Multilevel–Multigroup Analysis Using a Hierarchical Tensor SOM Network

Hideaki Ishibashi[1], Ryota Shinriki[1], Hirohisa Isogai[1], and Tetsuo Furukawa[1]

Department of Brain Science and Engineering,
Kyushu Institute of Technology
{ishibashi-hideaki,shinriki-ryota}@edu.brain.kyutech.ac.jp
{isogai,furukawa}@brain.kyutech.ac.jp

Abstract. This paper describes a method of multilevel–multigroup analysis based on a nonlinear multiway dimensionality reduction. To analyze a set of groups in terms of the probabilistic distribution of their constituent member data, the proposed method uses a hierarchical pair of tensor self-organizing maps (TSOMs), one for the member analysis and the other for the group analysis. This architecture enables more flexible analysis than ordinary parametric multilevel analysis, as it retains a high level of translatability supported by strong visualization. Furthermore, this architecture provides a consistent and seamless computation method for multilevel–multigroup analysis by integrating two different levels into a hierarchical tensor SOM network. The proposed method is applied to a dataset of football teams in a university league, and successfully visualizes the types of players that constitute each team as well as the differences or similarities between the teams.

Keywords: Self-organizing map, Multilevel–multigroup analysis, Tensor SOM

1 Introduction

The aim of this study is to develop a multilevel-multigroup analysis method based on a nonlinear multiway dimensionality reduction. The task is to analyze a set of groups in terms of the probability distributions of their constituent member data. A typical example is the analysis of sports teams belonging to the same league, in which the task is to visualize how each team is different or similar to other teams by comparing the statistical data of their members. In addition, we may like to visualize what types of players constitute those teams. A hierarchical analysis is required to realize this task, with the lower (member) level modeling the data distribution of each group and the upper (group) level analyzing the obtained distribution set.

The simplest method of multigroup analysis is to use the mean value of the constituent member data. In this case, the individual data are averaged over the members for each group to form the feature vector of the group [1] (although another parameter is sometimes used for the feature vectors [2]). Generally, a

This is a post-peer-review, pre-copyedit version of an article published in Lecture Notes in Computer Science. The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-319-46675-0_50.

parametric model can be employed to represent the data distribution of each group, and the obtained parameters are regarded as the feature vectors of the groups. This approach is convenient in that the employed parametric model sufficiently captures the data distributions. However, if the model is not adequate, the parametric approach produces false results. For example, if the members of a group form some distinct clusters, then the average only represents the intermediate point between clusters, and there may not be any members around the mean point.

The alternative is to use a nonparametric representation such as a histogram. In this case, the obtained histograms are regarded as the feature vectors of the groups. This provides a more flexible representation than the parametric approach, but is only available for low-dimensional cases. To overcome this limitation, a combination of dimensionality reduction and nonparametric modeling is required. Thus, high-dimensional data are mapped to a low-dimensional latent space in advance, and the data distributions in the latent space are estimated by a nonparametric model [3]. This approach is promising, but dimensionality reduction sacrifices the translatability of the results, especially in nonlinear cases.

In this paper, we introduce a nonlinear multiway dimensionality reduction based on the tensor self-organizing map (TSOM). The TSOM is a visualization tool for relational datasets that produces two (or more) maps [4]. Though designed to analyze relational datasets, the TSOM can also visualize ordinary high-dimensional datasets by regarding them as relational ones. In this case, the TSOM produces a map of the target objects and a map of the data components, with the latter visualizing the intrinsic factors underlying the data. Thus, the TSOM enhances the translatability of high-dimensional datasets. By introducing the TSOM to multilevel–multigroup analysis, we achieve both a flexible representation of the member distributions and a translatable result.

Furthermore, the TSOM can be employed to visualize the groups, as the member–group affiliations can be represented by relational data. As a result, the entire architecture becomes a hierarchical network of two TSOMs, one for the member-level analysis and the other for the group-level analysis. Using the TSOM network, multilevel–multigroup analysis can be executed in a consistent and seamless manner.

The remainder of this paper is organized as follows. Section 2 formulates the task, before the algorithm is described in Sec. 3. In Sec. 4, this method is applied to analyze the football teams in a university league. The final section contains the conclusions to this study.

2 Problem formulation

Suppose that we survey I subjects (members) with J queries. The entire dataset can be represented by a matrix $\mathbf{X} = (x_{ij}) \in \mathbb{R}^{I \times J}$. Suppose that there are K groups, and each member belongs to at least one of the groups. (Duplication is allowed.) This member–group affiliation is represented by a matrix $\mathbf{Y} = (y_{ik}) \in$

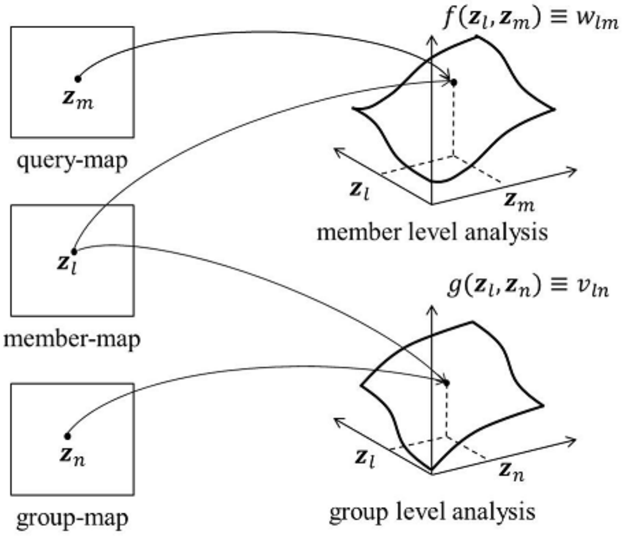


Fig. 1. Architecture of the proposed method.

$\{0,1\}^{I \times K}$, where $y_{ik} = 1$ if the ith member belongs to the kth group, and otherwise $y_{ik} = 0$.

The aim of the proposed method is to visualize the relationships between the members, the groups, and the queries by mapping them to low-dimensional latent spaces via *member-maps*, *query-maps*, and *group-maps*, respectively. Thus, our task is to estimate the three latent variable sets $\{\mathbf{z}_i^{\text{member}}\}$, $\{\mathbf{z}_j^{\text{query}}\}$, $\{\mathbf{z}_k^{\text{group}}\}$. This task consists of two subtasks, corresponding to the member-level analysis and the group-level analysis.

In the member-level analysis, the subtask is to estimate the latent variables $\left\{\mathbf{z}_i^{(\text{member})}\right\}$, $\left\{\mathbf{z}_j^{(\text{query})}\right\}$, and a smooth nonlinear map f, so that the observed data is approximated as

$$x_{ij} \simeq f(\mathbf{z}_i^{(\text{member})}, \mathbf{z}_j^{(\text{query})}). \tag{1}$$

In the group-level analysis, the subtask is to estimate the latent variable $\left\{\mathbf{z}_k^{(\text{group})}\right\}$ and a nonlinear smooth map g, so that the member distribution of the kth group is represented as

$$p\left(\mathbf{z}^{(\text{member})} \,\middle|\, \mathbf{z}_k^{(\text{group})}\right) = g(\mathbf{z}^{(\text{member})}, \mathbf{z}_k^{(\text{group})}). \tag{2}$$

3 Architecture and Algorithm

3.1 Architecture

In an ordinary SOM, the latent space is discretized to grid nodes, and a reference vector is assigned to every node. In the case of the TSOM of order N, the TSOM has N latent spaces that are discretized to grid nodes, as in the ordinary SOM. However, unlike the conventional SOM, the reference vectors of the TSOM are assigned to all combinations of L nodes. Thus, if the latent spaces are discretized to L nodes, there are L^N reference vectors.

The proposed architecture for multilevel-multigroup analysis consists of two TSOMs of order 2, one for the member-level and the other for the group-level (Fig. 1). Suppose that $\mathbf{z}_l^{(\text{member})}$, $\mathbf{z}_m^{(\text{query})}$, $\mathbf{z}_n^{(\text{group})}$ are the positional vectors in the latent spaces of the discretized nodes l, m, n. In the TSOM for the member-level analysis, reference vectors are assigned to all combinations of $\left\{(\mathbf{z}_l^{(\text{member})}, \mathbf{z}_m^{(\text{query})})\right\}$ so that $w_{lm} \equiv f(\mathbf{z}_l^{(\text{member})}, \mathbf{z}_m^{(\text{query})})$. Then, the entire set of reference vectors becomes a matrix $\mathbf{W} \equiv (w_{lm}) \in \mathbb{R}^{L\times M}$. The row vectors $\mathbf{w}_l^{(\text{member})} \equiv (w_{l1}, \ldots, w_{lM})$ and the column vectors $\mathbf{w}_m^{(\text{query})} \equiv (w_{1m}, \ldots, w_{Lm})$ act as the conventional reference vectors for members and for queries, respectively.

Similarly, reference vectors in the group-level TSOM are assigned to all combinations of $\left\{(\mathbf{z}_l^{(\text{member})}, \mathbf{z}_n^{(\text{group})})\right\}$ so that $v_{ln} \equiv g(\mathbf{z}_l^{(\text{member})}, \mathbf{z}_n^{(\text{group})})$. Thus, the entire set becomes a matrix $\mathbf{V} = (v_{ln}) \in \mathbb{R}^{L\times N}$. In this case, the column vector $\mathbf{v}_n^{(\text{group})} \equiv (v_{1n}, \ldots, v_{Ln})$ acts like a reference vector for groups.

Besides these reference vectors, the algorithm includes the vectors $\mathbf{u}_i^{(\text{member})} \in \mathbb{R}^L$, $\mathbf{u}_j^{(\text{query})} \in \mathbb{R}^M$, and $\mathbf{u}_k^{(\text{group})} \in \mathbb{R}^N$. These play the role of data vectors for members, queries, and groups, and are calculated iteratively in the algorithm.

3.2 TSOM for member-level analysis

The proposed algorithm consists of two stages: the member-level TSOM learns first, followed by the group-level TSOM. The learning algorithm for the TSOM is the expectation-maximization (EM) algorithm in a broad sense, with the E and M steps iterating alternately.

The learning algorithm of the member-level TSOM can be described as follows.

E Step At calculation time t, the best matching nodes are determined from the result of the M step at $t-1$.

$$l_i^\star = \arg\min_l \|\mathbf{u}_i^{(\text{member})} - \mathbf{w}_l^{(\text{member})}\|^2 \tag{3}$$

$$m_j^\star = \arg\min_m \|\mathbf{u}_j^{(\text{query})} - \mathbf{w}_m^{(\text{query})}\|^2. \tag{4}$$

(For the first loop, the best matching nodes are assigned randomly.) The latent variables are then estimated as $\mathbf{z}_i^{(\text{member})} = \mathbf{z}_{l_i^\star}^{(\text{member})}$, $\mathbf{z}_j^{(\text{query})} = \mathbf{z}_{m_j^\star}^{(\text{query})}$.

M-Step $\mathbf{W}$ is calculated using a kernel smoother:

$$w_{lm} = \sum_{i=1}^{I} \sum_{j=1}^{J} R_{li}^{(\text{member})} R_{mj}^{(\text{query})} x_{ij}, \tag{5}$$

where $R_{li}^{(\text{member})}$ and $R_{mj}^{(\text{query})}$ are the neighborhood functions normalized with respect to data points *s.t.* $\sum_i R_{li}^{(\text{member})} = 1$ and $\sum_j R_{mj}^{(\text{query})} = 1$. In this paper, we use $R_{li} \propto \exp\left[\|\mathbf{z}_l - \mathbf{z}_i\|^2 / 2\sigma^2\right]$. After updating $\mathbf{W}$, $\{\mathbf{u}^{(\text{member})}\}$ and $\{\mathbf{u}^{(\text{query})}\}$ are also updated by

$$u_{im}^{(\text{member})} = \sum_{j=1}^{J} R_{mj}^{(\text{query})} x_{ij} \tag{6}$$

$$u_{jl}^{(\text{query})} = \sum_{i=1}^{I} R_{li}^{(\text{member})} x_{ij}. \tag{7}$$

These E and M steps are iterated until the TSOM reaches the steady state with a reduced neighborhood size.

3.3 TSOM for group-level analysis

Once the first stage is finished, the group-level TSOM is trained.

Initialization Before starting the process, $\{\mathbf{u}_k^{(\text{group})}\}$ are calculated in advance as:

$$u_{kl}^{(\text{group})} = \sum_{i=1}^{I} \tilde{R}_{li}^{(\text{member})} \tilde{y}_{ik}, \tag{8}$$

where $\tilde{y}_{ik} \equiv y_{ik}/Y_k$ and $Y_k \equiv \sum_i y_{ik}$. $\tilde{R}_{li}^{(\text{member})}$ is the neighborhood function normalized with respect to the latent space *s.t.* $\sum_l \tilde{R}_{li}^{(\text{member})} = 1$. Note that $\mathbf{u}_k^{(\text{group})}$ represents the conditional probability $p(\mathbf{z}^{(\text{member})}|y_{ik} = 1)$, which means the member distribution of the kth group.

E Step The E step determines the best matching nodes. As $\mathbf{u}_k^{(\text{group})}$ and $\mathbf{v}_n^{(\text{group})}$ represent conditional probabilities, the errors are evaluated by the cross entropy

$$n_k^{\star} = \arg\max_n \sum_{n=1}^{N} u_{kl}^{(\text{group})} \ln v_{ln}. \tag{9}$$

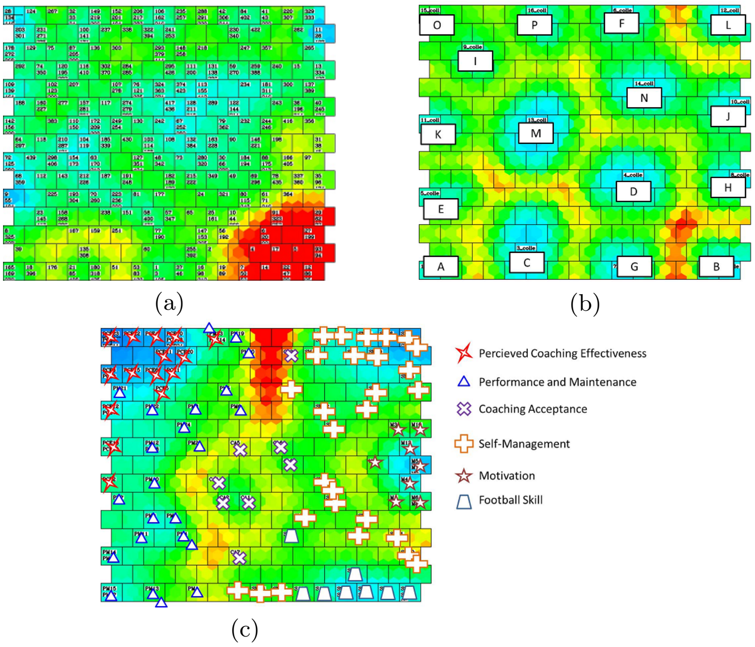
(For the first loop, the best matching nodes are assigned randomly.) The latent variables are then estimated as $\mathbf{z}_k^{(\text{group})} = \mathbf{z}_{n_k^{\star}}^{(\text{group})}$.

M Step In the M step, the reference vector $\mathbf{V}$ is updated by

$$\mathbf{v}_n^{(\mathrm{group})} = \sum_{k=1}^{K} \tilde{R}_{nk}^{(\mathrm{member})} \mathbf{u}_k^{(\mathrm{group})}. \tag{10}$$

Note that $\mathbf{V}$ represents the conditional probability $p(\mathbf{z}^{(\mathrm{member})}|\mathbf{z}^{(\mathrm{group})})$.

These E and M steps are iterated alternately until the group map converges.



(a) (b)

(c)

Fig. 2. The maps generated by our algorithm. (a) The member (player) map. (b) The group (team) map. (c) The query map. The color of these maps is represented by U-matrix.

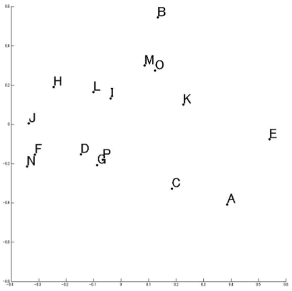


Fig. 3. Result of the team analysis by PCA. The average score of the members is used as the feature vector of each team.

4 Experiments

The proposed method was applied to a questionnaire survey from a university football league in Japan. The survey was taken for the 439 players of 16 universities. The questionnaire consisted of 112 queries asking how players assess themselves and their coaches. The queries were categorized as six types: (1) motivation, (2) self-management skill, (3) football skill, (4) coach acceptance, (5) performance and maintenance (PM) of coaching, and (6) perceived coaching effectiveness (PCE). Questions (1)–(3) concern self-assessment, and (4)–(6) concern the assessments of the coach by the players. The data were standardized in advance so that the mean and variance were zero and one, respectively.

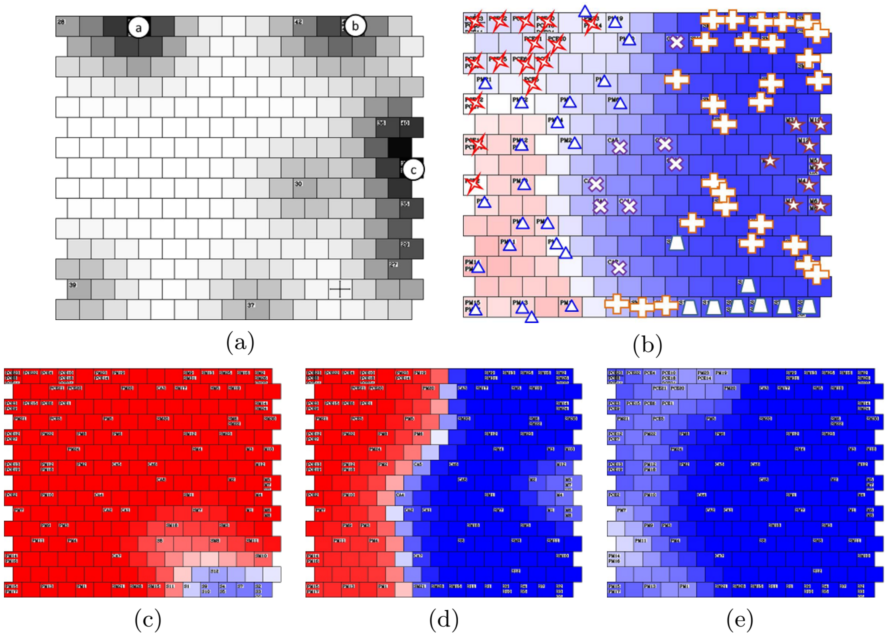
The results are shown in Fig. 2. In the query map (c), the six categories are separated, suggesting the result is plausible. To compare with the conventional method, the averaged team scores were analyzed by principal component analysis (PCA). The results are shown in Fig. 3. Compared with Fig. 2 (b), the arrangement of the teams is similar, but some teams (such as B and O) are located differently. Fig. 4 and 5 show the constitution of teams B and O, indicating that both consist of three clusters of players (Fig. 4a, 5a). The averaged score distributions are almost equal in the query maps (Fig. 4b, 5b), but the score distributions of each cluster are quite different (Fig. 4c–e, 5c–e). The proposed method clearly discriminates between such differences. In addition, it is easy to see the properties of clusters intuitively using the interactive graphical interface of TSOM. These are the major advantages of using TSOM in such multilevel–multigroup analysis.

5 Conclusion

In this paper, we proposed a multilevel–multigroup analysis method using a hierarchical TSOM network. This method provides an effective analysis tool with intuitive visualization and an interactive interface. Thus, this approach represents a powerful tool for both multigroup analysis and for hierarchical data in general.

References

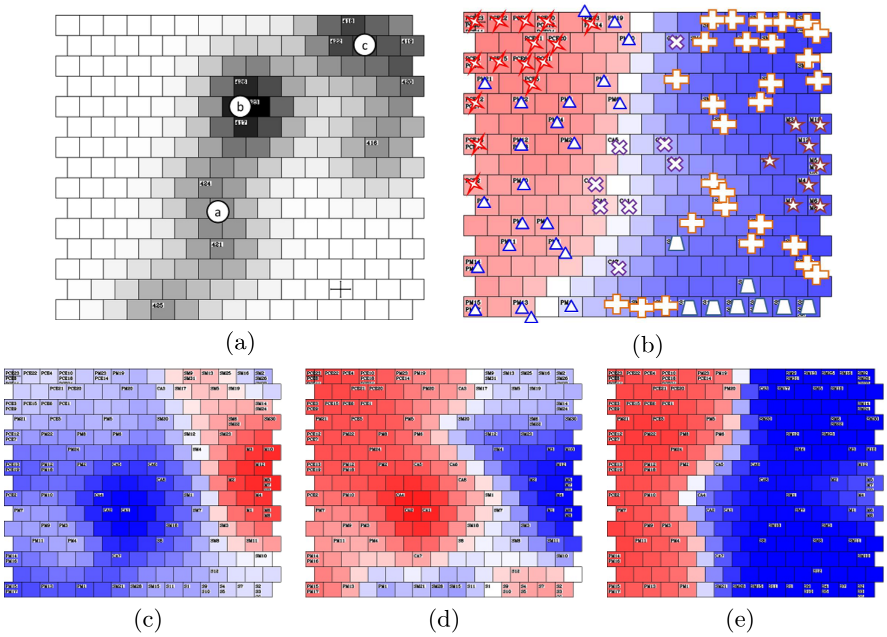
1. Timmerman, M.E.: Multilevel component analysis, British Journal of Mathematical and Statistical Psychology, 59, 301–320, (2006)
2. Eslami, A., Qannari, E.M., Kohler, A., Bougeard, S.: General overview of methods of analysis of multi-group datasets, RNTI 25, 108–123, (2013)
3. Friedman, J.H.: Exploratory projection pursuit, Journal of the American Statistical Association, 82, 259–266, (1987)
4. Iwasaki, T., Furukawa, T.: Tensor SOM and Tensor GTM: Nonlinear Tensor Analysis by Topographic Mappings, Neural Networks, 77, 107–125 (2016)



(a) (b)

(c) (d) (e)

Fig. 4. The player distribution of team B in Fig. 2b. (a) The distribution of players from team B in the member map. (b) The averaged score of the members in the query map. (c)–(e) The scores of clusters a, b, c in the query map.



(a) (b)

(c) (d) (e)

Fig. 5. The player distribution of team O in Fig. 2b. (b) The averaged score of the members in the query map. (c)–(e) The scores of clusters a, b, c in the query map.