Multi-task manifold learning using hierarchical modeling for insufficient samples

Hideaki Ishibashi[1], Kazushi Higa[1], and Tetsuo Furukawa[1]

Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan

Abstract. In this paper, we propose a method for multi-task manifold learning. For a set of tasks of dimensionality reduction, the aim of the method is to model each given dataset as a manifold, and map it to a low-dimensional space. For this purpose, we use a hierarchical manifold modeling approach. Thus, while each data distribution is represented by a manifold model, the obtained models are further modeled by a higher-order manifold in a function space. The higher-order model mediates the information transfer between tasks, and as a result, the performance of each task is improved. The results of simulations show that the proposed method can estimate manifolds approximately, even in cases in which a tiny number of samples are provided for each task.

Keywords: Multi-task learning · Multi-task unsupervised learning · Manifold learning · Hierarchical modeling · Multi-level modeling.

1 Introduction

Multi-task learning is a paradigm of machine learning that aims to improve performance by simultaneously learning similar tasks [2, 28]. Many studies have been conducted on multi-task learning, particularly supervised learning. By contrast, there have been few studies on multi-task unsupervised learning, and only a few studies have been conducted on multi-task clustering [28]. To date, few works have been reported on dimensionality reduction, particularly in the context of non-linear manifold learning. The purpose of this study is to develop a method for multi-task manifold learning. We focus in particular on scenarios in which the number of data samples is too small to estimate manifolds and the assistance of other tasks is indispensable.

A typical example is face image modeling. It is well known that face images are modeled by a manifold [23, 3]. To estimate a face manifold, we typically need a sufficient number of photographs taken from various viewpoints with various expressions that cover the manifold entirely. However, in practice, it is typically difficult to obtain such an exhaustive image set of a single person. Instead, we typically have a huge number of photographs of other people. Thus, we have many image sets of various people, each of which consists of a small number (i.e., insufficient number) of photographs. In such a scenario, our aim is to improve modeling performance by transferring the information between tasks.

To achieve the above, we use a hierarchical modeling approach in this study. Thus, while each given dataset is modeled by a manifold, the manifold models are further modeled by a higher-order manifold in a function space. This higher-order model mediates information transfer among the given tasks, thereby improving the performance of manifold modeling. The proposed method consists of hierarchically coupled manifold models based on the kernel smoother (kernel-smoother-based manifold modeling: KSMM), referred to as the hierarchical KSMM (H-KSMM).

The remainder of the paper is structured as follows: The problem is formulated in Section 2, and related work is introduced in Section 3. The proposed method is presented in Section 4 and experiment results to verify it are described in Section 5. A discussion of the results and the conclusions of this paper are provided in the final section.

2 Problem formulation

Suppose we have I tasks. Thus, we have I datasets $\{\mathcal{S}_1, \ldots, \mathcal{S}_I\}$ in high-dimensional space $\mathcal{X} = \mathbb{R}^{D_\mathcal{X}}$, each of which consists of N_i samples. The entire dataset is denoted as $\mathcal{S} = \bigcup_i \mathcal{S}_i = \{\mathbf{x}_n\}_{n=1}^N$, where $N = \sum_i N_i$. We also describe the entire dataset using matrix $\mathbf{X} = (\mathbf{x}_n^\mathrm{T}) \in \mathbb{R}^{N \times D_\mathcal{X}}$. Furthermore, let i_n be the task index of sample n and $\mathcal{N}_i$ the index set of samples that belong to task i.

When such datasets are provided, our first aim is to map the data to low-dimensional space $\mathcal{Z} = \mathbb{R}^{D_\mathcal{Z}}$. Thus, the first aim is to estimate $\{\mathbf{z}_n\}$ that corresponds to $\{\mathbf{x}_n\}$. Our second aim is to model each data distribution using a nonlinear manifold. Thus, for the ith dataset, the method models $\mathbf{x} \,|\, \mathbf{z} \sim \mathcal{N}(f_i(\mathbf{z}), \beta^{-1}\mathbf{I})$, where $f_i : \mathcal{Z} \to \mathcal{X}$ is a smooth embedding from $\mathcal{Z}$ to $\mathcal{X}$. Then, the image of f_i becomes a nonlinear manifold $\mathcal{M}_i = f_i(\mathcal{Z})$ in $\mathcal{X}$. In this work, f_i is referred to as the '*task model*.' Note that $\{f_i\}$ belongs to the same function space $\mathcal{F}$, because $\mathcal{X}$ and $\mathcal{Z}$ are common to all tasks in this paper.

To achieve the above aims, the following hierarchical model is assumed in this work. Suppose that $\mathcal{Y}$ is another low-dimensional space for task sets, and all task models $\{f_i\}$ are assigned to $\{\mathbf{y}_i\}$ as low-dimensional representations. Suppose further that $g : \mathcal{Y} \to \mathcal{F}$ is a smooth embedding that satisfies $f_i = g[\mathbf{y}_i]$. Thus, the task models are further modeled by manifold $\mathcal{L} = g[\mathcal{Y}]$ in function space $\mathcal{F}$. Then, all datasets are modeled as $\mathbf{x} \,|\, \mathbf{z}, \mathbf{y} \sim \mathcal{N}(F(\mathbf{z}, \mathbf{y}), \beta^{-1}\mathbf{I})$, where $F : \mathcal{Z} \times \mathcal{Y} \to \mathcal{X} : (\mathbf{z}, \mathbf{y}) \mapsto (g[\mathbf{y}])(\mathbf{z})$. In this paper, F is referred to as a '*general model*.' Under these assumptions, the aim of multi-task manifold learning is then to estimate $\{\mathbf{z}_n\}$, $\{\mathbf{y}_i\}$, and F simultaneously.

3 Related work

To date, few studies have reported multi-task learning in the context of dimensionality reduction tasks, subspace methods, and manifold learning. To the best of our knowledge obtained from a survey, multi-task principal component analysis is the only development in the literature that is expressly aimed at the multi-task learning of subspace methods [27]. However, by extending the scope of our survey, we can locate related methods in the field of hierarchical modeling (or multi-level modeling) that aim to obtain higher-order models of tasks [5]. Although hierarchical modeling does not aim to improve the performance of tasks, the areas of hierarchical modeling and multi-task learning overlap, where the former is sometimes used as an approach to the latter [29, 9, 10].

Among methods for unsupervised hierarchical modeling, the higher rank of self-organizing maps (SOM^2) is the most relevant work to this study [7, 6]. SOM^2 has been applied to several problems in multi-task learning, such as face images of various people [14], nonlinear dynamical systems with latent state variables [21, 22], the shapes of various objects [25, 26], and members of various groups [12, 13]. In this sense, SOM^2 is one of the earliest examinations of multi-task unsupervised learning for nonlinear subspaces.

Although SOM^2 works like a multi-task learning method, it remains challenging to estimate manifolds when the number of samples per task is small. Moreover, SOM^2 has several limitations that originate from SOM itself, such as poor manifold representation using discretized nodes and the brute force optimization of latent variables. In this paper, we attempt to eliminate such limitations from SOM^2 by replacing it with KSMM and extending it for the multi-task learning paradigm.

4 Proposed method

KSMM is used as the building block of hierarchical manifold modeling in the proposed method. In this section, we first describe KSMM and introduce the proposed method, called H-KSMM.

4.1 Kernel-smoother-based manifold modeling (KSMM)

Generally, nonlinear methods for dimensionality reduction are categorized into two groups [17]. The first consists of methods that project data points *from* a high-dimensional space (data space) *to* a low-dimensional space (visualization space). Most dimensionality reduction methods are in this group. By contrast, the second group consists of methods that estimate the mapping *from* a low-dimensional space (latent space) *to* a high-dimensional space (visible space). As the latter group of methods aim to model the data distribution using a manifold, we refer to the group as *manifold modeling*. Representative methods of manifold modeling are generative topographic mapping (GTM) [1] and the Gaussian process latent variable model (GPLVM) [18, 17], which originate from self-organizing maps (SOMs) [16]. To estimate a smooth manifold, GTM and GPLVM use the Gaussian process, whereas SOM uses a kernel smoother.

KSMM uses a kernel smoother, such as the original SOM, instead of a Gaussian process because this makes it easier to extend SOM^2 to H-KSMM. Moreover, to the best of our knowledge, the kernel smoother stabilizes manifold modeling to a greater extent than the Gaussian process, particularly in challenging conditions, such as the case that we consider.

Although not by this particular name, KSMM has been proposed in many studies as a theoretical generalization of SOM [20, 4, 8, 11, 24]. According to these studies, the cost function of KSMM is given by

$$E = \frac{\beta}{2} \sum_{n} \int h(\mathbf{z}, \mathbf{z}_n) \, \|\mathbf{x}_n - f(\mathbf{z})\|^2 \; p(\mathbf{z}) \, d\mathbf{z}. \tag{1}$$

In (1), $h(\mathbf{z}, \mathbf{z}')$ is a non-negative smoothing kernel defined on $\mathcal{Z}$, which is typically $h(\mathbf{z}, \mathbf{z}') = \mathcal{N}(\mathbf{z} \mid \mathbf{z}', \lambda_{\mathcal{Z}}^2 \mathbf{I})$. The prior of $\mathbf{z}$ is a uniform distribution on a unit square space, that is, $p(\mathbf{z}) = 1$ for $\mathbf{z} \in [-1/2, +1/2]^{D_{\mathcal{Z}}}$; otherwise, $p(\mathbf{z}) = 0$. In this study, nonlinear mapping f is represented parametrically using orthonormal basis functions (e.g., normalized Legendre polynomials). Thus, $f(\mathbf{z} \mid \mathbf{V}) = \mathbf{V}^{\mathrm{T}} \boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{z})$, where $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_L)^{\mathrm{T}}$ is the basis set and $\mathbf{V} \in \mathbb{R}^{L \times D_{\mathcal{X}}}$ is the coefficient matrix.

Nonlinear mapping f and latent variables $\{\mathbf{z}_n\}$ are alternately updated, as in a generalized expectation maximization algorithm. To update f, coefficient matrix $\mathbf{V}$ is calculated as $\mathbf{V} = \mathbf{A}^{-1}\mathbf{B}\mathbf{X}$, where

$$\mathbf{A} = \int \boldsymbol{\varphi}(\mathbf{z})\, \boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{z})\, \overline{h}(\mathbf{z})\, p(\mathbf{z})\, d\mathbf{z} \tag{2}$$

$$\mathbf{B} = \int \boldsymbol{\varphi}(\mathbf{z})\, \mathbf{h}(\mathbf{z})^{\mathrm{T}}\, p(\mathbf{z})\, d\mathbf{z}, \tag{3}$$

where $\mathbf{h}(\mathbf{z}) = (h(\mathbf{z}, \mathbf{z}_1), \ldots, h(\mathbf{z}, \mathbf{z}_N))^{\mathrm{T}}$ and $\overline{h}(\mathbf{z}) = \sum_n h(\mathbf{z}, \mathbf{z}_n)$. By contrast, $\{\mathbf{z}_n\}$ are updated using a gradient method so that the value of the objective function (1) is reduced.

4.2 Hierarchical KSMM (H-KSMM)

H-KSMM consists of two hierarchically coupled KSMMs: a lower-KSMM and higher-KSMM. The lower KSMM estimates each task model, whereas the higher-KSMM estimates the general model.

In H-KSMM, task information is transferred in two ways. The first involves forming a *weighted mixture of the sample datasets*. If task i' is a neighbor of task i in latent space $\mathcal{Y}$, then sample set $\mathcal{S}_{i'}$ is merged into target set $\mathcal{S}_i$ as an auxiliary sample set with a larger weight. By contrast, if task i'' is far from task i in $\mathcal{Y}$, then $\mathcal{S}_{i''}$ is merged into $\mathcal{S}_i$ with a small (or zero) weight. Let us denote the weight of sample n of task i_n with respect to target task i as ρ_{in} $(0 \le \rho_{in} \le 1)$. Typically, $\rho_{in} \equiv \rho(\mathbf{y}_i, \mathbf{y}_{i_n}) = \exp\left[-\frac{1}{2\lambda_\rho^2} \left\|\mathbf{y}_i - \mathbf{y}_{i_n}\right\|^2\right]$, where λ_ρ determines the size of the neighborhood for data mixing. By contrast, the second way of transferring task information involves forming a *weighted mixture of the task models* among neighboring tasks, that is, the kernel smoothing of the task models.

The H-KSMM algorithm is as follows:

Step 1: Suppose $\{\mathbf{z}_n\}$ and $\{\mathbf{y}_i\}$ have been estimated in a preceding calculation loop (or initialized randomly in the first loop). In Step 1, ρ_{in} is calculated as described above.

Step 2: To obtain task models $\{f_i\}$, corresponding coefficient matrices $\{\mathbf{V}_i\}$ are calculated by $\mathbf{V}_i = \mathbf{A}_i^{-1}\mathbf{B}_i\mathbf{X}$, where

$$\mathbf{A}_i = \int \boldsymbol{\varphi}(\mathbf{z})\, \boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{z})\, \overline{h}_i(\mathbf{z})\, p(\mathbf{z})\, d\mathbf{z} \tag{4}$$

$$\mathbf{B}_i = \int \boldsymbol{\varphi}(\mathbf{z})\, \mathbf{h}_i(\mathbf{z})^{\mathrm{T}}\, p(\mathbf{z})\, d\mathbf{z}. \tag{5}$$

In (4) (5), $\mathbf{h}_i(\mathbf{z}) = (\rho_{i1} h(\mathbf{z}, \mathbf{z}_1), \ldots, \rho_{iN} h(\mathbf{z}, \mathbf{z}_N))^{\mathrm{T}}$, and $\overline{h}_i(\mathbf{z}) = \sum_n \rho_{in}\, h(\mathbf{z}, \mathbf{z}_n)$. The coefficient matrices are collectively expressed as third-order tensor $\underline{\mathbf{V}} = (\mathbf{V}_i) \in \mathbb{R}^{I \times L \times D_{\mathcal{X}}}$.

Step 3: To obtain general model F, coefficient tensor $\underline{\mathbf{W}} \in \mathbb{R}^{I\times L\times D_{\mathcal{X}}}$ is calculated by

$$\underline{\mathbf{W}} = \underline{\mathbf{V}} \times_1 \left(\mathbf{C}^{-1}\mathbf{D}\right) \tag{6}$$

$$\mathbf{C} = \int \boldsymbol{\psi}(\mathbf{y})\, \boldsymbol{\psi}^{\mathrm{T}}(\mathbf{y})\, \bar{k}(\mathbf{y})\, p(\mathbf{y})\, d\mathbf{y} \tag{7}$$

$$\mathbf{D} = \int \boldsymbol{\psi}(\mathbf{y})\, \mathbf{k}^{\mathrm{T}}(\mathbf{y})\, p(\mathbf{y})\, d\mathbf{y}, \tag{8}$$

where $\mathbf{k}(\mathbf{y}) = (k(\mathbf{y}, \mathbf{z}_1), \ldots, k(\mathbf{y}, \mathbf{y}_I))^{\mathrm{T}}$, $\bar{k}(\mathbf{y}) = \sum_i k(\mathbf{y}, \mathbf{y}_i)$, and $k(\mathbf{y}, \mathbf{y}')$ and $\boldsymbol{\psi}(\mathbf{y})$ are the smoothing kernel and basis functions for the higher-KSMM, respectively. Symbol $\times_m$ denotes the tensor matrix product of the mth mode. Then, the general model can be represented as

$$F(\mathbf{z}, \mathbf{y}) = \underline{\mathbf{W}} \times_1 \boldsymbol{\psi}(\mathbf{y}) \times_2 \boldsymbol{\varphi}(\mathbf{z}). \tag{9}$$

Step 4: Using a gradient method, latent variables $\{\mathbf{y}_i\}$ are updated so that the approximated cost function of the higher-KSMM decreases in value. The approximated cost function is given by

$$E(\mathbf{y}_i) = \frac{\beta}{2} \sum_{n \in \mathcal{N}_i} \left\| \mathbf{x}_n - F(\mathbf{z}_n, \mathbf{y}_i \,|\, \underline{\mathbf{W}}) \right\|^2 . \tag{10}$$

The integral with respect to $\mathbf{y}$ is omitted to simplify the calculation. Such an approximation is commonly used in SOM and KSMM literatures.

Step 5: Finally, latent variables $\{\mathbf{z}_n\}$ are updated using the gradient method so that the approximated cost function of the lower-KSMM decreases. The cost function is given by

$$E(\mathbf{z}_n) = \frac{\beta}{2} \left\| \mathbf{x}_n - F(\mathbf{z}_n, \mathbf{y}_{i_n}) \right\|^2 . \tag{11}$$

These five steps are repeated until the calculation converges. During the iterations, the length constant of the smoothing kernels is gradually reduced to avoid local minima.

5 Experimental results

5.1 Artificial datasets

The performance of the proposed method was examined using an artificial dataset. We used sinusoidal shape manifolds with different biases (Fig. 1 (a)). Although the original data were two-dimensional (2D), we provided eight extra dimensions, and added 10-dimensional (10D) Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_{10}, \sigma^2 \mathbf{I}_{10})$, where $\sigma = 0.2$. Thus, one-dimensional manifolds were embedded into 10D space. For the training dataset, we prepared 200 tasks, each of which consisted of N_i samples (N_i was common to all tasks) generated randomly. We compared the results of H-KSMM (the proposed method), SOM^2, and a single task on KSMM[1].

[1] For a fair comparison, we modified SOM^2 so that it could represent a continuous mapping using basis functions in the same manner as KSMM. Thus, it should be rather referred to as KSMM^2. By this modification, the result shown for SOM^2 is better than that of the original.

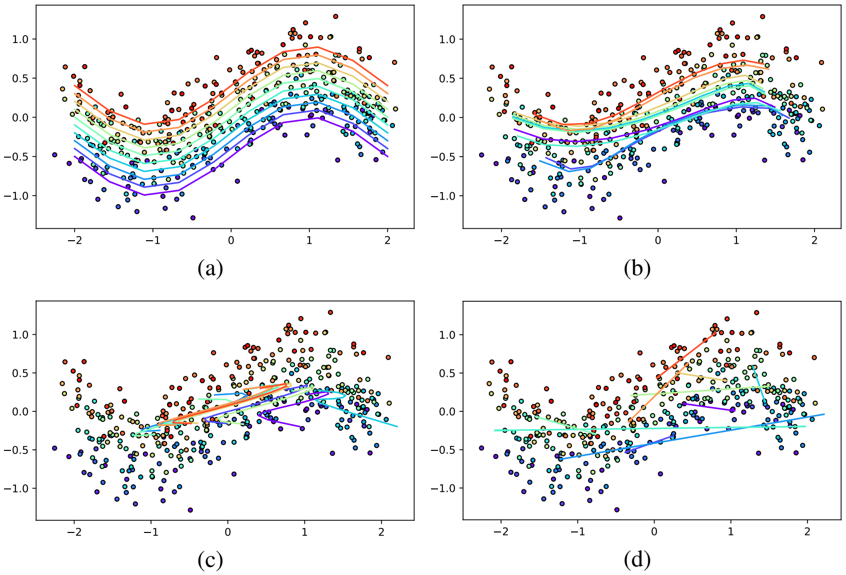


Fig. 1. Results of the artificial dataset. A total of 200 tasks and 2 samples/task were used for training, and 10 of 200 manifolds are shown in the figures. (a) Ground truth. (b) H-KSMM (multi-task learning). (c) SOM^2 (multi-task learning). (d) KSMM (single-task learning).

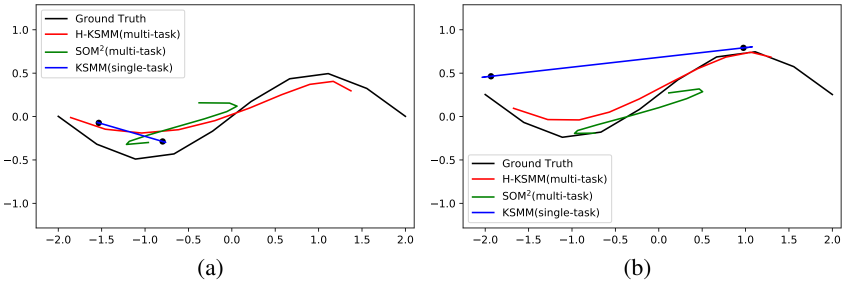
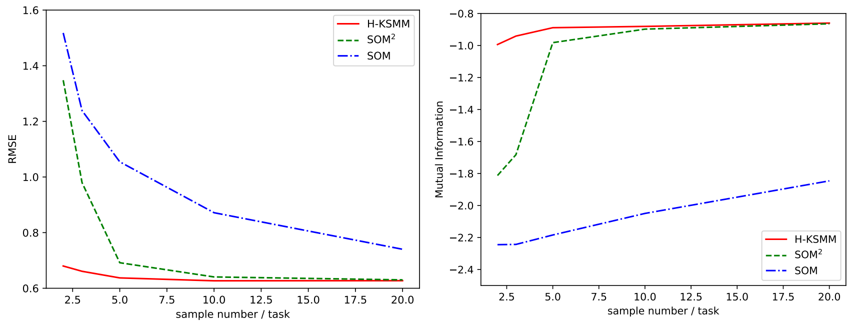


Fig. 2. Two representative tasks extracted from Fig. 1. The two black markers represent the data for the task.

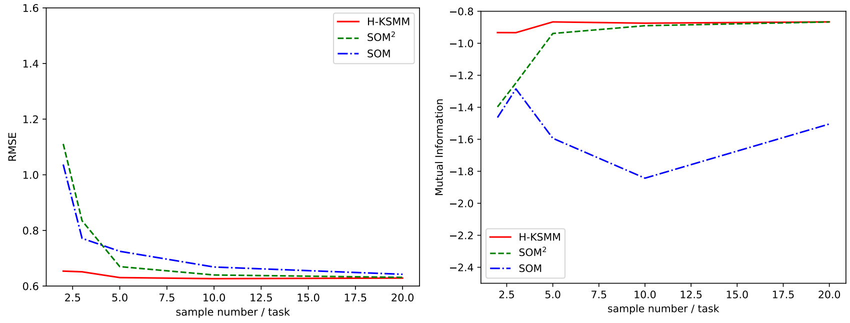
A representative result is shown in Fig. 1. In this case, each task has only two samples. Thus, it is impossible to estimate the manifold shape using single-task learning (Fig. 1 (d)). Surprisingly, the proposed algorithm was able to capture the outlines of the manifold shapes (Fig. 1 (b)). To show details, two of the 200 tasks are shown in Fig. 2. Because only two samples were provided to the task, single-task KSMM estimated the manifold as a straight-line segment that connected two data points. By contrast, H-KSMM was able to reproduce the sinusoidal manifold shape, although its marginal area was truncated because there were insufficient samples.

We assessed learning performance quantitatively using two methods: the root mean square error (RMSE) between the test data and manifold, and mutual information (MI)



(a) (b)

Fig. 3. Generalization performance of existing tasks on the test data. The horizontal axis denotes samples/task for training. (a) Root mean square error between the data and models. (b) Mutual information between the true and estimated latent variables.



(a) (b)

Fig. 4. Generalization performance on new tasks. The horizontal axis denotes samples/task for training. (a) Root mean square error between the data and models. (b) Mutual information between the true and estimated latent variables.

of the true and estimated latent variables. RMSE evaluates the error in visible space $\mathcal{X}$, whereas MI evaluates accuracy in latent space $\mathcal{Z}$. Fig. 3 (a) and (b) show the RMSE and MI measured using the test data on the given tasks, respectively. The results show that H-KSMM exhibited excellent performance, particularly when the number of samples/task was small.

Using the general model, it is not only possible to estimate the manifolds of the given tasks, but also possible to predict manifolds of unseen tasks. Fig. 4 shows the RMSE and MI for 100 new tasks. The results show that H-KSMM has a high generalization capability, even for new tasks.

5.2 Face image datasets

We applied the proposed method to face image modeling. The dataset used was a subset of the extended Cohn–Kanade (CK+) face image database [15, 19]. The data used in the experiment consisted of image sequences of 78 people, where each sequence began

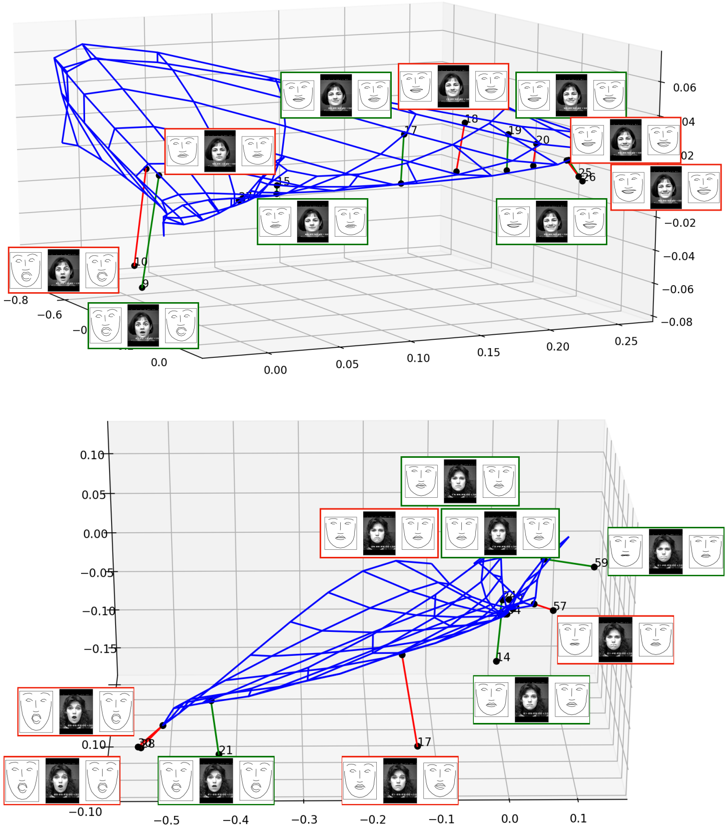


Fig. 5. Results of face image modeling of H-KSMM. Face manifolds of 2 of 78 people are shown. The red boxes represent the training data (5 samples per task) and the green boxes represent the test data. In each box, the original face image is displayed at the center and the corresponding landmark face is indicated on the left-hand side. The landmark face reconstructed by H-KSMM is indicated on the right-hand side.

with a neutral expression and proceeded to a distinct emotional expression. The dataset thus contained a large number of intermediate expressions. In this study, we used four types of sequences: anger, fear, happiness, and surprise. We also used landmark data as features. Thus, each face datum was represented by a 136-dimension vector that corresponded to the 2D coordinates of 68 landmarks. To construct the training data, we sampled five images randomly from each person. Thus, the entire dataset consisted of 78 tasks, each of which consisted of five samples. Note that two expressions were often missing in each task, and it was nearly impossible to estimate the face manifolds using single-task learning.

The results are shown in Fig. 5, which represents the face manifolds estimated by H-KSMM depicted in 3D space spanning the first three principal components. H-KSMM represented the training data well (indicated by red boxes) and reproduced the test data successfully (indicated by green boxes). Thus, these face manifolds successfully represented various facial expressions, even though the data provided were insufficient.

6 Discussion and conclusion

In this paper, we proposed a method for multi-task manifold modeling based on the hierarchical modeling approach. Characteristics of the method are two means of information transfer: *the weighted mixture of sample datasets* and *the weighted mixture of task models*. The latter method of information transfer is mediated by a higher-order model in hierarchical modeling; that is, the former method of information transfer was executed *before* the manifold modeling of each task, whereas the latter was executed *after* manifold modeling. Providing a theoretical basis for these means of information transfer will form the focus of our future work in this area.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 18K11472. We thank Saad Anis, PhD and Maxine Garcia, PhD, from Edanz Group for editing a draft of this manuscript.

References

1. Bishop, C.M., Svensen, M., Williams, C.K.I.: GTM: The generative topographic mapping. Neural Computation **10**, 215–234 (1998)
2. Caruana, R.: Multitask learning. Machine Learning **28**, 41–75 (1997)
3. Chang, Y., Hu, C., Feris, R., Turk, M.: Manifold based analysis of facial expression. Image and Vision Computing **24**(6), 605–614 (2006)
4. Cheng, Y.: Convergence and ordering of kohonen's batch map. Neural Computation **9**(8), 1667–1676 (Nov 1997)
5. Dedrick, R.F., Ferron, J.M., Hess, M.R., Hogarty, K.Y., Kromrey, J.D., Lang, T.R., Niles, J.D., Lee, R.S.: Multilevel modeling: A review of methodological issues and applications. Review of Educational Research **79**(1), 69–102 (2009)
6. Furukawa, T.: SOM of SOMs: Self-organizing map which maps a group of self-organizing maps. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **3696 LNCS**, 391–396 (2005)
7. Furukawa, T.: SOM of SOMs. Neural Networks **22**(4), 463–478 (2009)
8. Graepel, T., Burger, M., Obermayer, K.: Self-organizing maps: Generalizations and new optimization techniques. Neurocomputing **21**, 173–190 (1998)
9. Han, L., Zhang, Y.: Learning multi-level task groups in multi-task learning. vol. 4, pp. 2638–2644 (2015)
10. Han, L., Zhang, Y.: Learning tree structure in multi-task learning. vol. 2015-August, pp. 397–406 (2015)
11. Heskes, T., Spanjers, J.J., Wiegerinck, W.: EM algorithms for self-organizing maps. In: IJCNN (6). pp. 9–14 (2000)
12. Ishibashi, H., Furukawa, T.: Multilevel-multigroup analysis by hierarchical tensor SOM network. In: Proc. of ICONIP. pp. 459–466 (2016)
13. Ishibashi, H., Furukawa, T.: Hierarchical tensor SOM network for multilevel-multigroup analysis. Neural Processing Letters pp. 1–15 (2017)

14. Jiang, J., Zhang, L., Furukawa, T.: Improving the generalization of fisherface by training class selection using SOM^2. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **4233 LNCS - II**, 278–285 (2006)
15. Kanade, T., Cohn, J., Tian, Y.: Comprehensive database for facial expression analysis. pp. 46–53 (2000)
16. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics **43**(1), 59–69 (1982)
17. Lawrence, N.D.: Probabilistic non-linear principal component analysis with gaussian process. Journal of Machine Learning Research **6**, 1783–1816 (2005)
18. Lawrence, N.D.: Gaussian process latent variable models for visualisation of high dimensional data. In: Neural Information Processing Systems (NIPS). vol. 16, pp. 329–336 (2003)
19. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. pp. 94–101 (June 2010)
20. Luttrell, S.P.: Self-organization: A derivation from first principle of a class of learning algorithms. In: IEEE Conference on Neural Networks. pp. 495–498 (1989)
21. Ohkubo, T., Tokunaga, K., Furukawa, T.: RBF×SOM: An efficient algorithm for large-scale multi-system learning. IEICE Transactions on Information and Systems **E92-D**(7), 1388–1396 (2009)
22. Ohkubo, T., Furukawa, T., Tokunaga, K.: Requirements for the learning of multiple dynamics. In: Laaksonen, J., Honkela, T. (eds.) Advances in Self-Organizing Maps. Lecture Notes in Computer Science, vol. 6731, pp. 101–110. Springer (2011)
23. Shan, C., Gong, S., McOwan, P.: Appearance manifold of facial expression. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **3766 LNCS**, 221–230 (2005)
24. Verbeek, J., Vlassis, N., Krose, B.: Self-organizing mixture models. Neurocomputing **63**, 99–123 (Jan 2005)
25. Yakushiji, S., Furukawa, T.: Shape space estimation by SOM^2. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **7063 LNCS**(PART 2), 618–627 (2011)
26. Yakushiji, S., Furukawa, T.: Shape space estimation by higher-rank of SOM. Neural Computing and Applications **22**(7-8), 1267–1277 (2013)
27. Yamane, I., Yger, F., Berar, M., Sugiyama, M.: Multitask principal component analysis. In: Durrant, R.J., Kim, K.E. (eds.) Proceedings of The 8th Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 63, pp. 302–317. PMLR, The University of Waikato, Hamilton, New Zealand (16–18 Nov 2016)
28. Zhang, Y., Yang, Q.: An overview of multi-task learning. National Science Review p. nwx105 (2017)
29. Zweig, A., Weinshall, D.: Hierarchical regularization cascade for joint learning. pp. 1074–1082. No. PART 2 (2013)