PAPER

Accurate Estimation of Personalized Video Preference Using Multiple Users' Viewing Behavior

Yoshiki ITO[†a)], ***Student Member***, **Takahiro OGAWA**[†b)], ***and*** **Miki HASEYAMA**[†c)], ***Members***

SUMMARY A method for accurate estimation of personalized video preference using multiple users' viewing behavior is presented in this paper. The proposed method uses three kinds of features: a video, user's viewing behavior and evaluation scores for the video given by a target user. First, the proposed method applies Supervised Multiview Spectral Embedding (SMSE) to obtain lower-dimensional video features suitable for the following correlation analysis. Next, supervised Multi-View Canonical Correlation Analysis (sMVCCA) is applied to integrate the three kinds of features. Then we can get optimal projections to obtain new visual features, "canonical video features" reflecting the target user's individual preference for a video based on sMVCCA. Furthermore, in our method, we use not only the target user's viewing behavior but also other users' viewing behavior for obtaining the optimal canonical video features of the target user. This unique approach is the biggest contribution of this paper. Finally, by integrating these canonical video features, Support Vector Ordinal Regression with Implicit Constraints (SVORIM) is trained in our method. Consequently, the target user's preference for a video can be estimated by using the trained SVORIM. Experimental results show the effectiveness of our method.

key words: *multiview approach, spectral embedding, canonical correlation analysis, video preference, viewing behavior*

1. Introduction

The Internet has made it easier to access many videos via video-sharing services such as YouTube[*] and video-streaming services such as Netflix[**] and Hulu[***]. The number of videos posted to these services is expected to continue to increase [1]. Users are generally required to input queries when they want to watch a video. Thus, when users cannot provide suitable queries that accurately reflect their desired video, successful retrieval of a video becomes difficult [2]. In order to solve this problem, many video recommendation methods that do not require any queries have been studied, and they are broadly classified into two main types of method [3]: collaborative filtering [4], [5] and content-based filtering [6], [7]. In methods based on collaborative filtering, users who have similar preferences are found on the basis of evaluation scores given by the users. However, there is the problem that videos that have not been evaluated in advance cannot be recommended by these methods (first-rater problem). In methods based on content-based filtering, on the other hand, videos are recommended to the target user by using features obtained from the video (video features) to overcome the problem of collaborative filtering. However, video features obtained by these methods do not take the user's preference into consideration. Thus, it is difficult to recommend videos by using only video features.

In order to solve the above problem, it is necessary to extract each user's preference. There have been some benchmarking studies on extraction of important features included in original features by using the relationship between original features and their corresponding scores (e.g., original video features and evaluation scores for the video) [8]–[13]. However, when different users give the same evaluation scores for the same video, features extracted by these methods become the same. In other words, supervised feature extraction methods for original features are still not sufficient to extract each user's preference. Therefore, the introduction of other features that can represent the target user's preference more accurately is needed.

Many methods use biological signals (e.g., brain waves and heart rate) to extract unique features for each user [14]–[19]. However, these approaches put a physical burden on the users since biological signals are usually obtained by a device attached to the body. However, since in-cameras are mounted on some information and communication devices (e.g., personal computers, smartphones and tablet-type information terminals), user's viewing behavior can be determined without putting a physical burden on the user. In addition, viewing behavior such as gazing, facial expressions and body movements is closely related to the user's attention, and they are important factors for extracting the user's individual preference [20]. From these viewpoints, some methods to predict a target user's evaluation score for a video by using his/her viewing behavior have been reported [21]–[24]. However, these methods have not taken into account users who do not show conspicuous viewing behavior. There are clear differences in the degrees of viewing behavior shown by users. Thus, the problem of the use of only a target user's viewing behavior not working effectively must be overcome.

In this paper, we propose a new framework to further improve the accuracy of estimation of personalized video

Manuscript received May 31, 2017.
Manuscript revised October 7, 2017.
Manuscript publicized November 22, 2017.

[†]The authors are with the Graduate School of Information Science and Technology, Hokkaido University, Sapporo-shi, 060–0814 Japan.
a) E-mail: ito@lmd.ist.hokudai.ac.jp
b) E-mail: ogawa@lmd.ist.hokudai.ac.jp
c) E-mail: miki@ist.hokudai.ac.jp

DOI: 10.1587/transinf.2017EDP7178

[*] http://www.youtube.com/
[**] http://www.netflix.com/
[***] http://www.hulu.com/

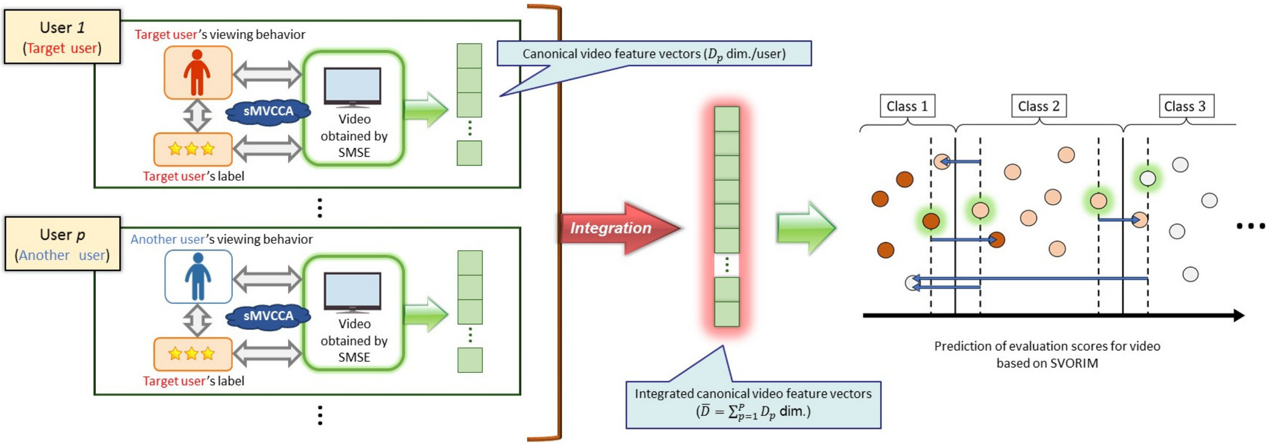
User 1
(Target user)
Target user's viewing behavior
sMVCCA
Video obtained by SMSE
Target user's label
Canonical video feature vectors (D_p dim./user)
User p
(Another user)
Another user's viewing behavior
sMVCCA
Video obtained by SMSE
Target user's label
Integration
Integrated canonical video feature vectors
($\bar{D} = \sum_{p=1}^{P} D_p$ dim.)
Class 1
Class 2
Class 3
Prediction of evaluation scores for video based on SVORIM

Fig. 1 Overview of the proposed method. Our method uses viewing behavior obtained from multiple users (P users), not just a target user. The use of multiple users' viewing behavior is supportive for estimation of the target user's preference.

preference in order to solve the above problem. Figure 1 shows an overview of the proposed method. As shown in the figure, we focus on the relationships between the video, viewing behavior and evaluation scores. Furthermore, our method uses not only the target user's viewing behavior but also other users' viewing behavior. Therefore, the proposed method tries to find the target user's preference based on multiple relationships obtained from "video", "viewing behavior acquired from multiple users including the target user" and "evaluation scores given by the target user for the video". In our method, we first introduce Supervised Multiview Spectral Embedding (SMSE) [25], which is one of the state-of-the-art methods for multimodal dimensionality reduction. Using this method, the dimension of the video features can be effectively reduced while weights of each feature are being considered. Next, our method calculates "canonical video features", which maximize correlations between the three kinds of features using Supervised Multi-View Canonical Correlation Analysis (sMVCCA) [26]. Furthermore, we also calculate several canonical video features using multiple users' viewing behavior, and these canonical video features are integrated to represent the target user's video preference. This approach using multiple users' viewing behavior provides a solution to the above-mentioned problem, and this is the biggest contribution of this paper. Finally, Support Vector Ordinal Regression with Implicit Constraints (SVORIM) [27] (i.e., a classifier) is trained by the integrated canonical video features, and prediction of evaluation scores for a new video becomes feasible. Note that our work shown in this paper is an extended version of [28].

In our method, all of the three features (video, viewing behavior and evaluation scores for videos) are required for training. Meanwhile, in the test phase to predict evaluation scores for new videos, only the video features are input to the classifier; that is, any users' viewing behavior and evaluation scores are not needed. This prediction is made possible by transforming the raw video features into the canonical video features that have strong correlations with the users' viewing behavior and evaluation scores. This is one of the strengths of our method. On the other hand, if one of these features (e.g., viewing behavior or evaluation scores) cannot be obtained at all, our method does not work. In actual application, viewing behavior may not be obtained in sections of videos by the error of face or body tracking. However, in our method, the viewing behavior (and evaluation scores) is used in the training phase only, and not used in the test phase described the above. We do not need to be concerned about the tracking error of short time since training data that failed to obtain can be substituted for data obtained from other videos.

This paper is organized as follows. First, we explain extraction of three kinds of features used in our method and their smoothing in Sect. 2. In Sect. 3, we calculate the integrated canonical video features that are newly desired based on SMSE and sMVCCA using multiple users' viewing behavior. In Sect. 4, SVORIM is trained from the integrated canonical video features to realize estimation of the evaluation scores for a new video. In Sect. 5, we show experimental results to confirm the effectiveness of our method. Finally, conclusions are given in Sect. 6.

2. Feature Calculation

This section shows the extraction of features. First, we explain three kinds of features used in our method in **2.1**. Next, we explain procedures for smoothing of these features in **2.2**.

2.1 Extraction of Three Kinds of Features

In this subsection, we explain three features used in our method: video features, viewing behavior features and label features. In a training dataset, the proposed method calculates video features $\boldsymbol{v}_i$ ($i = 1, 2, \cdots, N$) and their corresponding pth ($1 \leq p \leq P$) user's viewing behavior features

Table 1 Three kinds of features used in the proposed method.

	Feature Type	Dimensions
Video features	Audio (Dynamics, Spectral, Timbre, Tonal, Rhythm) [29]	145
	HSV Color Histogram (HSVCH) [30]	64
	Bag of features [31] based on Speeded-Up Robust Features [32] (SURF-Bof)	1000
Total	-	**1209**
Viewing behavior features (Face)	2D rectangle region of the face	2
	3D angle of the face	3
	3D movement of the head position	3
	Facial expression descriptor based on Action Units [33]	6
Viewing behavior features (Body)	Distance between the user's centroid and a display	1
	2D movement of the user's centroid	2
	2D rectangle region of the body	2
	Angle of the body based on distance between both shoulders and a display	3
Total	-	**22**
Label features	Features based on the score evaluated by a target user	R
Total	-	***R***

$\boldsymbol{b}_i^{(p)}$ and label features $\boldsymbol{l}_i$ of a target user, where $N = \sum_{i=1}^{N_v} n_i$, and N_v means the number of training video, and n_i is the number of frames in ith video; that is, N is the number of all training samples. The details of these three features are shown below.

Video features (1209 dimensions):
We adopt 145-dimensional audio features obtained by MIRtoolbox [29], which consists of Dynamics, Spectral, Timbre, Tonal and Rhythm as shown in Table 1. In addition, we use HSV Color Histogram (HSVCH) [30] and Bag of features [31] based on Speeded-Up Robust Features [32] (SURF-Bof). Thus, the video feature vectors $\boldsymbol{v}_i \in \mathbb{R}^{D_v}$ are extracted for each ith sample, where $D_v = 1209$ as shown in Table 1. Note that we apply SMSE to the 1209-dimensional original video features in 3.1 and then reduce these dimensions.

Viewing behavior features (22 dimensions):
The user's facial features and body movement features are obtained by using a Kinect sensor[†] as shown in Table 1. In order to calculate the facial features, Kinect detects landmark points on the user's face and then constructs a 3D face model corresponding to the landmark points. This 3D face model enables acquisition of some data such as head poses and facial expression descriptor based on Action Units (AUs) [33] provided by the Microsoft Face Tracking Software Development Kit for Kinect for Windows (Face Tracking SDK)[††]. These AUs support six points of facial movements (Upper lip raiser, Jaw lowerer, Lip stretcher, Brow lowerer, Lip corner depressor and Outer brow raiser). Thus, 14-dimensional facial features can be obtained. In addition, we also obtain the user's angle or distance to a display based on the user's skeleton extracted from the Kinect. Thus, we obtain 8-dimensional body movement features. In this way, we obtain the viewing behavior feature vectors $\boldsymbol{b}_i^{(p)} \in \mathbb{R}^{D_b}$ for each ith sample, where $D_b = 22$.

Label features (R dimensions):
A target user evaluates all videos in R grades while watching them. Thus, we obtain evaluation scores for video $l_i \in \mathbb{R}^1$ from the target user. Note that these scores are expanded into R-dimensional binary vectors based on [26] after they are smoothed in 2.2. Finally, the label feature vectors $\boldsymbol{l}_i \in \mathbb{R}^{D_l}$ are extracted for each ith sample, where $D_l = R$.

2.2 Smoothing of Features

In this subsection, we explain smoothing of the three features described in the previous subsection. These features are smoothed in order to improve their robustness based on [28]. First, we define a constant parameter of short time width s corresponding to a frame of the video. We use before and after s samples for each ith sample. In other words, $2s + 1$ samples in total are used for each sample. Next, the three features shown in the previous subsection are smoothed as follows:

$$
\begin{aligned}
\boldsymbol{v}_i &\leftarrow \frac{1}{2s+1}\sum_{w=i-s}^{i+s} \boldsymbol{v}_w, \\
\boldsymbol{b}_i^{(p)} &\leftarrow \frac{1}{2s+1}\sum_{w=i-s}^{i+s} \boldsymbol{b}_w^{(p)}, \\
l_i &\leftarrow \mathrm{round}\left(\frac{1}{2s+1}\sum_{w=i-s}^{i+s} l_w\right),
\end{aligned}
\tag{1}
$$

where round($\cdot$) means an operator to round off evaluation scores. As described in 2.1, label features are expanded into binary vectors after smoothing. Since the smoothing of the three features enables improvement of their robustness, we will be able to observe more practical features.

3. Calculation of Integrated Canonical Video Features

In this section, the method for calculation of the integrated canonical video features is shown. First, we explain the dimensionality reduction of video features based on SMSE in

[†] http://www.microsoft.com/en-us/kinectforwindows/
[††] http://msdn.microsoft.com/en-us/library/jj130970.aspx

3.1. Next, we explain the method for calculating sMVCCA-based integrated canonical video features using the three kinds of features in **3.2**. It should be noted that the maximum dimensionality of the canonical video features calculated by sMVCCA becomes less than the lowest one in each modality. Since the dimensionality of video features is higher than those of the other features, viewing behavior features and label features, we first show reduction of the dimensionality of video features in **3.1**.

3.1 Dimensionality Reduction of Video Features Based on SMSE

SMSE is one of the state-of-the-art supervised dimensionality reduction methods for multimodal features. In our method, SMSE is used in order to reduce dimensions of video features while weights of each feature can be adjusted.

As described in the previous section, our method prepares three kinds of video features, Audio, HSVCH and SURF-Bof. We then define the mth ($m \in \{\mathrm{A}, \mathrm{H}, \mathrm{S}\}$) features as $\boldsymbol{V}_m = [\boldsymbol{v}_{m,1}, \cdots, \boldsymbol{v}_{m,N}] \in \mathbb{R}^{D_{v_m} \times N}$, where A, H and S mean Audio, HSVCH and SURF-Bof, respectively. Furthermore, D_{v_m} means the dimensionality of the mth feature. Next, we define a matrix containing a target sample and its K-nearest neighborhoods as $\boldsymbol{V}_{m,i} = [\boldsymbol{v}_{m,i}, \boldsymbol{v}_{m,i_1}, \cdots, \boldsymbol{v}_{m,i_K}] \in \mathbb{R}^{D_{v_m} \times (K+1)}$ for each sample i. In our method, the mapping of $\boldsymbol{V}_{m,i}$ in the spaces constructed for dimensionality reduction (embedding spaces) is defined as $\hat{\boldsymbol{V}}_{m,i} = [\hat{\boldsymbol{v}}_{m,i}, \hat{\boldsymbol{v}}_{m,i_1}, \cdots, \hat{\boldsymbol{v}}_{m,i_K}] \in \mathbb{R}^{D_{\hat{v}} \times (K+1)}$, where $D_{\hat{v}}$ means the dimensionality of the mapped video features in the embedding space, and $D_{\hat{v}} \leq D_{v_m}$. In order to preserve local neighborhood embeddings for all samples, the following optimization problem is solved:

$$\underset{\hat{\boldsymbol{V}}, \boldsymbol{\alpha}}{\arg\min} \sum_{m \in \{\mathrm{A,H,S}\}} \alpha_m \sum_{i=1}^{N} \sum_{k=1}^{K} ||\hat{\boldsymbol{v}}_{m,i} - \hat{\boldsymbol{v}}_{m,i_k}||^2 \cdot (\omega_{m,i_k}), \tag{2}$$

where $\hat{\boldsymbol{V}}_i = \{\hat{\boldsymbol{V}}_{m,i}\}_{m \in \{\mathrm{A,H,S}\}}$, and $\hat{\boldsymbol{V}} = \{\hat{\boldsymbol{V}}_i\}_{i=1}^{N}$. Moreover, $\boldsymbol{\alpha} = \{\alpha_m\}_{m \in \{\mathrm{A,H,S}\}}$ means weights for each feature. In addition, we calculate the following weights in order to preserve the same evaluation scores for a video and calculate the similarity between $\boldsymbol{v}_{m,i}$ and $\boldsymbol{v}_{m,i_k}$ in the original video feature space:

$$\omega_{m,i_k} = \begin{cases} \exp(-||\boldsymbol{v}_{m,i} - \boldsymbol{v}_{m,i_k}||^2 / t_m) & \text{if } l_i = l_{i_k} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where l_i is an evaluation score of $\boldsymbol{v}_{m,i}$, and l_{i_k} is an evaluation score of $\boldsymbol{v}_{m,i_k}$, and t_m is defined as follows:

$$t_m = \frac{2}{N(N-1)} \sum_{p=1}^{N} \sum_{q=p+1}^{N} \exp(-||\boldsymbol{v}_{m,p} - \boldsymbol{v}_{m,q}||^2). \tag{4}$$

Thus, the optimization problem is solved with consideration of the similarity within the same evaluation scores.

Next, we explain global coordinate alignment. Each local neighborhood embedding $\hat{\boldsymbol{V}}_{m,i} \in \mathbb{R}^{D_{\hat{v}} \times (K+1)}$ is a subset of a global embedding $\hat{\boldsymbol{V}} \in \mathbb{R}^{D_{\hat{v}} \times N}$. Thus, Eq. (2) can be rewritten as follows:

$$\begin{aligned} &\underset{\hat{\boldsymbol{V}}, \boldsymbol{\alpha}}{\arg\min} \sum_{m \in \{\mathrm{A,H,S}\}} \alpha_m^{\gamma} \sum_{i=1}^{N} \hat{\boldsymbol{V}} \boldsymbol{L}_{m,i}^{(\mathrm{n})} \hat{\boldsymbol{V}}^{\mathrm{T}} \\ &\quad \text{s.t. } \hat{\boldsymbol{V}} \hat{\boldsymbol{V}}^{\mathrm{T}} = \boldsymbol{I}_{D_{\hat{v}}}; \sum_{m \in \{\mathrm{A,H,S}\}} \alpha_m = 1,\ \alpha_i \geq 0, \end{aligned} \tag{5}$$

where $\boldsymbol{L}_{m,i}^{(\mathrm{n})} \in \mathbb{R}^{N \times N}$ is the normalized Laplacian matrix, and $\boldsymbol{I}_{D_{\hat{v}}} \in \mathbb{R}^{D_{\hat{v}} \times D_{\hat{v}}}$ is the identity matrix. The matrix $\boldsymbol{L}_{m,i}^{(\mathrm{n})}$ is calculated as follows. We first define the weight matrix $\boldsymbol{W}_m \in \mathbb{R}^{N \times N}$, whose (i, i_k) element is ω_{m,i_k} defined in Eq. (3). At the same time, we also define the diagonal matrix $\boldsymbol{D}_m \in \mathbb{R}^{N \times N}$, whose diagonal element has $\sum_l [\boldsymbol{W}_m]_{i,l}$. From these, Laplacian matrices $\boldsymbol{L}_m = \boldsymbol{D}_m - \boldsymbol{W}_m$ are calculated. Thus, we can obtain the normalized Laplacian matrix as follows:

$$\begin{aligned} \boldsymbol{L}_m^{(\mathrm{n})} &= \boldsymbol{D}_m^{-1/2} \boldsymbol{L}_m \boldsymbol{D}_m^{-1/2} \\ &= \boldsymbol{I}_N - \boldsymbol{D}_m^{-1/2} \boldsymbol{W}_m \boldsymbol{D}_m^{-1/2}. \end{aligned} \tag{6}$$

Note that we use the symmetric normalized graph Laplacian used in SMSE [25] and multiview spectral embedding (MSE) [34], which is an unsupervised version of SMSE. Since α_m must be non-negative values in Eqs. (2) and (5), the property that the eigenvalues of the positive semi-definite matrix are non-negative values is used in [25], [34]. Therefore, other than the symmetric version, we will also be able to use other versions of graph Laplacian (e.g., combinatorial or random-walk version) in SMSE. Moreover, γ in Eq. (5) is a parameter that can control the weights of each feature. The domain of this parameter is $1 < \gamma$. If the parameter is set as $\gamma \to 1$, the bias of the weights of each feature becomes large. On the other hand, if the parameter is set as $\gamma \to \infty$, the bias becomes small; that is, each weight becomes equal.

Finally, in order to solve Eq. (5), an alternating optimization algorithm is adopted on the basis of [34], [35]. First, the weights are initialized as $\alpha_i = 1/3$, and the following $\boldsymbol{\Lambda}$ is calculated:

$$\boldsymbol{\Lambda} = \sum_{m \in \{\mathrm{A,H,S}\}} \alpha_m^{\gamma} \sum_{i=1}^{N} \boldsymbol{L}_{m,i}^{(\mathrm{n})}. \tag{7}$$

Next, we obtain embedding video features $\hat{\boldsymbol{V}} \in \mathbb{R}^{D_{\hat{v}} \times N}$ ($D_{\hat{v}} = \min\{D_{v_1}, D_{v_2}, D_{v_3}\}$) by solving the following optimization problem using $\boldsymbol{\Lambda}$ obtained from Eq. (7):

$$\underset{\hat{\boldsymbol{V}}}{\arg\min}\ \hat{\boldsymbol{V}} \boldsymbol{\Lambda} \hat{\boldsymbol{V}}^{\mathrm{T}} \qquad \text{s.t. } \hat{\boldsymbol{V}} \hat{\boldsymbol{V}}^{\mathrm{T}} = \boldsymbol{I}_{D_{\hat{v}}}, \tag{8}$$

where $\hat{\boldsymbol{V}}$ is a matrix including eigenvectors corresponding to eigenvalues obtained by eigenvalue decomposition of $\boldsymbol{\Lambda}$. Note that we choose the smallest $D_{\hat{v}}$ eigenvalues. From Eq. (5), we obtain the following Lagrange function using the Lagrange multiplier approach:

$$\begin{aligned}&\boldsymbol{\Lambda}(\boldsymbol{\alpha},\zeta)\\&=\sum_{m\in\{\mathrm{A,H,S}\}}\alpha_m^{\gamma}\sum_{i=1}^{N}\hat{\boldsymbol{V}}\boldsymbol{L}_{m,i}^{(\mathrm{n})}\hat{\boldsymbol{V}}^{\mathrm{T}}-\zeta\left(\sum_{m\in\{\mathrm{A,H,S}\}}\alpha_m-1\right),\end{aligned}\tag{9}$$

where ζ is the Lagrange multiplier. From Eq. (9), α_m are updated by solving the above function as follows:

$$\alpha_m=\frac{\left\{1/\mathrm{tr}\left(\hat{\boldsymbol{V}}\boldsymbol{L}_m^{(\mathrm{n})}\hat{\boldsymbol{V}}^{\mathrm{T}}\right)\right\}^{\frac{1}{\gamma-1}}}{\sum_{m\in\{\mathrm{A,H,S}\}}\left\{1/\mathrm{tr}\left(\hat{\boldsymbol{V}}\boldsymbol{L}_m^{(\mathrm{n})}\hat{\boldsymbol{V}}^{\mathrm{T}}\right)\right\}^{\frac{1}{\gamma-1}}},\tag{10}$$

where $\mathrm{tr}(\cdot)$ is an operator of trace. Since $\boldsymbol{L}_m^{(\mathrm{n})}$ is a positive semi-definite matrix, its eigenvalue and α_m are non-negative values. We then perform calculations from Eq. (7) to Eq. (10), iteratively. Our method continues to update α_m until the difference by updates is less than Th_{α}. From the converged α_m, we can obtain the final $\hat{\boldsymbol{V}}$ using Eqs. (7) and (8). Specifically, the objective function in Eq. (8) can be rewritten as follows:

$$\arg\min_{\hat{\boldsymbol{V}}}\ \hat{\boldsymbol{V}}\boldsymbol{\Lambda}\hat{\boldsymbol{V}}^{\mathrm{T}}=\arg\min_{\boldsymbol{Q}}\ \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{Q},\tag{11}$$

where $\boldsymbol{Q}$ is the optimal projection for $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$ under the constraint of $\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}=\boldsymbol{I}_{D_v}$. As seen from the above, SMSE effectively reduces dimensions from the original video features $\boldsymbol{V}\in\mathbb{R}^{D_v\times N}$ ($D_v=1209$) to the embedding video features $\hat{\boldsymbol{V}}\in\mathbb{R}^{D_{\hat{v}}\times N}$ ($D_{\hat{v}}=\min\{D_{v_1},D_{v_2},D_{v_3}\}$) while giving consideration to the weights of each feature and the local neighborhood structures within classes.

3.2 sMVCCA-based Integrated Canonical Video Features

In this subsection, we calculate sMVCCA-based integrated canonical video features using the three features described in the previous section. First, we obtain an embedding video feature matrix $\hat{\boldsymbol{V}}=[\hat{\boldsymbol{v}}_1,\hat{\boldsymbol{v}}_2,\cdots,\hat{\boldsymbol{v}}_N]\in\mathbb{R}^{D_{\hat{v}}\times N}$ as mentioned in the previous subsection. Next, we define several matrices for viewing behavior features since our method uses several viewing behavior obtained from multiple users, not just a target user. Specifically, we define a pth user's viewing behavior feature matrix $\boldsymbol{B}^{(p)}=[\boldsymbol{b}_1^{(p)},\boldsymbol{b}_2^{(p)},\cdots,\boldsymbol{b}_N^{(p)}]\in\mathbb{R}^{D_b\times N}$, where $p\in\{1,2,\cdots,P\}$, and P is the number of integrated candidate users. We define $p=1$ as the target user. Moreover, we define a binary vector $\boldsymbol{l}_i\in\mathbb{R}^{D_l}$ obtained from evaluation scores given by the target user. Based on [26], the vector is expanded into a binary matrix $\boldsymbol{L}=[\boldsymbol{l}_1,\boldsymbol{l}_2,\cdots,\boldsymbol{l}_N]\in\mathbb{R}^{D_l\times N}$ since the dimensionality of the canonical video features integrated by sMVCCA is less than $\min\{D_{\hat{v}},D_b,D_l\}$. From these matrices, we calculate the optimal projection vectors by maximizing the sum of three kinds of pair correlations: correlation between $\hat{\boldsymbol{V}}$ and $\boldsymbol{B}^{(p)}$ (video and pth user's viewing behavior), correlation between $\hat{\boldsymbol{V}}$ and $\boldsymbol{L}$ (video and label) and correlation between $\boldsymbol{B}^{(p)}$ and $\boldsymbol{L}$ (pth user's viewing behavior and label). Specifically, we calculate the following optimization problem in order to obtain the optimal projection vectors $\overline{\boldsymbol{w}}_{\hat{v}}^{(p)}\in\mathbb{R}^{D_{\hat{v}}}$, $\overline{\boldsymbol{w}}_b^{(p)}\in\mathbb{R}^{D_b}$ and $\overline{\boldsymbol{w}}_l^{(p)}\in\mathbb{R}^{D_l}$:

$$\begin{aligned}&\left(\overline{\boldsymbol{w}}_{\hat{v}}^{(p)},\overline{\boldsymbol{w}}_b^{(p)},\overline{\boldsymbol{w}}_l^{(p)}\right)\\&\qquad=\arg\max_{\boldsymbol{w}_{\hat{v}}^{(p)},\boldsymbol{w}_b^{(p)},\boldsymbol{w}_l^{(p)}}\left(\boldsymbol{w}_{\hat{v}}^{(p)\mathrm{T}}\hat{\boldsymbol{V}}\boldsymbol{B}^{(p)\mathrm{T}}\boldsymbol{w}_b^{(p)}\right.\\&\qquad\quad\left.+\boldsymbol{w}_{\hat{v}}^{(p)\mathrm{T}}\hat{\boldsymbol{V}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{w}_l^{(p)}+\boldsymbol{w}_b^{(p)\mathrm{T}}\boldsymbol{B}^{(p)}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{w}_l^{(p)}\right)\\&\text{s.t.}\quad\boldsymbol{w}_{\hat{v}}^{(p)\mathrm{T}}\left(\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^{\mathrm{T}}+\epsilon\boldsymbol{I}_{D_{\hat{v}}}\right)\boldsymbol{w}_{\hat{v}}^{(p)}\\&\qquad\quad+\boldsymbol{w}_b^{(p)\mathrm{T}}\left(\boldsymbol{B}^{(p)}\boldsymbol{B}^{(p)\mathrm{T}}+\epsilon\boldsymbol{I}_{D_b}\right)\boldsymbol{w}_b^{(p)}\\&\qquad\quad+\boldsymbol{w}_l^{(p)\mathrm{T}}\left(\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}+\epsilon\boldsymbol{I}_{D_l}\right)\boldsymbol{w}_l^{(p)}=1,\end{aligned}\tag{12}$$

where ϵ is a small hyperparameter to balance the regularization term. The above optimization problem is rewritten in the following generalized eigenvalue problem using the Lagrange multiplier approach:

$$\begin{aligned}&\begin{bmatrix}\boldsymbol{0}&\hat{\boldsymbol{V}}\boldsymbol{B}^{(p)\mathrm{T}}&\hat{\boldsymbol{V}}\boldsymbol{L}^{\mathrm{T}}\\\boldsymbol{B}^{(p)}\hat{\boldsymbol{V}}^{\mathrm{T}}&\boldsymbol{0}&\boldsymbol{B}^{(p)}\boldsymbol{L}^{\mathrm{T}}\\\boldsymbol{L}\hat{\boldsymbol{V}}^{\mathrm{T}}&\boldsymbol{L}\boldsymbol{B}^{(p)\mathrm{T}}&\boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{w}_{\hat{v}}^{(p)}\\\boldsymbol{w}_b^{(p)}\\\boldsymbol{w}_l^{(p)}\end{bmatrix}\\&=\lambda\left(\begin{bmatrix}\hat{\boldsymbol{V}}\hat{\boldsymbol{V}}^{\mathrm{T}}&\boldsymbol{0}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{B}^{(p)}\boldsymbol{B}^{(p)\mathrm{T}}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{0}&\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}\end{bmatrix}+\epsilon\boldsymbol{I}_D\right)\begin{bmatrix}\boldsymbol{w}_{\hat{v}}^{(p)}\\\boldsymbol{w}_b^{(p)}\\\boldsymbol{w}_l^{(p)}\end{bmatrix},\end{aligned}\tag{13}$$

where λ is an eigenvalue, and $D=D_{\hat{v}}+D_b+D_l$. By solving the above generalized eigenvalue problem, we can obtain the following projection matrix for a video:

$$\overline{\boldsymbol{W}}_{\hat{v}}^{(p)}=\left[\overline{\boldsymbol{w}}_{\hat{v},1}^{(p)},\overline{\boldsymbol{w}}_{\hat{v},2}^{(p)},\cdots,\overline{\boldsymbol{w}}_{\hat{v},d}^{(p)},\cdots,\overline{\boldsymbol{w}}_{\hat{v},D_p}^{(p)}\right]\in\mathbb{R}^{D_{\hat{v}}\times D_p},\tag{14}$$

where $D_p<\min\{D_{\hat{v}},D_b,D_l\}$, and λ_d ($\lambda_d>\lambda_{d+1}$; $d=1,2,\cdots,D_p-1$). Note that $\overline{\boldsymbol{w}}_{\hat{v},d}^{(p)}$ is an optimal projection vector for a video corresponding to an eigenvalue λ_d. Next, we calculate canonical video features based on the pth users' viewing behavior using the optimal projection matrix for a video as follows:

$$\overline{\boldsymbol{V}}^{(p)}=\overline{\boldsymbol{W}}_{\hat{v}}^{(p)\mathrm{T}}\hat{\boldsymbol{V}}\in\mathbb{R}^{D_p\times N}.\tag{15}$$

We use P users as we defined earlier. Thus, we can obtain a set of P optimal projection matrices. Finally, these matrices are integrated by the following concatenation:

$$\overline{\boldsymbol{V}}=\begin{bmatrix}\overline{\boldsymbol{V}}^{(1)}\\\overline{\boldsymbol{V}}^{(2)}\\\vdots\\\overline{\boldsymbol{V}}^{(P)}\end{bmatrix}\in\mathbb{R}^{\overline{D}\times N},\tag{16}$$

where $\overline{D}=\sum_{p=1}^{P}D_p$. In this way, we calculate the integrated canonical video features from the collaborative use of multiple users' viewing behavior. The use of other users' viewing behavior, not just the target user's viewing behavior, is supportive for estimation of the preference for a video that the target user has not seen yet.

4. SVORIM-Based Score Prediction for a Video

In this section, we explain the SVORIM-based score prediction method for a video. We first explain the reason why SVORIM [27] is used in our method as a classifier to predict evaluation scores for a video. In [36], sixteen kinds of ordinal regression classification methods are compared in terms of computation time, Mean Absolute Error (MAE) and Mean Zero-one Error (MZE) shown as:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{N_t}\sum_{i=1}^{N_t}\left|l_i^{\mathrm{Pre}} - l_i^{\mathrm{GT}}\right|, \\ \mathrm{MZE} &= \frac{1}{N_t}\sum_{i=1}^{N_t}[\![l_i^{\mathrm{Pre}} \neq l_i^{\mathrm{GT}}]\!], \end{aligned} \tag{17}$$

where N_t is the number of test samples, l_i^{Pre} is the predicted evaluation score of the ith sample, and l_i^{GT} is the ground truth (real score evaluated by the user) of the ith sample. Moreover, $[\![\cdot]\!]$ is a Boolean expression that outputs one if the inner condition $l_i^{\mathrm{Pre}} \neq l_i^{\mathrm{GT}}$ is true, otherwise zero. The range of MAE values is from zero to $R-1$, and that of MZE values is from zero to 1. Thus, we can compare an average error by using MAE and accuracy without considering the order by using MZE. The lower their values are, the higher the performance is. In [36], it has been shown that SVORIM is one of the most effective ordinal regression methods in terms of computation time, MAE and MZE†. Thus, we apply SVORIM to predict evaluation scores.

Next, we explain how to predict evaluation scores for a video based on SVORIM. This classifier is trained by the integrated canonical video features $\overline{\boldsymbol{V}}$ obtained in the previous section. First, we define pairs of integrated canonical video feature vectors and evaluation scores of the training video $(\overline{\boldsymbol{v}}_i, l_i)$ $(i = 1, 2, \cdots, N)$, where $l_i \in \{1, 2, \cdots, R\}$. Then $\overline{\boldsymbol{v}}_i$ are mapped into a high-dimensional Reproducing Kernel Hilbert Space (RKHS) to obtain $\phi(\overline{\boldsymbol{v}}_i)$. The prediction of evaluation scores for new test video vectors $\overline{\boldsymbol{v}}^{(new)}$ can be calculated by the following discriminant function:

$$\begin{aligned} &\arg\min_i \left\{ i : f(\overline{\boldsymbol{v}}^{(new)}) < \tau_i \right\}, \\ &f(\overline{\boldsymbol{v}}^{(new)}) = \left\langle \boldsymbol{u} \cdot \phi(\overline{\boldsymbol{v}}^{(new)}) \right\rangle, \end{aligned} \tag{18}$$

where $\langle\cdot\rangle$ denotes the inner product in the RKHS, $\boldsymbol{u}$ is a mapping direction, and τ_i are thresholds between classes. In order to obtain the optimal $\boldsymbol{u}$ and τ_i, we solve the following primal problem:

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{\tau},\boldsymbol{\xi},\boldsymbol{\xi}^*} \ & \frac{1}{2}\langle \boldsymbol{u} \cdot \boldsymbol{u} \rangle + C\sum_{r=1}^{R-1}\left(\sum_{\hat{r}=1}^{r}\sum_{i=1}^{N^{\hat{r}}}\xi_{\hat{r}i}^{r} + \sum_{\hat{r}=r+1}^{R}\sum_{i=1}^{N^{\hat{r}}}\xi_{\hat{r}i}^{*r}\right) \\ \text{s.t.} \quad & \left\langle \boldsymbol{u} \cdot \phi(\overline{\boldsymbol{v}}_i^{\hat{r}}) \right\rangle - \tau_r \leq -1 + \xi_{\hat{r}i}^{r},\ \xi_{\hat{r}i}^{r} \geq 0, \\ & \text{for } \hat{r} = 1, \cdots, r \ \text{ and } \ i = 1, \cdots, N^{\hat{r}}; \\ & \left\langle \boldsymbol{u} \cdot \phi(\overline{\boldsymbol{v}}_i^{\hat{r}}) \right\rangle - \tau_r \geq 1 - \xi_{\hat{r}i}^{*r},\ \xi_{\hat{r}i}^{*r} \geq 0, \\ & \text{for } \hat{r} = r+1, \cdots, R \ \text{ and } \ i = 1, \cdots, N^{\hat{r}}, \end{aligned} \tag{19}$$

where $C > 0$ is a constant variable to suppress over-learning from the small number of samples, $N^{\hat{r}}$ is the number of samples in class $\hat{r}$, and $\overline{\boldsymbol{v}}_i^{\hat{r}} \in \mathbb{R}^D$ is the canonical video feature vector belonging to class $\hat{r}$. Furthermore, $\xi_{\hat{r}i}^{r}$ and $\xi_{\hat{r}i}^{*r}$ are slack variables corresponding to the left and right parts for the rth parallel hyperplane, respectively. The first group of constraints corresponds to the left part of the $\hat{r}$th hyperplanes, and the second group of constraints corresponds to the right part. The schematic view of SVORIM is shown in the right side of Fig. 1. In this figure, the horizontal axis is $\langle \boldsymbol{u} \cdot \phi(\overline{\boldsymbol{v}}) \rangle$ in Eq. (18). In addition, vertical solid lines and dotted lines are τ_i and $\tau_i \pm 1$ in Eq. (19), respectively. Moreover, blue arrows are slack variables, and points flamed in green are support vectors.

Next, the dual problem of Eq. (19) can be rewritten as the following maximization problem:

$$\begin{aligned} \max_{\boldsymbol{\eta},\boldsymbol{\eta}^*} \ & -\frac{1}{2}\sum_{\hat{r},i}\sum_{\hat{r}',i'}\left(\sum_{r=1}^{\hat{r}-1}\eta_{\hat{r}i}^{*r} - \sum_{r=\hat{r}}^{R-1}\eta_{\hat{r}i}^{r}\right)\left(\sum_{r=1}^{\hat{r}'-1}\eta_{\hat{r}'i'}^{*r} - \sum_{r=\hat{r}'}^{R-1}\eta_{\hat{r}'i'}^{r}\right) \\ & \times \mathcal{K}(\overline{\boldsymbol{v}}_i^{\hat{r}}, \overline{\boldsymbol{v}}_{i'}^{\hat{r}'}) + \sum_{\hat{r},i}\left(\sum_{r=1}^{\hat{r}-1}\eta_{\hat{r}i}^{*r} + \sum_{r=\hat{r}}^{R-1}\eta_{\hat{r}i}^{r}\right) \\ \text{s.t.} \quad & \sum_{\hat{r}=1}^{r}\sum_{i=1}^{N^{\hat{r}}}\eta_{\hat{r}i}^{r} = \sum_{\hat{r}=r+1}^{R}\sum_{i=1}^{N^{\hat{r}}}\eta_{\hat{r}i}^{*r} \ \ \forall r, \\ & 0 \leq \eta_{\hat{r}i}^{r} \leq C \quad \forall r \text{ and } \hat{r} \leq r, \\ & 0 \leq \eta_{\hat{r}i}^{*r} \leq C \quad \forall r \text{ and } \hat{r} > r, \end{aligned} \tag{20}$$

where $\eta_{\hat{r}i}^{r}$ and $\eta_{\hat{r}i}^{*r}$ are Lagrangian multipliers, and $\mathcal{K}(\overline{\boldsymbol{v}}_i^{\hat{r}}, \overline{\boldsymbol{v}}_{i'}^{\hat{r}'}) = \langle \phi(\overline{\boldsymbol{v}}_i^{\hat{r}}) \cdot \phi(\overline{\boldsymbol{v}}_{i'}^{\hat{r}'}) \rangle$. Finally, the discriminant function in Eq. (18) can be rewritten as the following function by using the optimal $\eta_{\hat{r}i}^{r}$ and $\eta_{\hat{r}i}^{*r}$:

$$\begin{aligned} &\arg\min_i \left\{ i : f(\overline{\boldsymbol{v}}^{(new)}) < \tau_i \right\}, \\ &f(\overline{\boldsymbol{v}}^{(new)}) = \sum_{\hat{r},i}\left(\sum_{r=1}^{\hat{r}-1}\eta_{\hat{r}i}^{*r} - \sum_{r=\hat{r}}^{R-1}\eta_{\hat{r}i}^{r}\right)\mathcal{K}(\overline{\boldsymbol{v}}_i^{\hat{r}}, \overline{\boldsymbol{v}}^{(new)}). \end{aligned} \tag{21}$$

In this way, we can predict the target user's evaluation scores for a video based on SVORIM using the integrated canonical video features.

5. Experimental Results

In this section, we show experimental results to confirm the effectiveness of our method. First, we explain the experimental conditions. In this experiment, three keywords, "movie", "news" and "sports", were given as queries to YouTube. Five video clips were obtained for each keyword; that is, 15 video clips (65 seconds for each video) in total were prepared for the experiment. The subjects were eight men and two women of about 22 years of age. We set

† Among 16 kinds of methods, it has been reported in [36] that SVORIM recorded the sixth fastest computation time, the third lowest MAE and the second lowest MZE.

Table 2 Specific procedures and conditions for video and viewing behavior features used in the proposed method and the five comparative methods.

	PM	CM1	CM2	CM3	CM4	CM5
Video	SMSE	Vector concat.	SMSE	Vector concat.	SMSE	Vector concat.
Viewing behavior	Multiple users	Multiple users	Target user	Target user	-	-

Table 3 MAE and MZE values of the proposed method and the five comparative methods.

	PM		CM1		CM2		CM3		CM4		CM5	
Subject	MAE	MZE	MAE	MZE	MAE	MZE	MAE	MZE	MAE	MZE	MAE	MZE
1	**0.618**	**0.473**	0.719	0.528	0.645	**0.473**	0.738	0.498	0.759	0.578	1.098	0.849
2	0.589	0.484	**0.569**	**0.480**	0.622	0.488	0.619	0.496	0.643	0.534	0.900	0.766
3	**0.707**	**0.519**	0.713	0.535	0.744	0.539	0.771	0.551	0.808	0.582	1.102	0.738
4	**0.660**	**0.499**	0.747	0.528	0.735	0.545	0.844	0.523	0.756	0.544	1.232	0.777
5	**0.474**	**0.378**	0.505	0.412	0.515	0.408	0.542	0.428	0.602	0.488	0.710	0.584
6	**0.546**	**0.466**	0.549	0.494	0.568	0.482	0.557	0.496	0.571	0.496	0.734	0.641
7	**0.417**	**0.370**	0.482	0.416	0.433	0.389	0.535	0.445	0.509	0.444	0.676	0.598
8	**0.617**	**0.452**	0.626	0.495	0.702	0.487	0.659	0.492	0.793	0.551	1.024	0.742
9	**0.431**	**0.369**	0.508	0.419	0.447	0.384	0.523	0.443	0.510	0.451	0.701	0.554
10	0.644	**0.506**	**0.630**	0.518	0.695	0.532	0.677	0.534	0.728	0.578	1.009	0.704
Average	**0.570**	**0.452**	0.605	0.483	0.611	0.473	0.646	0.491	0.668	0.525	0.919	0.695

the above experimental conditions using [18], [19] as references.

The subjects watched all of the video clips in a sitting position. A 15-inch display was set at a distance of one meter from the subjects, and a Kinect sensor to extract their viewing behavior was set on the display. Then the subjects evaluated all of the video clips by five ordinal grades, i.e., 5 (high preference), 4 (preference), 3 (undecided), 2 (low preference) and 1 (very low preference), by a console input using a keyboard. Note that three features (video, viewing behavior, label) used in our method were extracted by 10 fps, and we did not extract features for five seconds immediately after watching each video to avoid noise by the user. In this way, a dataset including the three features could be obtained.

Next, we explain the parameter settings. First, the time width s in Eq. (1) was set as one second, and the threshold in **3.1** was set as $\mathrm{Th}_\alpha = 10^{-3}$. In this experiment, we empirically set $D_{\hat{v}} = \min\{D_{v_1}, D_{v_2}, D_{v_3}\} = 64$, $\epsilon = 0.01$, $D_p = 4 < \min\{D_{\hat{v}}, D_b, D_l\}$ for all subjects, and the number of users to integrate the canonical video features in Eq. (16) was set as $P = 10$ (all users including the target user). Moreover, the constant variable C in Eq. (19) was chosen by searching the following parameters: $C \in [2^{-5}, 2^{-3}, 2^{-1}, \cdots, 2^5, 2^7]$. Additionally, we adopted the Gaussian kernel in Eq. (20) as follows:

$$\mathcal{K}(\bar{\boldsymbol{v}}_i^{\hat{r}}, \bar{\boldsymbol{v}}_i^{\hat{r}'}) = \exp\left(\frac{-\|\hat{\boldsymbol{v}}_i^{\hat{r}} - \hat{\boldsymbol{v}}_i^{\hat{r}'}\|^2}{2\sigma^2}\right), \tag{22}$$

where the kernel width σ^2 was chosen by searching the following parameters: $\sigma^2 \in [2^{-15}, 2^{-13}, 2^{-11}, \cdots, 2^1, 2^3]$. We then decided an optimal set of two parameters by a grid search [37]. In this experiment, we conducted 15-fold cross-validation and compared the performance of our method with the performances of comparative methods by using MAE and MZE described in the previous section.

Next, we explain the comparative methods. We compared the Proposed Method (PM) with five Comparative Methods (CMs) as shown in Table 2. Note that "Vector concat." means the vector concatenation of the three kinds of features used as video features (Audio, HSVCH and SURF-Bof). CM1 is a method based on [28], which is the latest method in our previous work. CM3 is a method using not multiple users' viewing behavior but only the target user's viewing behavior [23]. CM4 and CM5 are methods not using any user's viewing behavior, and the methods maximize correlations between video features and label features based on Canonical Correlation Analysis [8]. Note that SMSE was used in CM4 but was not used in CM5. Moreover, we adopted Welch's t-test to determine whether the difference between "MAE and MZE of the PM" and "those of CMs" was significant or not.

The results presented in Table 3 show the effectiveness of the PM since we can see that MAE and MZE of the PM are lower than those of all CMs for the most part. In addition, we confirmed that the PM met the significance level of 5% from Welch's t-test compared to all of the CMs for both MAE and MZE. Specifically, by comparing CM3 with CM5 and comparing CM2 with CM4, it was confirmed that the target user's viewing behavior is effective. Next, by comparing CM4 with CM5 and comparing CM2 with CM3, it was confirmed that SMSE is also effective. Moreover, a comparison of CM1 and CM3 showed that the use of multiple users' viewing behavior is effective. Finally, we could confirm the effectiveness of our method by comparing the PM with CM1. The results indicate that our method can more accurately extract video features that reflect the target user's preference.

Next, we discuss the reason why SMSE and multiple users' viewing behavior contributed to good performance. First, we discuss the contribution of using SMSE. In our method, video features (Audio, HSVCH and SURF-Bof) are integrated by SMSE before integration of three kinds of fea-

Table 4 A comparison of "sMVCCA" and "SLPCCA and DLPCCA".

	SLPCCA		DLPCCA	
Subject	MAE	MZE	MAE	MZE
1	1.062	0.763	1.071	0.696
2	0.849	0.696	0.899	0.765
3	1.028	0.641	1.100	0.710
4	1.232	0.777	1.232	0.776
5	0.706	0.579	0.708	0.583
6	0.689	0.613	0.689	0.608
7	0.676	0.598	0.676	0.598
8	1.023	0.737	1.024	0.742
9	0.701	0.554	0.701	0.554
10	1.008	0.704	1.009	0.704
Average	0.897	0.666	0.911	0.674

tures (video, viewing behavior and label) using sMVCCA. By comparing PM with CM1 and comparing CM2 with CM3 in the experiment, it was confirmed that SMSE was effective for the most part. If video features are used as concatenation of each vector, the dimensionality of them is very high (=1209 dim.), which may cause the drop of generalization performance. Since SMSE has greatly reduced the dimensionality of the video features from 1209 dim. to 64 dim., the original video features have been transformed into embedding video features, which is useful condensed information for a target user. Therefore, the embedding to a low-dimensional space contributed to good performance.

Second, we discuss the contribution of using multiple users' viewing behavior. Our method uses viewing behavior obtained from multiple users, not just a target user. By comparing PM with CM2 and comparing CM1 with CM3 in the experiment, it was confirmed that the collaborative use of viewing behavior obtained from multiple users was effective for the most part. Specifically, it is conceivable that the viewing behavior of other users was supportive for viewing behavior of the target user when much viewing behavior for specific videos was not caused by the target user.

Moreover, we discuss which of video features and viewing behavior features is the most important features. Actually, which of these features is important depends on each user. Our method uses sMVCCA for correlation analysis, which maximizes the sum of all pair correlations including video features and viewing behavior features. By using the correlation analysis method, our method can automatically concern about the differences depending on each user.

In Sect. 3, sMVCCA was used for correlation analysis, which integrated the following three kinds of features: video, viewing behavior and label. Besides sMVCCA, we conducted experiments using supervised locality preserving CCA (SLPCCA) [38] and discriminative locality preserving CCA (DLPCCA) [39]. SLPCCA and DLPCCA are one of the state-of-the-art methods of supervised multimodal CCA as well as sMVCCA. SLPCCA is a supervised version of locality preserving CCA (LPCCA) [40], which introduces locality preserving projection (LPP) [41] to CCA. LPP calculates the projections preserving the neighborhood structure between two kinds of features. On the other hand, DLPCCA is the method extending LPCCA by Fisher discriminant analysis (FDA) [9], which minimizes intra-class variance and maximizes inter-class variance. Experiment conditions are same as CM3. From a comparison of CM3 in Table 3 and Table 4, we can confirm that the method using sMVCCA is superior to the methods using SLPCCA and DLPCCA. In addition, it was confirmed that the sMVCCA-based method met the significance level of 1% from Welch's t-test. Therefore, it is appropriate to use sMVCCA for supervised multimodal correlation analysis in this method.

6. Conclusions

A method for accurate estimation of personalized video preference was presented in this paper. In order to estimate the target user's preference more accurately, the proposed method uses not only the target user's viewing behavior but also multiple users' viewing behavior. We showed the effectiveness of the proposed method from experimental results since estimation with a high level of accuracy was achieved by our method compared to the conventional methods.

Finally, we discuss remaining issues and future works. First, it is necessary to construct a framework to select users who will give viewing behavior that contributes to accuracy for the target user. Our method proposed in this paper is a framework using all other users, which is not realistic when used in actual applications. Next, hand-crafted features such as HSVCH and SURF-Bof are used as video features. However, since videos are time-series data, we have an idea using features generated from a framework that is suitable for time change like recurrent neural network (RNN) [42].

Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Numbers JP17H01744, JP15K12023.

References

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," IDC iView: IDC Analyze the future, vol.2007, pp.1–16, 2012.

[2] M. Haseyama, T. Ogawa, and N. Yagi, "A review of video retrieval based on image and video semantic understanding," ITE Trans. Media Technology and Applications, vol.1, no.1, pp.2–9, 2013.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Trans. Knowledge and Data Engineering, vol.17, no.6, pp.734–749, 2005.

[4] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Trans. Information Systems, vol.22, no.1, pp.5–53, 2004.

[5] H.S. Tan and H.W. Ye, "A collaborative filtering recommendation algorithm based on item classification," Proc. Pacific-Asia Conf. Circuits, Communications and System (PACCS), 2009, pp.694–697.

[6] M.J. Pazzani and D. Billsus, "Content-based recommendation systems," The Adaptive Web, pp.325–341, 2007.

[7] H. Li, F. Cai, and Z. Liao, "Content-based filtering recommendation algorithm using HMM," Proc. Int. Conf. Computational and Information Sciences (ICCIS), pp.275–277, 2012.

[8] H. Hotelling, "Relations between two sets of variates," Biometrika, vol.28, no.3/4, pp.321–377, 1936.

[9] K. Fukunaga, "Introduction to statistical pattern recognition," Academic Press, 1990.

[10] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," Proc. IEEE Signal Processing Society Workshop, pp.41–48, 1999.

[11] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis - a brief tutorial," Institute for Signal and Information Processing, Department of Electrical and Computer Engineering. Mississippi State University, vol.18, pp.1–8, 1998.

[12] W. Max, "Fisher linear discriminant analysis," Department of Computer Science, University of Toronto, vol.3, pp.1–4, 2005.

[13] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," Journal of Machine Learning Research, vol.8, pp.1027–1061, May 2007.

[14] R. Sawata, T. Ogawa, and M. Haseyama, "Novel favorite music classification using EEG-based optimal audio features selected via KDLPCCA," Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), pp.759–763, 2016.

[15] S.K. Hadjidimitriou and L.J. Hadjileontiadis, "EEG-based classification of music appraisal responses using time-frequency analysis and familiarity ratings," IEEE Trans. Affective Computing, vol.4, no.2, pp.161–172, 2013.

[16] H. Gunes, B. Schuller, M. Pantic, and R. Cowie, "Emotion representation, analysis and synthesis in continuous space: A survey," Proc. IEEE Int. Conf. Automatic Face Gesture Recognition and Workshops (FG), pp.827–834, 2011.

[17] K.H. Kim, S.W. Bang, and S.R. Kim, "Emotion recognition system using short-term monitoring of psysiological signals," Medical and Biological Engineering and Computing, vol.42, no.3, pp.419–427, 2004.

[18] J. Moon, Y. Kim, H. Lee, C. Bae, and W.C. Yoon, "Extraction of user preference for video stimuli using EEG-based user responses," Electronics and Telecommunications Research Institute Journal, vol.35, no.6, pp.1105–1114, 2013.

[19] A. Yazdani, J.-S. Lee, J.-M. Vesin, and T. Ebrahimi, "Affect recognition based on physiological changes during the watching of music video," ACM Trans. Interactive Intelligent Systems, vol.2, no.1, pp.7:1–7:26, 2012.

[20] M.I. Posner, "Orienting of attention," The Quarterly Journal of Experimental Psychology, vol.32, no.1, pp.3–25, 2007.

[21] M. Yamamoto, N. Nitta, and N. Babaguchi, "Automatic personal preference acquisition from TV viewer's behaviors," Proc. IEEE Int. Conf. Multimedia and Expo (ICME), pp.1165–1168, 2008.

[22] M. Takahashi, S. Clippingdale, M. Naemura, and M. Shibata, "Estimation of viewers' ratings of TV programs based on behaviors in home environments," Multimedia Tools and Applications, vol.74, no.19, pp.8669–8684, 2015.

[23] Y. Ito, T. Ogawa, and M. Haseyama, "Novel video feature-based favorite video estimation using users' viewing behavior and evaluation," Proc. IEEE Global Conf. Consumer Electronics (GCCE), pp.224–225, 2016.

[24] T. Ogawa, Y. Yamaguchi, S. Asamizu, and M. Haseyama, "Human-centered video feature selection via mRMR-SCMMCCA for preference extraction," IEICE Trans. Information and Systems, vol.E100-D, no.2, pp.409–412, 2017.

[25] S. Liu, L. Zhang, W. Cai, Y. Song, Z. Wang, L. Wen, and D.D. Feng, "A supervised multiview spectral embedding method for neuroimaging classification," Proc. IEEE Int. Conf. Image Processing (ICIP), pp.601–605, 2013.

[26] G. Lee, A. Singanamalli, H. Wang, M.D. Feldman, S.R. Master, N.N.C. Shih, E. Spangler, T. Rebbeck, J.E. Tomaszewski, and A. Madabhushi, "Supervised multi-view canonical correlation analysis (sMVCCA): Integrating histologic and proteomic features for predicting recurrent prostate cancer," IEEE Trans. Medical Imaging, vol.34, no.1, pp.284–297, 2015.

[27] W. Chu and S.S. Keerthi, "Support vector ordinal regression," Neural Computation, vol.19, no.3, pp.792–815, 2007.

[28] Y. Ito, T. Ogawa, and M. Haseyama, "Personalized video preference estimation based on early fusion using multiple users' viewing behavior," Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), pp.3006–3010, 2017.

[29] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," Proc. Int. Conf. Digital Audio Effects (DAFX), pp.237–244, 2007.

[30] A.R. Smith, "Color gamut transform pairs," ACM Siggraph Computer Graphics, vol.12, no.3, pp.12–19, 1978.

[31] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," Workshop on Statistical Learning in Computer Vision, ECCV, vol.1, no.1/22, pp.1–2, 2004.

[32] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "Speeded-up robust features (SURF)," Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.

[33] J.F. Cohn, Z. Ambadar, and P. Ekman, "Observer-based measurement of facial expression with the facial action coding system," The Handbook of Emotion Elicitation and Assessment, pp.203–221, 2007.

[34] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics), vol.40, no.6, pp.1438–1446, 2010.

[35] J.C. Bezdek and R.J. Hathaway, "Some notes on alternating optimization," Proc. Int. Conf. Fuzzy Systems (AFSS), vol.2275, pp.288–300, 2002.

[36] P.A. Gutierrez, M. Perez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martinez, "Ordinal regression methods: Survey and experimental study," IEEE Trans. Knowledge and Data Engineering, vol.28, no.1, pp.127–146, 2016.

[37] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Technical Report, Department of Computer Science, National Taiwan University, pp.1–16, 2003.

[38] J. Yang and X. Zhang, "Feature-level fusion of fingerprint and finger-vein for personal identification," Pattern Recognition Letters, vol.33, no.5, pp.623–628, 2012.

[39] X. Zhang, N. Guan, Z. Luo, and L. Lan, "Discriminative locality preserving canonical correlation analysis," Proc. Chinese Conf. Pattern Recognition (CCPR), vol.321, pp.341–349, 2012.

[40] T. Sun and S. Chen, "Locality preserving CCA with applications to data visualization and pose estimation," Image and Vision Computing, vol.25, no.5, pp.531–543, 2007.

[41] X. He and P. Niyogi, "Locality preserving projections," Proc. Advances in Neural Information Processing Systems (NIPS), pp.153–160, 2004.

[42] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," arXiv preprint arXiv:1409.2329, 2014.

Yoshiki Ito (S'16) received his B.S. degree in Electronics and Information Engineering from Hokkaido University, Japan in 2017. He is currently pursuing an M.S. degree at the Graduate School of Information Science and Technology, Hokkaido University. His research interests include multimodal signal processing. He is a student member of the IEICE and the IEEE.

Takahiro Ogawa (S'03–M'08) received the B.S., M.S., and Ph.D. degrees in Electronics and Information Engineering from Hokkaido University, Japan, in 2003, 2005, and 2007, respectively. He joined the Graduate School of Information Science and Technology, Hokkaido University, in 2008, where he is currently an Associate Professor. His research interests are multimedia signal processing and its applications. He has been an Associate Editor of the ITE Transactions on Media Technology and Applications. He is a member of the ACM, the EURASIP, the IEICE, and the ITE.

Miki Haseyama (S'88–M'91–SM'06) received her B.S., M.S. and Ph.D. degrees in Electronics from Hokkaido University, Japan in 1986, 1988 and 1993, respectively. She joined the Graduate School of Information Science and Technology, Hokkaido University as an associate professor in 1994. She was a visiting associate professor of Washington University, USA from 1995 to 1996. She is currently a professor in the Graduate School of Information Science and Technology, Hokkaido University. Her research interests include image and video processing and its development into semantic analysis. She has been a Vice President of the Institute of Image Information and Television Engineers, Japan (ITE), an Editor-in-Chief of ITE Transactions on Media Technology and Applications, a Director, International Coordination and Publicity of The Institute of Electronics, Information and Communication Engineers (IEICE). She is a member of the IEEE, IEICE, Institute of Image Information and Television Engineers (ITE) and Information Processing Society of Japan (IPSJ).