Learning Features with Structure-Adapting Multi-view Exponential Family Harmoniums

Yoonseop Kang, Taewoong Jang, and Seungjin Choi

Abstract— **Existing multi-view feature extraction methods are based on restrictive assumptions on the connections between feature vectors and input data. These assumptions damage the quality of learned features, and also require more effort on choosing right dimensions of feature vector components connected to each view. In this paper we present adaptive multi-view harmonium (SA-MVH) for multi-view feature extraction, where its each hidden node chooses the views to connect with while training phase via switch parameters. "Switch" parameters are multiplied to the connection weights of ordinary exponential family harmoniums (EFH) to decide the existence of connection between hidden nodes and views. With switch parameters, a SA-MVH automatically adapts its structure to achieve better representation of data distribution. The model can also be easily trained using the same training algorithms used for EFHs. Numerical experiments on synthetic and real-world datasets demonstrate the useful behavior of the SA-MVH, compared to the existing multi-view feature extraction methods.**

I. Introduction

THE performances of machine learning algorithms for various tasks including classification, clustering, and retrieval are significantly affected by the choice of data features [1] [2]. Learning features from data instead of using data-independent features like SIFT, TF-IDF, and MFCC further improves the results of the machine learning algorithms.

Unlike usual data, some data have more than single possible representation. For example, a video with sound is composed of visual and aural information, and a hypertext document can be represented using the word occurrences and a list of documents it is linked from. These kinds of data are called 'multi-view' data [3]. Making use of multiple views on semi-supervised classification or clustering has been successful [3][4], but there were only a few attempts for using multi-view data for unsupervised feature extraction [5][6][7].

Earlier multi-view feature extraction methods including canonical correlation analysis (CCA) [8] and dual-wing harmonium (DWH) [5] assume that all views are completely independent given a set of latent variables shared by all views. In other words, these methods assume that all views can be described using a single set of feature vector. However, views of real-world data are not completely correlated nor completely independent from each other. The performance of the earlier multi-view feature extraction algorithms are often degraded when given partially correlated multi-view data.

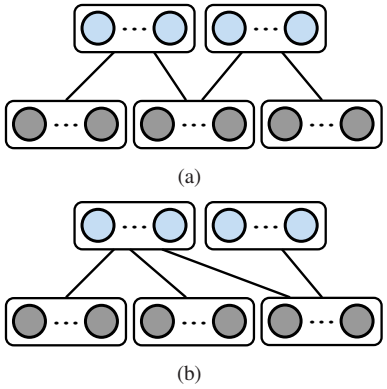


Fig. 1. Examples of partial correlations of three-view data that cannot be modeled by existing multi-view feature extraction methods.

More recent algorithms including factorized orthogonal latent spaces (FOLS) [6] or multi-view harmoniums [7] are based on a less restrictive assumption. These algorithms introduce view-specific feature vectors for all views in addition to the existing shared ones, assuming that views are generated from two sets of latent variables: view-specific latent variables and shared ones.

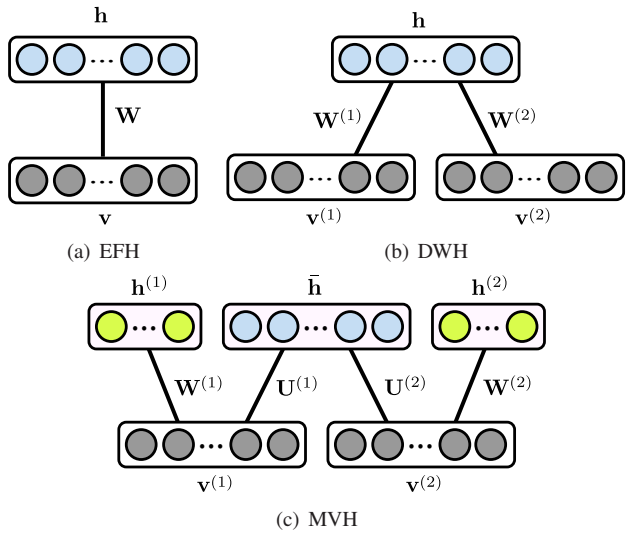
Drawbacks of multi-view feature extraction models with shared and view-specific feature vectors are not evident with two-view data, but the drawback becomes visible when we deal with data with three or more views. These models only allow the views to be generated from completely shared and completely view-specific latent variables. For example, these models do not allow a feature variable to be shared by view 2 and view 3, while not being shared with view 1 (Figure 1). Moreover, deciding dimensions of shared feature vectors and view-specific vectors is also a time-consuming task, and there is no well-established method for it.

To avoid all of these problems, we take a different approach from existing multi-view feature extraction models. Instead of separately defining view-specific and shared feature vectors in prior to the training of model, we only use one set of feature vector and let each dimension of the feature vector to decide the existence of connections to views during the training phase. This approach eliminates the need for choosing the number of view-specific latent variables. Moreover, this approach enables the proposed model to capture partial correlation among views.

In this work, we propose structure-adapting multi-view harmonium (SA-MVH), which is a multi-view feature extrac-

Yoonseop Kang, Taewoong Jang, and Seungjin Choi are with the Department of Computer Science and Engineering, Pohang University of Science and Technology, 77 Cheongam-ro, Nam-gu, Pohang 790-784, Korea (email: {e0en,seungjin}@postech.ac.kr).

This work was supported by National Research Foundation (NRF) of Korea (NRF-2013R1A2A2A01067464).



(a) EFH (b) DWH (c) MVH

Fig. 2. Graphical models of (a) exponential family harmonium, (b) dual-wing harmonium and (c) multi-view harmonium.

tion method based on exponential family harmonium (EFH) [9] that automatically decides the existence of connections between views and feature vector elements. We first review EFH that our model is based on, and also review the structure of existing multi-view models and discuss their abilities and limits. In Section 3, we introduce the proposed model, SA-MVH and describe its structure and the training algorithm for the method. Then we qualitatively and quantitatively compare the existing multi-view feature extraction methods to SA-MVH with various experiments including feature extraction from two synthetic datasets, single-label and multi-label image classification in Section 4. Finally, Conclusions are drawn in Section 5.

II. Related Works

Exponential family harmonium (EFH) is a two-layered, stochastic unsupervised neural network composed with a layer of visible input nodes $\boldsymbol{v}$ and other layer of hidden nodes $\boldsymbol{h}$ and their connection weights [9] (Figure 2-(a)). An EFH can also be interpreted as a bipartite probabilistic graphical model that the joint probability of nodes is proportional to the exponential of a quadratic energy function. To define energy function and joint probability, we need to first define marginal distribution of nodes $\boldsymbol{v}$ and $\boldsymbol{h}$ as exponential family distributions:

$$p(v_i) \propto \exp(\sum_a \xi_{ia} f_{ia}(v_i) - A_i(\{\xi_{ia}\})) \quad (1)$$

$$p(h_j) \propto \exp(\sum_b \lambda_{jb} g_{jb}(h_j) - B_j(\{\hat{\lambda}_{jb}\})), \quad (2)$$

where $f_{ia}(\cdot)$, $g_{jb}(\cdot)$ are sufficient statistics of $\boldsymbol{v}$ and $\boldsymbol{h}$, and ξ, λ are their parameters. A and B are log-partition functions of the marginal distributions. With the marginal distributions and quadratic terms that denotes the relation between $\boldsymbol{v}$ and $\boldsymbol{h}$, the energy function and the joint distribution of an EFH are derived as below:

$$\begin{aligned} E(\boldsymbol{v}, \boldsymbol{h}; \theta) &= -\sum_{i,a} \xi_{ia} f_{ia}(v_i) - \sum_{j,b} \lambda_{jb} g_{jb}(h_j) \\ &\quad \sum_{i,a,j,b} \boldsymbol{W}_{iajb} f_{ia}(v_i) g_{jb}(h_j) \quad (3) \\ p(\boldsymbol{v}, \boldsymbol{h}; \theta) &\propto \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \theta)), \quad (4) \end{aligned}$$

where $\theta = \{\boldsymbol{W}, \xi, \lambda\}$.

An EFH can be extended to be used for feature extraction of multi-view data. The most natural and simple approach is connect multiple sets of inputs to a single set of hidden nodes $\boldsymbol{h}$ shared across all input views $\{\boldsymbol{v}^{(k)}\}_{k=1}^K$ (Figure 2-(b)). Among existing feature extraction methods, some methods including CCA and Dual-wing harmonium (DWH) took this approach, and they proved their usefulness on document classification and image retrieval [10] [5].

Dual-wing harmonium (DWH) extends EFH by introducing multiple set of visible nodes $\{\boldsymbol{v}^{(k)}\}_1^K$ connected to a single set of hidden nodes $\boldsymbol{h}$. The energy function of DWH is defined as below:

$$\begin{aligned} &E(\{\boldsymbol{v}^{(k)}\}, \boldsymbol{h}; \theta) \\ &= -\sum_{k,i,j,a,b} \boldsymbol{W}^{(k)}_{iajb} f^{(k)}_{ia}(v^{(k)}_i) g_{jb}(h_j) \\ &- \sum_{k,i,a} \xi^{(k)}_{ia} f^{(k)}_{ia}(v^{(k)}_i) - \sum_{jb} \lambda_{jb} g_{jb}(h_j). \quad (5) \end{aligned}$$

The marginal distributions of each set of visible nodes $p(v_i^{(k)})$ of DWH can take different parametrization to reflect the characteristics of data assigned to them. For example, DWH can learn joint distribution of continuous data and discrete data by assigning a continuous distribution(i.e. Gaussian) for the continuous view, and a discrete distribution (i.e. Bernoulli) for the discrete one.

Recent multi-view feature extraction models including FOLS and multi-view harmonium (MVH) further extends the old approach by allowing a set of view-specific hidden nodes $\boldsymbol{h}^{(k)}$ for each set of inputs $\boldsymbol{v}^{(k)}$ in addition to the shared hidden nodes $\bar{\boldsymbol{h}}$ (Figure 2-(c)), so that these additional nodes can model uncorrelated information of data as well [6] [7].

MVH outperformed its predecessors in tasks like image reconstruction and image annotation, but the capability of the model is still limited by its restrictive, pre-defined structure [7].

In the next section, we propose a new feature extraction model and discuss how the new model overcomes the limitations of its predecessors.

III. The Proposed Model

The limitations of existing multi-view feature extraction models including DWH and MVH are evident. In those existing models, hidden nodes of the existing models have to be connected to either one specific view or all views. It is not possible for a hidden node to model a partial correlation between views (i.e. correlation between view 1 and 2 only in a 3-view data).

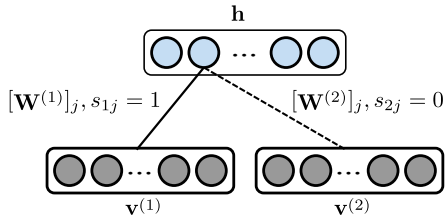


Fig. 3. Graphical models of structure-adapting multi-view harmonium (SA-MVH). Solid line: connections turned on by switch parameters, dashed line: connections turned off by switch parameters.

Moreover, Existing models with view-specific hidden nodes also suffer from the problem of deciding right number of latent variables. With K views, we need to specify K more parameters about number of latent variables for multi-view models in addition to the parameters for single-view models.

The main cause of these limitations is that we pre-define connection structure of inputs and feature vectors. Therefore, we need a model that does not constrain itself to the pre-defined structure, and learns its structure from its inputs.

A. Switch Parameters

As Mentioned above, the problem of restrictive assumptions on connections can be solved if we learn the connections between views and hidden nodes at the training time. The problem of choosing right number of parameters also banishs if we specify only a single set of hidden nodes. Therefore, these two problems can be solved at once with a model with single set of hidden nodes that each of them selects the connection to views automatically.

The definition of the proposed model, structure-adapting multi-view harmonium (SA-MVH) starts from defining marginal distributions of visible node sets and a set of hidden nodes:

$$\begin{aligned} p(v_i^{(k)}) &\propto \exp(\sum_a \xi_{ia} f_{ia}^{(k)}(v_i^{(k)}) - A_i^{(k)}(\{\xi_{ia}^{(k)}\})), \\ p(h_j) &\propto \exp(\sum_b \lambda_{jb} g_{jb}(h_j) - B_j(\{\lambda_{jb}\})). \end{aligned} \tag{6}$$

For real-valued visible nodes with Gaussian distribution, the sufficient statistics, parameters, and log-partition functions are as below:

$$\{f_{ia}^{(k)}\} = [v_i^{(k)}, v_i^{(k)2}]^\top, \tag{7}$$

$$\{\xi_{ia}^{(k)}\} = [\xi_i^{(k)}, -\frac{1}{2}]^\top, \tag{8}$$

$$A_i^{(k)}(\{\xi_i^{(k)}\})) = -\frac{\xi_i^{(k)2} + \log 2\pi}{2}. \tag{9}$$

On the other hand, binary-valued visible nodes with Bernoulli distribution have the sufficient statistics, parameters, and log-partition functions as below:

$$\{f_{ia}^{(k)}\} = v_i^{(k)}, \tag{10}$$

$$\{\xi_{ia}^{(k)}\} = \xi_i^{(k)}, \tag{11}$$

$$A_i^{(k)}(\{\xi_i^{(k)}\})) = \log(1 + \exp(\xi_i^{(k)2})). \tag{12}$$

Just as DWH, the visible nodes and hidden nodes of SA-MVH are connected with weight matrices $\boldsymbol{W}^{(k)}$. However, we need an additional mechanism to enable our model to adapt its structure to the given data distribution.

To solve this problem, we introduce binary switch parameters $s_{kj} \in \{0, 1\}$ that encode the connection structure of model. If s_{kj} is 1, k-th view and hidden node h_j are connected to nonzero weights, and the view and hidden node are disconnected when s_{kj} is 0.

More specifically, each column of connection weight matrices $[\boldsymbol{W}^{(k)}]_j = \left[\boldsymbol{W}_{1j}^{(k)}, \ldots, \boldsymbol{W}_{Dj}^{(k)}\right]^\top$ is turned on or turned off by being multiplied to the switch parameter s_{kj} (Figure 3). The proposed model extends DWH by introducing switch parameters as below:

$$\begin{aligned} &E(\{\boldsymbol{v}^{(k)}\}, \boldsymbol{h}; \theta, \{s_{kj}\}) \\ &= -\sum_{k,i,j} s_{kj} \boldsymbol{W}_{ij}^{(k)} f_i^{(k)}(\boldsymbol{v}_i^{(k)}) g_j(h_j) \\ &\quad - \sum_{k,i} \xi_i^{(k)} f_i^{(k)}(\boldsymbol{v}_i^{(k)}) - \sum_j \lambda_j g_j(h_j), \end{aligned} \tag{13}$$

note that indices a and b are omitted to keep the notations uncluttered.

However, the optimization involving s_{kj} with binary values is obviously intractable. Therefore we relax the problem and allow any real values for s_{kj} to make the optimization tractable. We also apply sigmoid function $\sigma(\cdot)$ to s_{kj} squash the switch parameters to the range between 0 and 1. By replacing s_{kj} by $\sigma(s_{kj})$, we get the final version of energy function of SA-MVH.

$$\begin{aligned} &E(\{\boldsymbol{v}^{(k)}\}, \boldsymbol{h}; \theta, \{s_{kj}\}) \\ &= -\sum_{k,i,j} \sigma(s_{kj}) \boldsymbol{W}_{ij}^{(k)} f_i^{(k)}(\boldsymbol{v}_i^{(k)}) g_j(h_j) \\ &\quad - \sum_{k,i} \xi_i^{(k)} f_i^{(k)}(\boldsymbol{v}_i^{(k)}) - \sum_j \lambda_j g_j(h_j). \end{aligned} \tag{14}$$

The inference on SA-MVH is simple because the model allows no within-layer connection. Therefore we can efficiently perform inference for feature extraction by evaluating the conditional distributions of nodes which is available in a closed form as below:

$$\begin{aligned} p(v_i^{(k)}|\boldsymbol{h}; \theta) &= \exp(\hat{\xi}_i^{(k)} f_i(\boldsymbol{v}_i^{(k)}) - A_i^{(k)}(\{\hat{\xi}_i^{(k)}\})) \\ p(h_j|\{\boldsymbol{v}^{(k)}\}; \theta) &= \exp(\hat{\lambda}_j g_j(h_j) - B_j(\{\hat{\lambda}_j\})), \end{aligned} \tag{15}$$

where the shifted parameters are

$$\hat{\xi}_i^{(k)} = \xi_i^{(k)} + \sum_j \sigma(s_{kj}) \boldsymbol{W}_{ij}^{(k)} g_j(h_j) \tag{16}$$

$$\hat{\lambda}_j = \lambda_j + \sum_{k,i} \sigma(s_{kj}) \boldsymbol{W}_{ij}^{(k)} f_i(\boldsymbol{v}_i^{(k)}). \tag{17}$$

At a first glance, introducing switch parameter may seem to be meaningless, because a SA-MVH with switch parameters s_{kj} and weights $\boldsymbol{w}_j^{(k)}$ is exactly equivalent to a DWH with weights $\hat{\boldsymbol{w}}_j^{(k)} = \boldsymbol{w}_j^{(k)} \sigma(s_{kj})$. However, introducing switch parameters affects the gradient of the objective

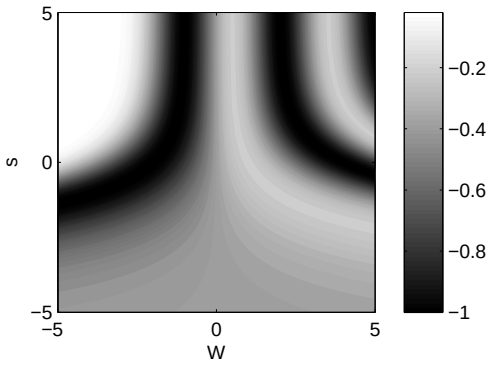


Fig. 4. Visualization of solution space of $\mathcal{L}(w,s) = \text{rbf}(w\sigma(s)-5) + \text{rbf}(w\sigma(s)+1) + \text{rbf}(w\sigma(s)-2)$, where $\text{rbf}(x) = \exp(-x^2)$. Lighter color corresponds to higher value of $\mathcal{L}$.

function. With a low value of switch parameter, the objective function is mostly affected by switch parameters, causing a gradient descent optimization change switch parameters and keep weight values almost unchanged. Conversely, when the switch parameter is sufficiently large, the objective function is mostly affected by weight values and gradient descent algorithm will change weight values (Figure 4).

Therefore, if we start training with sufficiently small value of switch parameters ($s_{kj} \leq -2$), the switch parameter will increase to a sufficient value, while the weight is also being optimized.

B. Training SA-MVH

We can train SA-MVHs by maximizing the likelihood of model via gradient ascent. The likelihood of SA-MVH is defined as the joint distribution of visible and hidden nodes summed over all possible values of hidden nodes $\boldsymbol{h}$:

$$\mathcal{L} = \langle \log p(\{\boldsymbol{v}^{(k)}\}) \rangle_{data} \quad (18)$$

$$= \left\langle \log \sum_{\boldsymbol{h}} p(\{\boldsymbol{v}^{(k)}\}, \boldsymbol{h}) \right\rangle_{data}, \quad (19)$$

where $\langle\cdot\rangle_{data}$ represents expectation over data distribution. Then the gradient of log-likelihood for the parameters $\boldsymbol{W}^{(k)}$, $\xi^{(k)}$, and λ is derived as follows:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ij}^{(k)}} \propto \left\langle \sigma(s_{kj}) f_i(\boldsymbol{v}_i^{(k)}) B_j'(\hat{\lambda}_j) \right\rangle_{data} - \left\langle \sigma(s_{kj}) f_i(\boldsymbol{v}_i^{(k)}) B_j'(\hat{\lambda}_j) \right\rangle_{model} \quad (20)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i^{(k)}} \propto \left\langle f_i^{(k)}(\boldsymbol{v}_i^{(k)}) \right\rangle_{data} - \left\langle f_i^{(k)}(\boldsymbol{v}_i^{(k)}) \right\rangle_{model} \quad (21)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} \propto \left\langle B_j'(\hat{\lambda}_j) \right\rangle_{data} - \left\langle B_j'(\hat{\lambda}_j) \right\rangle_{model}, \quad (22)$$

where $\langle\cdot\rangle_{model}$ represents expectation over model distribution $p(\boldsymbol{v}^{(k)}, \boldsymbol{h}; \theta)$.

To calculate the gradient exactly, we need to calculate exact partition function of the model distribution, and it requires exponential number of summations over every possible values of the nodes. This is definitely intractable. Instead, one can approximate model distribution by performing alternating gibbs sampling for a limited number of steps. Contrastive divergence learning well approximates the model distribution with samples obtained from gibbs chain initialized with data distribution [11]. With contrastive divergence learning, the training time becomes tractable.

Gradient of log-likelihood over switch parameters s_{kj} is as simple as other gradients due to the real-value relaxation.

$$\frac{\partial \mathcal{L}}{\partial s_{kj}} \propto \left\langle \sigma'(s_{kj}) \boldsymbol{W}_{ij}^{(k)} f_i(\boldsymbol{v}_i^{(k)}) B_j'(\hat{\lambda}_j) \right\rangle_{data} - \left\langle \sigma'(s_{kj}) \boldsymbol{W}_{ij}^{(k)} f_i(\boldsymbol{v}_i^{(k)}) B_j'(\hat{\lambda}_j) \right\rangle_{model} \quad (23)$$

where $\sigma'(\cdot)$ is the derivative of sigmoid function. Training of SA-MVH is done by repeating the gradient descent until the parameters are converged (see algorithm 1).

In addition, to discourage switch parameters being too soft – making connections half-on and half-off due to small absolute value of s_{kj}, we add a penalty term for s_{kj} in addition to log-likelihood $\mathcal{L}$ derived from the energy function above. To encourage hard decision, switch parameters $\boldsymbol{s}_j = [s_{1j}, s_{2j}, \ldots, s_{Kj}]$ of hidden node h_j has to have values away from zeros. Maximizing the difference of 1-norm and 2-norm exactly fits this purpose. Penalizing $|\boldsymbol{s}_j|_1 - |\boldsymbol{s}_j|_2$ will push away s_{kj} from zero, but it will not push it too hard also (Figure 5).

The objective function with this penalty term is as below:

$$\mathcal{L}_{penalty} = \mathcal{L} + \lambda \left(\sum_j |\boldsymbol{s}_j|_1 - |\boldsymbol{s}_j|_2 \right). \quad (24)$$

Algorithm 1 Training algorithm for SA-MVH.

Require: Training data $\boldsymbol{X} = \{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(K)}\}$, where each $\boldsymbol{X}^{(k)} = \{\boldsymbol{x}_1^{(k)}, \ldots, \boldsymbol{x}_N^{(k)}\}$ for $k = 1, \ldots, K$.

while parameters not converged **do**
  set $\hat{\boldsymbol{v}}_t^{(k)} = \boldsymbol{x}_t^{(k)}$.
  sample $\hat{\boldsymbol{h}}_t$ from $p(h_j | \{\boldsymbol{v}^{(k)}\}; \theta)$.
  for $i < K_{gibbs}$ **do**
    sample $\boldsymbol{v}_t^{(k)}$ from $p(v_i^{(k)} | \boldsymbol{h}; \theta)$.
    sample $\boldsymbol{h}_t$ from $p(h_j | \{\boldsymbol{v}^{(k)}\}; \theta)$.
  end for
  collect statistics from data distribution using samples $\hat{\boldsymbol{v}}_t^{(k)}$, $\hat{\boldsymbol{h}}_t$.
  collect statistics from model distribution using samples $\boldsymbol{v}_t^{(k)}$, $\boldsymbol{h}_t$.
  calculate gradients for the parameters using collected statics.
  update the parameters $\{\boldsymbol{W}^{(k)}, \xi^{(k)}, \lambda\}$.
end while

IV. Numerical Experiments

A. Discovering Latent Structure from Synthetic Data

To show the effectiveness of switch parameters, we performed an experiment taken from recent work of Salzmann et

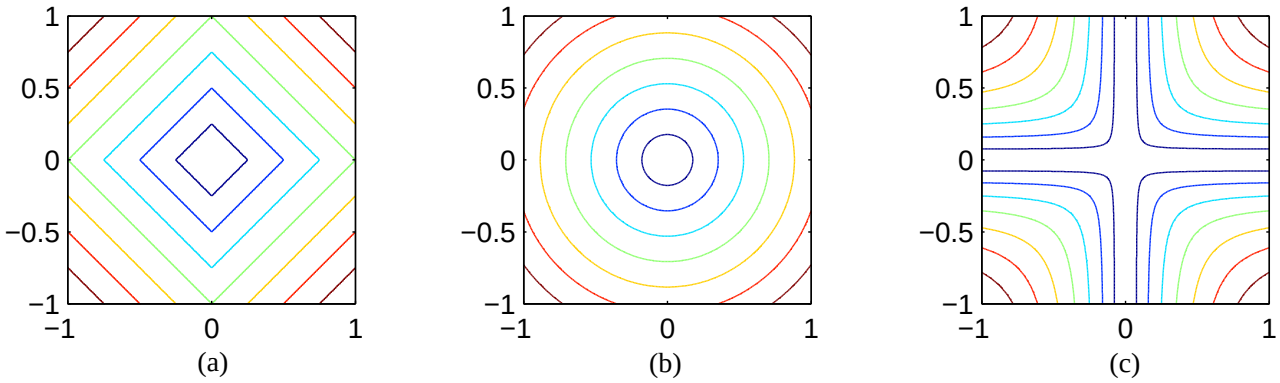


Fig. 5. Contour plot of the (a) 1-norm, (b) 2-norm, and (c) difference between 1-norm and 2-norm.

al. [6]. We constructed a synthetic, partially correlated multi-view dataset. To illustrate common and view-specific latent variables, we used sinusoidal functions of $\boldsymbol{t}$ with different phases and frequencies:

$$\boldsymbol{x} = \sin(2\pi \boldsymbol{t}), \quad \boldsymbol{z}^{(1)} = \cos(\pi\pi \boldsymbol{t}), \quad \boldsymbol{z}^{(2)} = \sin(5\pi \boldsymbol{t}),$$
$$\boldsymbol{m}^{(1)} = [\boldsymbol{x}, \boldsymbol{z}^{(1)}], \quad \boldsymbol{m}^{(2)} = [\boldsymbol{x}, \boldsymbol{z}^{(2)}].$$

We randomly projected $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$ to 20 dimensional space and added independent Gaussian noise of variance 0.01 and correlated noise $0.02\sin(3.6\pi \boldsymbol{t})$, and re-scaled the dataset to fit in the range between 0 and 1 to obtain our final multi-view synthetic dataset.

We trained DWH and SA-MVH with 3 hidden nodes for comparison. We also trained MVH with one view-specific hidden node for each view, and one shared hidden node. We assumed Bernoulli distribution for both visible and hidden nodes for all three models.

The models were trained using 2000 training samples, and we calculated hidden node activations from 1000 test samples. We checked the correspondence between ground-truth latent variables $\boldsymbol{x}$, $\boldsymbol{z}^{(1)}$, and $\boldsymbol{z}^{(2)}$, and hidden node activations obtained from feature extraction models.

Due to the correlated noise of the data, DWH failed to infer correct values of latent variables. Activations of two of three hidden node did not correspond to the latent variables used to generate data. On the other hand, SA-MVH found the correct connection structure from the given dataset, and also recovered the latent variables correctly. 2-view MVH also discovered latent variables correctly. However, Our model was able to separate common and view-specific information without any help of knowledge of latent structure of data given in prior, while MVH benefits from such knowledge.

B. Feature Extraction on Noisy Arabic-Roman Digit Dataset

To simulate the existence of view-specific and shared properties of multi-view data, we tested EFH and our model on a synthetic dataset designed for this purpose. This dataset, called Noisy Arabic-Roman digit dataset is a collection of 11,800 images of arabic and roman digits. Generation procedure of the dataset is as follows.

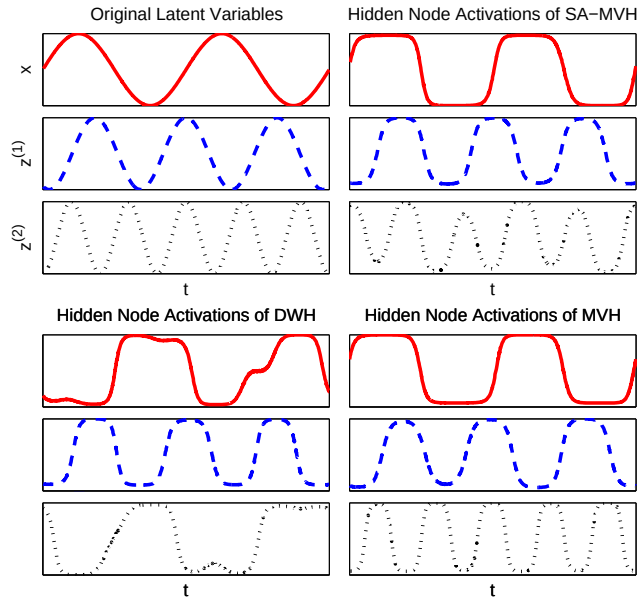
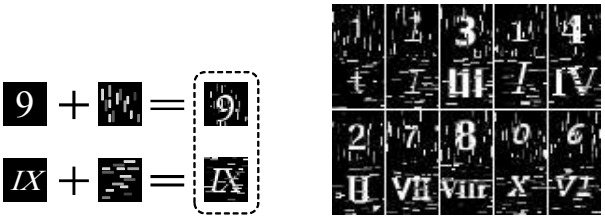


Fig. 6. Latent variables used to generate a synthetic data (top left) and hidden node activations of harmonium models. The hidden node activations were normalized to have zero mean and maximum absolute value to be 1.

First, we select a font and a number between 0 and 9. Then we create a pair of 28×28 pixel images of the number written in arabic and roman digits with the selected font, in white color on black background. As there is no roman number that corresponds to arabic number 0 (zero), we used symbol X as roman digit for number 0. Then we add vertical line noises with random length, position and intensity to the image of arabic digit images, and we add horizontal line noises to roman digits images in the similar way.

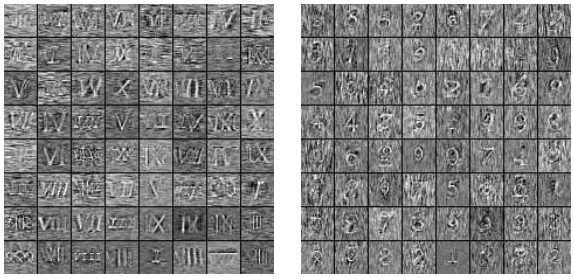
We repeat this procedure 10 times for every digit and 118 different fonts, to create 11,800 pairs of noisy digit images (Figure 7).

To compare the characteristics of SA-MVH and EFH, we trained these models with 200 hidden nodes. Bernoulli distribution was assumed for the both views of dataset, and the models were trained for 200 epochs with learning rate



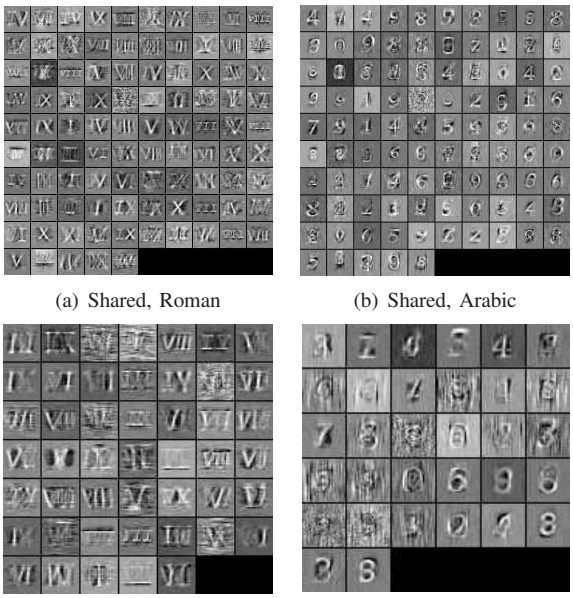
(a) (b)

Fig. 7. Construction procedure (a) and 10 examples from noisy arabic-roman digit dataset (b).



(a) Roman digit features (b) Arabic digit features

Fig. 8. Features of Noisy Arabic-Roman Digit Dataset learned by EFH with concatenated views. Only 64 features with largest 2-norms are presented.



(a) Shared, Roman (b) Shared, Arabic

(c) View-specific, Roman (d) View-specific, Arabic

Fig. 9. Shared and view-specific features of Noisy Arabic-Roman Digit Dataset learned by SA-MVH.

0.001, momentum 0.9, and batch size 100. After training, we visualized each column of connection weights $\boldsymbol{W}$ or ($\boldsymbol{W}^{(k)}$ for SA-MVH) for analysis.

Figure 8 shows features with largest 2-norms learned by EFH. The features were corrupted by horizontal and vertical noises in the dataset, indicating that EFH was not able to separate view-specific line noises from shared information, digits and fonts.

On the other hand, features learned by SA-MVH is a bit different 9. SA-MVH found 95 shared features, and 47 view-specific features for roman digits, and 32 view-specific features for arabic digits. Remaining 26 features were not connected to any views and ignored.

Most of the shared SA-MVH features were noise-free and encoded parts of roman and arabic numbers 9. On the other hand, the view-specific features had components with horizontal or vertical noises, as well as parts of the numbers. In this example, SA-MVH was able to automatically separate view-specific and shared information without any prior specification of the graph structure.

C. Image Classification

In addition to the feature learning on synthetic dataset, image classification on the datasets including CIFAR-10 and Caltech-256 datasets were done to examine the effectiveness of SA-MVH on real-world data.

- CIFAR-10 image database is a labeled subset of LabelMe dataset with 50,000 training and 10,000 test samples [12]. To simulate multi-view settings, we extracted two kinds of features from each images in the dataset. For a global representation of image, we extracted HSV histograms with 64 bins (8 for hue, 4 for saturation, 2 for value). As a local descriptor, we used bag of 101 visual words computed using colorSIFT algorithm [13]. Using global and local representation, we constructed two-view data out of CIFAR-10 dataset.
- Caltech-256 image dataset contains 30,608 labeled images from 256 categories and a "clutter" category [14]. Among 30,608 images, we picked 29,780 images from the 256 categories (excluding the "clutter" images), and extracted 512 dimensions of GIST and 1,536 dimensions of histogram of gradients (HoG) features. 19,780 samples were used for training feature extraction algorithms, and the remaining 10,000 samples were used for testing.

We compared our method, SA-MVH with other feature extraction methods based on EFH, including EFH itself, DWH, and MVH. We also compared our method to methods did not originated from EFH also. We used linear projection methods including principal component analysis (PCA) [15] as a baseline, and Sparse Filtering [16].

For the single-view methods including EFH, PCA, and Sparse Filtering, we concatenated multiple views into single view. Then the data is pre-processed by centering and re-scaling its dimensions to have standard deviation 1. On the other hand, Real-valued views of input data were assumed to follow Gaussian distribution, and the binary valued views were assumed to follow Bernoulli distribution in EFH-based multi-view feature extraction models.

All EFH-based models were trained for up to 200 epochs with had 1,000 hidden nodes, batch-size 128, learning rate 0.001, and momentum 0.9. For MVH, we assigned 10% of total hidden nodes as view-specific hidden nodes for each view, and used the remaining dimensions for shared hidden nodes. For SA-MVH, we set $\lambda = 0.01$ to penalize the norms

TABLE I

IMAGE CLASSIFICATION ACCURACY OF K-NEAREST NEIGHBOR CLASSIFIER ON FEATURES EXTRACTED BY VARIOUS FEATURE EXTRACTION METHODS TRAINED ON CIFAR-10 DATASET. FOR EACH VALUE OF k, THE BEST RESULT IS MARKED AS BOLD TEXT.

Method	# dim	10-NN	30-NN	50-NN	70-NN	100-NN
PCA (baseline)	10	0.205	0.222	0.227	0.227	0.233
Sparse Filtering	1000	0.225	0.237	0.24	0.242	0.241
EFH	1000	0.263	0.276	0.277	0.275	0.271
DWH	1000	0.213	0.228	0.236	0.238	0.236
MVH	1000	**0.322**	**0.334**	0.330	**0.330**	**0.329**
SA-MVH	1000	**0.322**	**0.334**	**0.335**	0.328	0.326

TABLE II

IMAGE CLASSIFICATION ACCURACY OF K-NEAREST NEIGHBOR CLASSIFIER ON FEATURES EXTRACTED BY VARIOUS FEATURE EXTRACTION METHODS TRAINED ON CALTECH-256 DATASET. FOR EACH VALUE OF k, THE BEST RESULT IS MARKED AS BOLD TEXT.

Method	# dim	10-NN	30-NN	50-NN	70-NN	100-NN
PCA (baseline)	10	0.164	0.173	0.172	0.168	0.165
Sparse Filtering	1000	0.161	0.165	0.163	0.16	0.155
EFH	1000	0.240	0.230	0.220	0.210	0.197
DWH	1000	0.237	0.231	0.217	0.207	0.194
MVH	1000	0.239	0.225	0.216	0.203	0.191
SA-MVH	1000	**0.246**	**0.232**	**0.223**	**0.212**	**0.198**

of switch parameter.

We trained the feature extraction methods and the features of training samples were extracted by those algorithms. Then we tested the quality of learned features with k-nearest neighbor classifiers. K-nearest neighbor classifiers with 10, 30, 50, 70, 100 neighbors for this experiment. Finally, we extracted features from test samples and measured the classification accuracy of each feature extraction algorithms. The classification accuracy with K-nearest neighbor classifier is shown on table I and table II.

SA-MVH model showed higher accuracy than other feature extraction models in all tested datasets, regardless of the values of k for nearest neighbor classifier. Although MVH also showed comparable result to SA-MVH on CIFAR-10 datasets, the method could not outperform other methods in Caltech-256 dataset. The linear projection methods failed to give comparable results to any EFH-based feature extraction models.

V. CONCLUSIONS

In this paper, we have proposed the multi-view feature extraction model that automatically decides relation between its latent variables and input views. The proposed method, SA-MVH models multi-view data distribution with less restrictive assumption and also reduce the number of parameters to tune by human hand. To achieve the useful properties, SA-MVH introduces 'switch parameters' that controls the connection between hidden nodes and input views, and finds the desirable configuration for it while training.

We have demonstrated the effectiveness of our approach by comparing our model to existing models including PCA, EFH, and MVH, on various experiments on synthetic dataset and simulated multi-view settings for image classification. On these experiments, we found significant improvement over other methods in the experiments in both qualitative and quantitative aspects.

In the future, we plan to investigate the modifications of SA-MVH model such as discriminative SA-MVH, or recurrent SA-MVH. We also plan to extend SA-MVH to deep network to model more complex relationship among views.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[3] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Annual Conference on Learning Theory (COLT)*, Madison, WI, 1998.

[4] I. Drost, S. Bickel, and T. Scheffer, "Discovering communities in linked data by multi-view clustering," in *Proceedings of Aunnual Conference on the German Classification Society*, 2005.

[5] E. P. Xing, R. Yan, and A. G. Hauptmann, "Mining associated text and images with dual-wing harmonium," in *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Edinburgh, UK, 2005.

[6] M. Salzmann, C. H. Ek, R. Urtasun, and T. Darrell, "Factorized orthogonal latent spaces," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy, 2010.

[7] Y. Kang and S. Choi, "Restricted deep belief networks for multi-view learning," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, Athens, Greece, 2011.

[8] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with applications to learning methods," *Neural Computation*, vol. 16, pp. 2639–2664, 2004.

[9] M. Welling, M. Rosen-Zvi, and G. Hinton, "Exponential family harmoniums with an application to information retrieval," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 17. MIT Press, 2005.
[10] Y. Li and J. Shawe-Taylor, "Using KCCA for Japanese-English cross-language information retrieval and document classification," *Journal of Intelligent Information Systems*, vol. 27, no. 2, pp. 117–133, 2006.
[11] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
[12] A. Krizhevsky and G. E. Hinton, "Learning multiple layers of features from tiny images," Computer Science Department, University of Toronto, Tech. Rep., 2009.
[13] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.
[14] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Caltech, Tech. Rep., 2007.
[15] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Spinger-Verlag, 2002.
[16] J. Ngiam, P. W. Koh, Z. Chen, S. Bhaskar, and A. Y. Ng, "Sparse filtering," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 24. MIT Press, 2011.