYASHI**[†††], **Kensuke FUKUDA**[†††], ***and* Osamu AKASHI**[†††], ***Nonmembers***

1. Introduction

Recently, the size of communication network is growing significantly because of promoting the use of smart devices (e.g., smart phone). Configuring such large scale communication network, an operator needs to input huge amount of network commands. Using a configuration template block of a setting scenario (e.g., set up VLAN configuration) makes the network operator easier to reflect the network settings, however, it takes much time to make the template blocks (i.e., a contextual group of commands) manually for more complicated and expanded networks. To address the problem, we propose an autonomic template block extraction scheme by using Tucker decomposition [1]. Our proposal can extract the configuration template blocks from a large number of configuration automatically.

2. Proposed Scheme

The proposed configuration template extraction based on Tucker decomposition consists of following steps. First, we replace the numeric expression (e.g., IP address) contained in each line of network configuration files with '*', and define it as a configuration template. Second, we replace the network configuration to an input matrix for Tucker decomposition. Let $\mathrm{C}_1, \mathrm{C}_2, \cdots, \mathrm{C}_S$ be bodies of each configuration file on S files and $\mathrm{e}_i \in \mathrm{E}(1 \leq \mathrm{i} \leq |\mathrm{E}|)$ be a configuration template noting that $|E|$ is the number of kinds of templates. Here, we divide $\mathrm{C}_s(1 \leq \mathrm{s} \leq \mathrm{S})$ into M parts, and count the number of lines corresponded to template e_i in the divided configuration. We express the union of the counted numbers in $C_{s,j}(1 \leq j \leq M)$ as a vector $\boldsymbol{C}_{s,j}$, and build a feature matrix $\boldsymbol{C_s}$ with the M vectors where $C_{s,j}$ denotes a divided part of C_s. Notes that $\boldsymbol{C_s}$ becomes a two-dimensional matrix and its size is $M \times |E|$. Finally, we build a three-dimensional matrix by combining S files' feature matrices, and input it to Tucker decomposition noting that the three-dimensional matrix size is $M \times |E| \times S$. Tucker decomposition can split the input matrix to $\boldsymbol{Q}(M \times F)$, $\boldsymbol{W}(|E| \times G)$ and $\boldsymbol{H}(S \times I)$ matrices where F, G, and I indicate pre-determined latent variables against each input dimension respectively. For extracting configuration template blocks of the input matrix, we focus on the feature matrix $\boldsymbol{W}(|E| \times G)$.

[†]Department of Informatics, Kindai University 3-4-1 Kowakae, Higashi Osaka, Osaka, Japan

[††]Cyber Informatics Research Institute, Kindai University, Same location as Kindai University.

[†††]National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo.

a) E-mail: 1810370110h@kindai.ac.jp

Suppose $\boldsymbol{W}(e, g)(1 \leq e \leq |E|, 1 \leq g \leq G)$ exceeds certain threshold (e.g., 0.5) where $\boldsymbol{W}(e, g)$ is a value of the matrix of $\boldsymbol{W}(|E| \times G)$. Then, we can recognize a template numbered e belongs a group g, and the union of templates grouped to g can be extracted as a templates block.

3. Evaluation

Here, we demonstrate extracting configuration template blocks with real network configuration which consists of approximately 1,500 command lines and 130 configuration templates on 16 files [2]. We empirically set all parameters (i.e., M, F, G, I) to 5. Fig 1 shows one example of the result (Our Proposal) with manually annotated ground truth (Expected). From the result, our proposal could extract configuration template blocks as same as the ground truth.

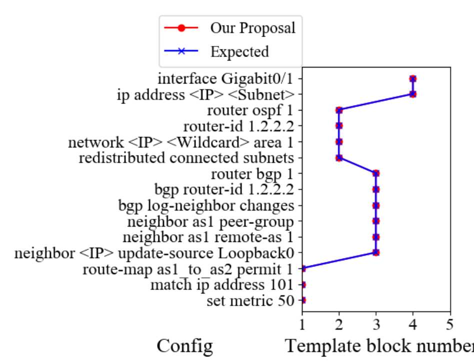

Fig. 1 Result of extracting configuration template blocks

4. Conclusion

In this paper, we proposed a novel scheme of extracting template blocks of network configuration based on Tucker decomposition. In the evaluation, we found out that our proposal could extract the template blocks accurately. In future work, we will tune our proposal with a large number of network configuration and examine the extraction accuracy.

Acknowledgement

This work is partially supported by JSPS Kakenhi 20H04185.

References

[1] Tamara G. Kolda et al, "Tensor Decomposition and Applications". SIAM Review, vol. 51, pp. 455-500, 2009.

[2] Batfish. https://github.com/batfish/batfish.