Robust Multi-view Subspace Learning with Non-independently and Non-identically Distributed Complex Noise

Zongsheng Yue, Hongwei Yong, Deyu Meng, Member, IEEE, Qian Zhao, Yee Leung, and Lei Zhang, Fellow, IEEE

Abstract—Multi-view Subspace Learning (MSL), which aims at obtaining a low-dimensional latent subspace from multi-view data, has been widely used in practical applications. Most recent MSL approaches, however, only assume a simple i.i.d. Gaussian/Laplacian noise for all views of data, which largely under-estimates the noise complexity in practical multi-view data. Actually, in real cases, noises among different views generally have three specific characteristics. First, in each view the data noise always has a complex configuration beyond a simple Gaussian/Laplacian distribution. Second, the noise distributions of different views of data are generally non-identical and with evident distinctiveness. Third, noises among all views are non-independent but obviously correlated. Based on such understandings, we elaborately construct a new MSL model by more faithfully and comprehensively considering all these noise characteristics. Firstly, the noise in each view is modeled as a Dirichlet process Gaussian mixture model (DPGMM), which can fit a wider range of complex noise types than conventional Gaussian/Laplacian. Secondly, the DPGMM parameters in each view are different from one another, which encodes the “non-identical” noise property. Thirdly, the DPGMMs on all views share the same high-level priors by using the technique of hierarchical Dirichlet process, which encodes the “non-independent” noise property. All the aforementioned ideas are incorporated into an integrated graphic model which can be appropriately solved by the variational Bayes algorithm. The superiority of the proposed method is verified by experiments on 3D reconstruction simulations, multi-view face modeling and background subtraction, as compared with the current state-of-the-art MSL methods.

Index Terms—Multi-view, subspace learning, Dirichlet process mixture model, hierarchical Dirichlet process, variational Bayes.

I. INTRODUCTION

CONSIDERABLE data used in real applications, like face recognition [1], video surveillance [2], background subtraction [3] and 3D structure reconstruction [4]–[6], are collected from various domains or extracted with diverse features in order to capture possibly comprehensive underlying information of data. Such kinds of data are known as multi-view data. A typical example is shown in Fig. 1 (a), where a single scene is captured by multiple cameras located at different positions.

Since information obtained only from one view of data is relatively insufficient to describe the full shape of an object [7], it is always advantageous to leverage the information of multiple views to obtain a better description. Multi-view learning has thus been attracting much research attention recently in both academic and practical fields [8], and various methods have been proposed in recent years [9]. Multi-view subspace learning (MSL) [7], [10]–[12] represents one of the most typical categories of approaches along this research line.

The basic assumption of MSL is that each view is generated from one latent subspace, which is shared by all views. Most current MSL methods aim to embed the data into this low-dimensional latent subspace with a set of dictionaries such that each view of the data could be linearly combined through the coefficients and corresponding dictionary [7], [10], [11], [13], [14]. Through such learning mechanism, many of the current methods can achieve good performance on multi-view data collected from ideal scenarios.

However, one apparent drawback of these current methods is that they generally specify a simple $L_2$-norm or $L_1$-norm loss in the model, implying that the noise across all views of data is assumed to be an i.i.d. Gaussian or Laplacian distribution. This, however, always deviates from the real
noise configurations in practical cases. Specifically, noises in real multi-view data always possess the following three fold characteristics.

Firstly, the noise in each view is always too complex to be simply modeled as a Gaussian or Laplacian distribution. Like the surveillance multi-view videos shown in Fig. 1 (d), the noise (i.e., residual besides the stable background) in each view can be decomposed into multiple modalities (i.e., the foreground objects, the shadow of the objects, and some camera noises), which need to be finely depicted by more complex noise models beyond Gaussian/Laplacian [15]–[17].

Secondly, in practical cases, noises in different data views are non-identically distributed, i.e., they are always of evident distinctiveness due to different collecting angles, domains or sources. For example, as shown in Fig 1 (e), the residuals’ histograms of different views are obviously different, implying that noises of different views should have different distributions. The negligence to such inter-view noise difference tends to impact adversely on the performance of current methods.

Thirdly, for real multi-view data, the noise distributions are always non-independent and correlations among different views are evident [18]. For instance, for multiple surveillance videos captured by cameras with different angles of the same scene, the foreground objects incline to appear or leave simultaneously across all views of the videos, making the noise types among all views evidently correlated, as can be observed from the common noise shapes among views shown in Fig. 1 (c) and (e). By taking such noise correlations among all data views into consideration should more faithfully reflect the noise characteristics of multi-view data and enhance the robustness of an MSL method for practical applications under complicated noises.

To address the aforementioned noise fitting issues, in this paper we propose a new MSL method that fully takes the noise characteristics into consideration. The main contribution of this work can be summarized as follows:

- Instead of assuming simple Gaussian or Laplacian noise like conventional studies, we model the noise in each view of data as a Dirichlet process Gaussian mixture model (DPGMM), which is capable of fitting wider range of noise types [3], [19], [20] beyond the Gaussian/Laplacian distribution, thus finely encoding the “complex” noise property underlying data.

- Compared with the traditional i.i.d. noise assumption, we initiate a hierarchical model for modeling the non-i.i.d. noise in multi-view data. The DPGMM noise parameters in each view are different from others in order to encode the “non-identical” noise property. Furthermore, the DPGMM parameters among all views are shared from the similar high-level Dirichlet process (DP) parameters via the technique of Hierarchical Dirichlet Process (HDP), which encodes the “non-independent” noise property. Under this modeling strategy, the noise in practical multi-view data p can be more faithfully represented so as to enhance the robustness of an MSL method in dealing with real complex noises.

- The proposed model corresponds to a nonparametric Bayesian generative model, which is capable of automatically adapting the noise complexity based on data. A variational Bayes algorithm is readily designed to solve the model, and each of the involved parameters can be effectively updated in closed-form.

The paper is organized as follows. Section 2 introduces related works on MSL. Section 3 provides some preliminaries on the HDP technique. Section 4 presents the main model against the investigated problem, and the variational Bayes algorithm is given in Section 5. Experimental results on synthetic and real datasets are demonstrated in Section 6, and the paper is rounded up with a conclusion in Section 7.

II. RELATED WORK

During the past decades, many multi-view subspace learning approaches have been proposed. The Canonical Correlation Analysis (CCA) [21], [22], which exploits the shared latent subspace across diverse views, is a typical fundamental work along this line of research. Subsequently, a series of related approaches based on CCA have been proposed. For instance, Bach and Jordan [23] provided a probabilistic interpretation of CCA and perfected the theoretical basis of CCA. To handle nonlinear alignment, the kernel CCA [24] was proposed by projecting data onto a high-dimensional feature space. Besides, sparse formulation of CCA was also proposed in [25]. To enable the method to perform in complex noises, Nicolaou et al. [26] adopted $L_1$ loss to enhance the robustness of CCA, and Bach et al. [23] put forward a Student-t density model to handle more outliers.

Several other methods have also been presented to deal with the MSL task. Shared Gaussian Latent Variable Model (sGP LVM) [27], [28] learns the common latent structure of multi-view data by the Gaussian Process. Jia et al. [10] imposed structured sparsity to MSL through solving two convex subproblems alternately. Similarly, White et al. [11] proposed a more general unified framework under arbitrary convex loss function for multi-view latent subspace learning and then reformulated the problem to obtain a more precise solution of the problem. Other related methods on convex formulations for MSL can be found in [14], [29]–[31]. Cauchy loss [32] was firstly put forward as a more robust estimator for MSL in [7]. By simultaneously considering correlation and independence of multi-view data, some approaches divided the data into correlated components among all views and specific components with respect to each view. For example, JIVE [33] decomposed multi-view data into the sum of three terms: a low-rank approximation capturing the joint structure among different views, low-rank approximations capturing individual structures for each view and a residual noise. Motivated by JIVE, Zhou et al. [34] used a common orthogonal basis extraction (COBE) algorithm to identify and separate the shared and individual features.

However, most traditional MSL methods generally assume a simple form for the noise in multi-view data, like an i.i.d. Gaussian or Laplacian distribution, which always largely deviates from the noise configuration in practical multi-view
data, and thus tends to have an unstable performance in the presence of non-i.i.d. complex noises. Thus, our aim is to
more faithfully capture and represent real noise structures to alleviate this robustness issue.

III. PRELIMINARIES

In this section, we review some preliminary knowledge about Dirichlet Process and Hierarchical Dirichlet Process to set the stage for the subsequent presentation of our model.

A. Dirichlet Process

The Dirichlet Process (DP), introduced by Ferguson [35], is a distribution over distributions. The DP is parameterized by a base distribution and a concentration parameter. Specifically, let \( H \) be a probability distribution over any measurable set \( \Theta \) and \( \gamma \) be a positive real number. Then we say \( G \) is a Dirichlet Process distributed with the base distribution \( H \) and concentration parameter \( \gamma \), written as \( G \sim \text{DP}(\gamma, H) \), if

\[
(G(A_1), \ldots, G(A_r)) \sim \text{Dir}(\alpha H(A_1), \ldots, \alpha H(A_r)) \tag{1}
\]

for every finite measurable partitions \( A_1, A_2, \ldots, A_r \) of \( \Theta \).

There is another more explicit characterization of DP in terms of a stick-breaking construction due to Sethuraman [36]. Considering two infinite collections of independent random variables, \( \pi_i \sim \text{Beta}(1, \gamma) \) and \( \eta_i \sim H \), for \( i = 1, 2, \ldots \), then the stick-breaking representation of \( G \) is as follows:

\[
\pi_i = \pi_i \prod_{j=1}^{i-1} (1 - \pi_j'), \quad G = \sum_{i=1}^{\infty} \pi_i \delta_{\eta_i^*} \tag{2}
\]

where \( \delta_{\eta_i^*} \) represents the Dirac Delta function concentrated at \( \eta_i^* \). Sethuraman [36] showed that \( G \) as defined in this way is a random probability measure according to \( \text{DP}(\gamma, H) \).

The Dirichlet process Gaussian mixture clustering model is a typical example of the DP models. It is an advanced version of conventional Gaussian mixture model (GMM), with capability of adaptively adjusting its Gaussian component number based on data. The main idea of the model is as follows: Setting \( H \) as a Gaussian-Inverse-Wishart distribution (conjugate priors of \( (\mu, \Sigma) \)), thus \( G \) is a distribution over infinite number of clusters, each corresponding to a pair value \( (\mu_i, \Sigma_i) \). A draw from \( G \) \( (G \sim \text{DP}(\gamma, H)) \) will choose one cluster \( k \) and return one pair value \( (\mu_k, \Sigma_k) \), which determines one specific Gaussian component. Thus, for every data sample needed to be clustered, we can sample from the posterior of \( G \) until convergence is reached. We can also understand DP in the sense of the stick-breaking construction. Instead of sampling the Gaussian component (i.e., cluster) directly for each data sample, it completes the clustering process through sampling the component weight \( \pi_k \) of the Gaussian mixture for each cluster. In this clustering process, the value of the concentration parameter \( \gamma \) controls the final number of clusters. Through this stochastic dynamical mechanism, DP offers full flexibility in selecting the number of clusters (i.e., the number of Gaussian components).

B. Hierarchical Dirichlet Process

The Hierarchical Dirichlet Process (HDP) [37] is originally proposed as a hierarchical Bayesian model for natural language processing, which provides a flexible framework for sharing mixture components among groups of related data. A two-level HDP is a collection of DPs that share a base distribution \( H \), meanwhile these DPs are also drawn from a higher level DP, i.e.,

\[
G \sim \text{DP}(\gamma, H), \quad G_j \sim \text{DP}(\alpha_j, G), \tag{3}
\]

where \( j \) is an index for groups of data. It should be noted that all distributions \( G_j \) possess different parameters while share the base distribution \( G \), which can thus be regraded as a desirable noise prior readily representing our expected “non-independently and non-identically distributed” noise distribution. We can extend the DPGMM example in Section III-A to a clustering problem with three groups of related data, and then the clusters in each group share the parameters from a similar high-level clustering process, which leads to a hierarchical clustering process as shown in Fig. 2.

IV. MSL MODEL WITH NON-I.I.D. COMPLEX NOISE

A. Notations

For convenience of formulating our model, we firstly introduce some necessary notations in the following. We use light lowercase letters, bold lowercase letters and bold uppercase letters to denote scalars, vectors and matrices, respectively. Given a matrix \( X \), we use the term \( x_{ij} \) to denote its \( j \)th column vector, \( x_i \) its \( i \)th row vector, and \( x_{ij} \) the element in its \( i \)th row and \( j \)th column, respectively. For probability distributions, \( \mathcal{N}(\mu, \Sigma) \) denotes the multivariate Gaussian distribution with mean \( \mu \) and covariance matrix \( \Sigma \), \( \mathcal{N}(\mu, \xi^{-1}) \) the univariate Gaussian distribution with mean \( \mu \) and precision \( \xi \), Beta\((a, b)\) the Beta distribution with parameters \( a \) and \( b \),
Gam(c, d) the Gamma distribution with shape parameter c and scale parameter d, and Multi(π) the multinomial distribution with parameters π.

B. Model Formulation

Now we give the formulation of our proposed model. Let the observed multi-view data be $\mathbf{X} = \{\mathbf{X}^v\}_{v=1}^V$, where $\mathbf{X}^v \in \mathbb{R}^{d \times n}$ represents the vth view of data, and n, d are the number of data samples and the dimensionality in each view, respectively. By considering a generative model, we can decompose the observed data into:

$$\mathbf{X}^v = \mathbf{S}^v + \mathbf{E}^v,$$

(4)

where $\mathbf{E}^v = \{e_{ij}^v\}_{d \times n}$ denotes the residual term (i.e., noise component) and $\mathbf{S}^v \in \mathbb{R}^{d \times n}$ is the expected data located on the latent subspace.

1) Non-i.i.d. noise modeling: Instead of using a simple i.i.d. unimodal distribution (Gaussian or Laplacian) to model the residual noise in all views of data, we model the noise of each as a DPGMM, which is an advanced version of non-parametric GMM model capable of adaptively rectifying the number of Gaussian components based on data [38], [39], and then associate these DPGMMs by considering the intrinsic correlations among all views. This can be achieved by a two-level HDP as described in Section III-B.

Specifically, we consider the noise of each view $\mathbf{E}^v$. Following the idea of [19], [20], we can model it as an element-wise Gaussian mixture distribution:

$$e_{ij}^v \sim \sum_{k=1}^{K} \pi_{ij}^v \mathcal{N}(0, \xi_k^{-1}),$$

(5)

which can be equivalently reformulated as a two level generative process with a latent variable $b_{ij}^v \in \mathbb{R}^{d \times n}$:

$$e_{ij}^v \sim \mathcal{N}(0, \xi_{b_{ij}^v}^{-1}), \quad b_{ij}^v \sim \text{Multi}(\mathbf{p}^v),$$

(6)

where $\mathbf{p}^v \in \mathbb{R}^K$, $b_{ij}^v \in \{1, 2, \ldots, K\}$. In order to encode the correlation and distinctiveness of noise in different views, we adopt a hierarchical dirichlet distribution prior, i.e.:

$$\mathbf{p}^v \sim \text{Dir}(\alpha^v \mathbf{B}), \quad \mathbf{B} \sim \text{Dir}(\gamma/K, \gamma/K, \ldots, \gamma/K).$$

(7)

To make the model flexible, we further assume $K \rightarrow +\infty$, Teh et al. [37] showed that the limit of Eq. (5)-(7) is the following HDP noise model:

$$G \sim \text{DP}(\gamma, H), \quad G^v \sim \text{DP}(\alpha^v, G), \quad e_{ij}^v \sim \mathcal{N}(0, \psi_{ij}^v^{-1}),$$

(8)

where $H = \text{Gam}(e_0, f_0)$.

Alternatively, we can intuitively explain our noise model (Eq. (8)) from another hierarchical bayesian model equivalently based on the stick-breaking construction. Instead of placing a prior on $\mathbf{B}$, each view can choose a subset of $T$ mixture components from a high-level set of $K$ mixture components as shown in Fig. 2:

(1) Draw $K$ (is sufficiently large) samples from the Gamma distribution, i.e.

$$\xi_k \sim \text{Gam}(e_0, f_0),$$

and form $K$ Gaussian distributions $\mathcal{N}(0, \xi_k^{-1})$ as listed in the red dotted rectangle in Fig. 2.

(2) $T$ ($T \leq K$) values are selected from $\{\xi_k\}_{k=1}^K$ for each view, whose index $e_t^v \in \mathbb{R}^T$ is constructed by the following stick-breaking representation:

$$e_t^v \sim \text{Multi}(\mathbf{\beta}), \quad \beta_k = \beta_0^t \prod_{l=1}^{k-1} (1 - \beta_l^t), \quad \beta_0^t \sim \text{Beta}(1, \gamma),$$

where $e_t^v \in \{1, 2, \ldots, T\}, \mathbf{\pi}^v \in \mathbb{R}^T$.

(3) Generate indicated variable $z_{ij}^v$ from another stick-breaking process:

$$z_{ij}^v \sim \text{Multi}(\mathbf{\pi}^v), \quad \pi_0^v = \pi_0^v \prod_{l=1}^{k-1} (1 - \pi_l^v), \quad \pi_0^v \sim \text{Beta}(1, \alpha^v),$$

where $z_{ij}^v \in \{1, 2, \ldots, T\}$.

(4) Generate noise for each view of data, i.e.:

$$e_{ij}^v \sim \mathcal{N}(0, \xi_{z_{ij}^v}^{-1}).$$

Combining the above steps (1)-(4), we can get the following hierarchical noise model represented by two coupled stick-breaking process:

$$\xi_k \sim \text{Gam}(e_0, f_0), \quad e_{ij}^v \sim \mathcal{N}(0, \xi_{z_{ij}^v}^{-1}),$$

$$c_t^v \sim \text{Multi}(\mathbf{\beta}), \quad z_{ij}^v \sim \text{Multi}(\mathbf{\pi}^v),$$

$$\beta_k = \beta_0^t \prod_{l=1}^{k-1} (1 - \beta_l^t), \quad \pi_0^v = \pi_0^v \prod_{l=1}^{k-1} (1 - \pi_l^v), \quad \beta_0^t \sim \text{Beta}(1, \gamma), \quad \pi_0^v \sim \text{Beta}(1, \alpha^v).$$

(9)

It should be noted that the $\xi$ in Eq. (9) and the $\psi$ in Eq. (8) have the following relation:

$$\xi_{z_{ij}^v} = \psi_{ij}^v.$$
2) **Latent subspace modeling:** As conventional MSL methods [7], [10], we assume that each view of data is embedded into a latent subspace $\mathbf{R}$ with a dictionary $\mathbf{L}^v$. Based on such an assumption, we formulate $\mathbf{S}^v \in \mathbb{R}^{d \times n}$ as the product of two smaller matrices $\mathbf{L}^v \in \mathbb{R}^{d \times l}$ and $\mathbf{R} \in \mathbb{R}^l \times n$, i.e.,

$$\mathbf{S}^v = \mathbf{L}^v \mathbf{R} = \sum_{r=1}^{l} \mathbf{L}^v_r \mathbf{R}_r.$$ (11)

In order to obtain a full Bayesian model, we impose the following priors on $\mathbf{R}$ [20]:

$$\mathbf{R}_r \sim \mathcal{N}(\mathbf{0}, \frac{1}{\tau_r} \mathbf{I}_n).$$ (12)

Besides, we also impose column priors on $\mathbf{L}^v$ [10], [41], i.e.,

$$\mathbf{L}^v_r \sim \mathcal{N}(\mathbf{0}, \frac{1}{\lambda_r} \mathbf{I}_d).$$ (13)

The intuition behind this formulation is that we expected each $\mathbf{S}^v$ to only depend on a subset of latent dimensions. And the conjugate priors imposed on the precision variables $\lambda_r$ and $\tau_r$ are:

$$\lambda_r \sim \text{Gam}(a_0, b_0), \quad \tau_r \sim \text{Gam}(c_0, d_0),$$ (14)

respectively, where $a_0$, $b_0$, $c_0$ and $d_0$ are all hyperparameters.

3) **Full Bayesian model of NIID-MSL:** Combining Eqs. (4), (9), (10)-(14) together, we can construct a full bayesian model of MSL with non-i.i.d. noise. We thus call our model NIID-MSL in brief. The corresponding graphical model of our model is shown in Fig. 3. The goal turns to infer the posteriors of all involved variables:

$$p(\mathbf{L}, \mathbf{R}, \mathbf{\xi}, \mathbf{C}, \mathbf{Z}, \beta, \pi, \alpha, \lambda, \tau, \gamma | \mathbf{X}),$$ (15)

where $\mathbf{C} = \{c_r^v\}$, $\mathbf{Z} = \{z_{ij}^v\}$.

**C. Discussion**

Our proposed NIID-MSL model considers the correlation and distinctiveness of the noises among different views, which is inspired by the general non-identical and non-independent noise characteristics of practical multi-view data. It should be noted that the proposed method can be easily transformed into two of its degenerated versions, including the (1) i.i.d. noise version, by using a single DPGMM to fit the noise of all views, and (2) the non-identical but independent noise version by assuming a different DPGMM for each view. Due to the page limitation, we put the models of both such degenerated versions in the supplementary material. Besides, we also list experimental results in this supplementary file to show the necessity of considering such non-i.i.d. assumptions in our noise model.

**V. Variational Inference**

Directly computing the posterior distribution under a HDP mixture prior is intractable, thus approximate inference methods are required to be designed. Although Markov chain Monte Carlo (MCMC) sampling method can provide very accurate approximation to the posterior [42], this method is limited for massive-scale data by its high computational cost and the difficulty to detect convergence. As a more efficient and deterministic alternative to MCMC, we adopt a Variational Bayesian (VB) method for NIID-MSL in this paper [43].

VB seeks an approximation distribution $q(\mathbf{x})$ to the true posteriors $p(\mathbf{x} | \mathbf{D})$ ($\mathbf{D}$ is the observed data) by solving the following variational optimization problem:

$$\min_{q \in \mathcal{C}} KL(q || p) = - \int q(\mathbf{x}) \ln \left( \frac{p(\mathbf{x} | \mathbf{D})}{q(\mathbf{x})} \right) d\mathbf{x},$$ (16)

where $KL(q || p)$ denotes the KL divergence between $q(\mathbf{x})$ and $p(\mathbf{x} | \mathbf{D})$, and $\mathcal{C}$ denotes the set of probability densities with certain restrictions to make the minimization tractable. Assuming $q(\mathbf{x}) = \prod_i q_j(x_j)$, the closed-form solution to $q_j(x_j)$ with other factors fixed, can be attained as:

$$q_j(x_j) = \frac{\exp \left( E_{x \setminus x_j} \ln p(x | \mathbf{D}) \right)}{\int \exp \left( E_{x \setminus x_j} \ln p(x | \mathbf{D}) \right) dx},$$ (17)

where $E_{x} [\cdot]$ denotes the expectation over the variable $x$, and $x \setminus x_j$ represents the set of $x$ with $x_j$ removed. Eq. (16) can be solved by alternatively calculating Eq. (17) with respect to all of its involved variables.

**A. Estimation of the Posterior**

Based on the stick-breaking construction and the truncation strategy [44], we can approximate the posterior distribution Eq. (15) with the following factorized form:

$$q(\mathbf{L}, \mathbf{R}, \mathbf{\xi}, \mathbf{C}, \mathbf{Z}, \beta, \pi, \alpha, \lambda, \tau, \gamma) =$$

$$\prod_{v=1}^{V} \prod_{l} q(L_l^v)q(\lambda_r^v) \prod_{r=1}^{l} q(R_r)q(\tau_r) \prod_{k=1}^{K} q(\xi_k)q(\beta_k^v)$$

$$\prod_{d,n}^{V} \prod_{i,j}^{V} q(z_{ij}^v) \prod_{v=1}^{V} \prod_{e=1}^{T} q(c_{e}^v)q(\pi_e^v) \prod_{v=1}^{V} q(\alpha^v)q(\gamma).$$ (18)

Then we can analytically infer all the factorized distributions involved in Eq. (18) as below. The computational details are provided in the supplementary material.

---

Update noise component: The parameters involved in the noise components are $\xi$, $C$, $Z$, $\pi$, $\beta$. We use the stick-breaking procedure by setting a large enough value of $K$ and $T$ for truncated approximation, and then get the following updating equations on $C$ and $Z$:

$$q(e_{ij}^x) = \text{Multi}(e_{ij}^x | \rho_{ij}^x), \quad q(e_{ij}^z) = \text{Multi}(e_{ij}^z | \varphi_{ik}^z),$$

where

$$\rho_{ij}^x \propto \exp \left( \sum_{k} \varphi_{tk}^z \left[ \ln N(e_{ij}^z | 0, \xi_k^{-1}) \right] + E[\ln \pi_k^z] \right),$$

$$\varphi_{tk}^y \propto \exp \left( \sum_{i,j} \rho_{ij}^x \left[ \ln N(e_{ij}^y | 0, \xi_k^{-1}) \right] + E[\ln \beta_k^y] \right),$$

$$E[\ln \pi_k^z] = E[\ln \pi_k^x] + \sum_{s=1}^{t-1} E[\ln (1 - \pi_k^x)],$$

$$E[\ln \beta_k^y] = E[\ln \beta_k^x] + \sum_{t=1}^{k-1} E[\ln (1 - \beta_k^x)].$$

As for $\xi$, based on its conjugate priors, we have:

$$q(\xi_k) = \text{Gam}(\xi_k | c_k, f_k),$$

where

$$e_k = \frac{1}{2} \sum_{v,i,j,t} \rho_{ij}^z \varphi_{tk}^z + e_0,$$

$$f_k = \frac{1}{2} \sum_{v,i,j,t} \rho_{ij}^z \varphi_{tk}^z \left( x_{ij}^z - L_{i}^{v} R_{j}^{v} \right)^2 + f_0.$$

Similarly, for $\pi$, $\beta$, we have

$$q(\pi_k^y) = \text{Beta}(\pi_k^y | r_k^v, w_k^v), \quad q(\beta_k^z) = \text{Beta}(\beta_k^z | s_k^1, s_k^2),$$

where

$$r_k^v = \sum_{i,j} \rho_{ij}^z + 1, \quad w_k^v = \sum_{i,j,s=t+1} \rho_{ij}^z + E[\alpha^v],$$

$$s_k^1 = \sum_{v,t} \varphi_{tk}^y + 1, \quad s_k^2 = \sum_{v,t,s=k+1} \varphi_{ts}^y + E[\gamma].$$

Update HDP concentration parameters: According to the conjugate priors of $\alpha$ and $\gamma$ in Eq. (10), we can get the following updating equations:

$$q(\alpha^v) = \text{Gam}(\alpha^v | m^v, n^v), \quad q(\gamma) = \text{Gam}(\gamma | g, h),$$

where

$$m^v = T + m_0, \quad n^v = n_0 - \sum_t E[\ln (1 - \pi_k^x)],$$

$$g = K + g_0, \quad h = h_0 - \sum_k E[\ln (1 - \beta_k^y)].$$

Update the subspace components: The parameters involved in the latent subspace component are $L^v$, $R$, $\lambda$, $\tau$. For each row of $L^v$, using the factorization in Eq. (11), we can get

$$q(L_i^v) = \mathcal{N}(L_i^v | \mu_i^v, \Sigma_i^v)$$

with mean $\mu_i^v$ and covariance matrix $\Sigma_i^v$, given by

$$\Sigma_i^v = \left( \sum_{j,t,k} \rho_{ij}^v \varphi_{tk}^v E[\xi_k E[R_{j} R_{j}^T] + A_k^j] \right)^{-1},$$

$$\mu_i^v = \Sigma_i^v \left( \sum_{j,t,k} \rho_{ij}^v \varphi_{tk}^v E[\xi_k x_{ij}^v E[R_{j} R_{j}^T] \right),$$

where $A_k^j = \text{diag}(E[\lambda^v])$. Similarly, for each column of $R$, we have

$$q(R_{ij}) = \mathcal{N}(R_{ij} | \mu_j, \Sigma_j),$$

and the mean $\mu_j$ and the covariance $\Sigma_j$ are calculated by

$$\Sigma_j = \left( \sum_{v,i,t,k} \rho_{ij}^v \varphi_{tk}^v E[\xi_k E[L_i^v L_i^v]^T + A_R] \right)^{-1},$$

$$\mu_j = \Sigma_j \left( \sum_{v,i,t,k} \rho_{ij}^v \varphi_{tk}^v E[\xi_k x_{ij}^v E[L_i^v] \right),$$

where $A_R = \text{diag}(E[\tau])$.

For parameters $\lambda$ and $\tau$, we have

$$q(\lambda^v) = \text{Gam}(\lambda^v | a_v^\tau, b_v^\tau), \quad q(\tau) = \text{Gam}(\tau | c_v, d_v),$$

with

$$a_v^\tau = \frac{d}{2} + a_0, \quad b_v^\tau = \frac{1}{2} E[L_i^v L_i^v] + b_0,$$

$$c_v = \frac{n}{2} + c_0, \quad d_v = \frac{1}{2} E[R_{ij} R_{ij}] + d_0,$$

where $d$ and $n$ are the dimensionality and the number of the observed data $X^v$ in the $v$th view, respectively.

B. Settings of Hyperparameters

We set all the hyperparameters involved in our model in a non-informative manner to make as small an impact as possible on the inference of the posterior distribution [43]. Specifically, throughout our experiments, we easily set $a_0$, $b_0$, $c_0$, $d_0$, $f_0$, $g_0$, $h_0$, $m_0$, $n_0$ for a small value $10^{-6}$. Our method performs stably well in all experiments with these easy settings.

C. VB Algorithm Acceleration

As with the truncated Dirichlet Process [44], there is a trade-off issue in the designed VB algorithm for the NIID-MSL: the larger $K$ and $T$ are, the more accurate the model becomes, but the slower the model is in inferring the parameters. In order to accelerate the inference speed, we employed two strategies:

- Cut the components in the first-level GMM: The coefficient $\beta_k$ in Eq. (9) reflects the importance of the $k$th component, and discard the $k$th component if $\beta_k < 1e^{-4}$.
- Merge the mixture components of each view in the second-level GMMs: The value $\varphi_{tk}^v$ in Eq. (19) represents the exception that the $t$th Gaussian component in the $v$th view is shared from the $k$th component in the first-level GMM. In each iteration, we merge the $t_1$-th component
and the $t_2$-th component into one for each view if the following relation is satisfied:

$$\| \varphi_{t_1}^v - \varphi_{t_2}^v \|_{\text{max}} < 0.01,$$

where $\| \cdot \|_{\text{max}}$ means the Max norm of a vector.

VI. EXPERIMENTAL RESULTS

We evaluate the performance of our proposed NIID-MSL model qualitatively and quantitatively on both synthetic and real datasets. Five state-of-the-art multi-view subspace learning methods, including FLSSS [10], MSL [11], CSRL [14], JIVE [33] and MISL [7], are considered for comparison in the synthetic simulation, face image recovery and background subtraction experiments. All the involved parameters in these comparison methods are finely tuned by default settings or following the rules in their papers to guarantee their possibly optimal performance. The source code of the proposed NIID-MSL is available at https://github.com/zsyOAOA/NIID-MSL.

A. Synthetic simulations

We evaluate our proposed method on the problem of 3D structure reconstruction based on a synthetic “S-curve” dataset as shown in Fig. 4 (a), which is generated in the similar manner as [45]. We uniformly sampled 4000 3-dimensional data points from the “S-curve” manifold (Fig. 4 (b)), and then projected them onto three 2-dimensional planes (i.e., X-Y plane, X-Z plane, Y-Z plane) as three views of the data, as depicted in Fig. 4 (c)-(e). Obviously, each of these three views of data correspond to a 2-dimensional subspace in the 3-dimensional space. Our goal is to reconstruct its 3D “S-curve” shape by utilizing these three views of the 2D projected data. When the data are clean and directly generated from the “S-curve” manifold (i.e., Case 1 of Table I), all competing MSL methods can effectively reconstruct the manifold shapes, as displayed in the first row of Fig. 5. To evaluate the robustness of all competing methods, we further added five types of i.i.d. and non-i.i.d. noises with different signal to noise ratios (SNR) to the data to increase its complexity, in which the SNR of the $v$th view is calculated with the following equation:

$$\text{SNR}^v = 10 \log_{10} \left( \frac{1}{d \times n} \sum_{i,j} (x_{ij}^v - \bar{x}^v)^2 / \sigma_{\text{noise}}^v \right),$$

where $\bar{x}^v = \frac{1}{d \times n} \sum_{i,j} x_{ij}^v$. Details of the added noises are listed in Table I. It should be noted that Case 2 corresponds to the i.i.d. GMM noise while the others are non-i.i.d. ones.

Two criteria are utilized for performance assessment. (1) Relative Reconstruction Square Error (RRSE): $\| \hat{L} - L \|_F / \| L \|_F$, and (2) Relative Reconstruction Absolute Error (RRAE) $\| \hat{L} - L \|_1 / \| L \|_1$, where $\hat{L}$ and $\hat{L}$ denote the groundtruth and reconstructed data, respectively. The performance of all competing methods is listed in Table II. We can see from the table that, in the case without noise (Case 1), all competing methods can perform well with small reconstruction error. For all of the other noisy cases, however, our proposed method outperforms the other comparison methods in most cases. Specifically, for the i.i.d. noise case 2, our method performs substantially better than the other competing methods. As the noise becomes more complicated and with evidently more non-i.i.d. configurations, our proposed method performs evidently much better than all others. This superiority

![Fig. 4](image-url)
substantiates the robustness of our method against complex noise in the multi-view subspace learning task. For easy visualization, Fig. 5 displays the reconstructed 3-dimensional points by all the competing methods in all cases. It is easy to see that the performances of most competing methods degenerate under complex noise cases, especially in non-i.i.d. cases, while our proposed method can stably attain a more superior reconstruction performance. This complies with the quantitative comparison and further substantiates the robustness of our proposed method.

**B. Face image recovery experiments**

In this section, we test the effectiveness and the robustness of the proposed NIID-MSL method on multi-view face images by using the CMU Multi-PIE face dataset [46], which includes 337 subjects with multiple poses and expressions. For our purpose, 100 subjects were randomly selected to avoid the “out of memory” issue that tends to occur in the utilized MSL methods, and each subject contains one face image captured at 5 different angles ($-30^\circ$, $-15^\circ$, $0^\circ$, $15^\circ$, $30^\circ$) with size $128 \times 96$. A typical subject is shown in the first row of Fig. 6 (a). The original face images (No noise case) were directly fed into the algorithm to test the effectiveness of our proposed
method. Furthermore, we added three types of noises to the original images to test the robustness of different methods as follows:

**Gaussian Noise**: Zero-mean Gaussian noise was added to each view of images. The noise variances were all set for 0.02 for all views.

**Block Noise**: Two blocks were randomly selected in every face image for all 5 views to add “salt & pepper” noise. The density of “salt & pepper” was uniformly sampled from the $[0, 0.5]$ region.

**Mixing Noise**: Two types of noises randomly selected from the Gaussian noise, sparse noise (randomly select 10% of the image pixels and assign random values within the region $[-2, 2]$) and block noise, as mentioned above, were added to each view.

For fair comparison, the rank in all experiments is set for 15 for all of the competing methods. The performances are also evaluated in terms of RRSE and RRAE. The results are listed in Table III, and the images recovered by different methods are shown in Fig. 6. From Table III and Fig. 6, we have the following observations:

1) The NIID-MSL method attains the best (6 out of 8 cases) or second best (2 out of 8 cases) performance in both quantitative metrics and visualization. Such superiority is especially evident in the more complicated mixture noise case.

2) Most of the state-of-the-art multi-view subspace learning methods can remove the effect of illumination and reconstruct from the original images without noise, which validates the effectiveness of these methods.

3) Under the Gaussian noise case, the $L_2$ loss methods (JIVE, CSRL, FLSSS) outperform the $L_1$ loss method (MSL). In contrast, as for block noise, $L_1$ loss method establishes a clear superiority. The MISL method based on Cauchy loss also have a good performance, since Cauchy loss has a breakdown point of nearly 50 percent [32] and thus is more robust. Our proposed NIID-MSL evidently surpasses other methods, which proves its robustness against complex noise.

This ‘Mixing noise’ case simply indicates that we add more than one kind of noise to the original face images, which does not follow the mixture distribution in statistics.

---

**TABLE III**

Quantitative comparison of all competing methods on Multi-PIE face data. In each series of experiments, the best and the second best results are highlighted in bold and italic, respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FLSSS RSE</th>
<th>MSL RSE</th>
<th>JIVE RSE</th>
<th>CSRL RSE</th>
<th>MISL RSE</th>
<th>NIID-MSL RSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>No noise</td>
<td>0.031</td>
<td>0.015</td>
<td>0.0074</td>
<td>0.010</td>
<td>0.0092</td>
<td>0.0092</td>
</tr>
<tr>
<td>RRSE</td>
<td>0.13</td>
<td>0.090</td>
<td>0.068</td>
<td>0.077</td>
<td>0.070</td>
<td>0.067</td>
</tr>
<tr>
<td>Gaussian noise</td>
<td>0.018</td>
<td>0.022</td>
<td>0.019</td>
<td>0.018</td>
<td>0.017</td>
<td>0.014</td>
</tr>
<tr>
<td>RRSE</td>
<td>0.17</td>
<td>0.12</td>
<td>0.11</td>
<td>0.10</td>
<td>0.10</td>
<td>0.09</td>
</tr>
<tr>
<td>Block noise</td>
<td>0.019</td>
<td>0.011</td>
<td>0.022</td>
<td>0.018</td>
<td>0.019</td>
<td>0.011</td>
</tr>
<tr>
<td>RRSE</td>
<td>0.10</td>
<td>0.068</td>
<td>0.11</td>
<td>0.10</td>
<td>0.099</td>
<td>0.070</td>
</tr>
<tr>
<td>Mixture noise</td>
<td>0.14</td>
<td>0.020</td>
<td>0.17</td>
<td>0.11</td>
<td>0.11</td>
<td>0.015</td>
</tr>
<tr>
<td>RRSE</td>
<td>0.23</td>
<td>0.10</td>
<td>0.26</td>
<td>0.21</td>
<td>0.16</td>
<td>0.086</td>
</tr>
</tbody>
</table>

---

Fig. 6. Typical reconstructed face images of CMU Multi-PIE face dataset. From top to bottom: noise faces (or original face in (a)), reconstructed faces by different methods. From left to right: (a) typical reconstructed faces from original face images, (b)-(d) typical reconstructed faces under Gaussian noise, block noise and mixture noise.

Fig. 7. Visual comparison of the groundtruth (denoted by True) noise probability density functions and those estimated (denoted by Est-NIID) by the NIID-MSL method in the face image recovery experiments. The embedded sub-figures depict the zoom-in of the indicated portion.
The promising performance of the NIID-MSL method as shown in these face experiments can be easily explained by Fig. 7, which compares the groundtruth noise distributions and the estimated ones\(^4\). It can be easily observed that the estimated noise distributions can match the true ones to a very accurate extent for Gaussian and mixture noise. As for block occlusion, reasonably, a peak fat tail distribution, which is similar to the Laplacian distribution, is estimated by our NIID-MSL method, since the noise is only embedded in some local parts of the data. Such good noise-fitting capability of our proposed method then naturally leads to its good performance on its recovery of the groundtruth faces from their noisy versions.

C. Foreground Detection on RGB data

Surveillance videos with RGB color frames are very commonly collected in real life due to the wide spread of digital cameras. Intuitively, the information contained in the R, G and B channels of such kinds of data are highly correlated while also with differences. Here, we attempt to use MSL methods to detect the foreground on the RGB data regarding the three channels of a video as three views of data. We employ the Wallflower dataset\(^5\) [47] and I2R dataset\(^6\) to evaluate the proposed method. The former includes 6 video sequences and contains 8 video sequences. And each video contains hundreds or thousands of frames in total, of which one or several frames are pre-annotated as the groundtruth of the corresponding foreground object masks. Due to memory limitation of our computer, we only crop about 300 frames containing the ones with pre-annotated masks in each video sequence to test the effectiveness and robustness of different methods. The utilized video sequences are released at https://github.com/zsyOAOA/NIID-MSL for easy reproduction of our experiments.

\(^4\)The true noise distribution curve is depicted using kernel density estimator, and the estimated distributions obtained by the NIID-MSL method are also plotted using kernel density estimator, but according to the estimated residuals.


\(^6\)http://vis-www.cs.umass.edu/~narayana/castanza/I2Rdataset/

The residual in such kinds of data is generally non-i.i.d.. Specifically, each channel generally has its own different distribution since the sensitive spectral band of R, G and B sensors is distinct, as shown in Fig. 8. However, all sensors observed the same object, so the residual distributions are also of certain correlation. The noise in RGB surveillance videos are thus generally more complicated than those assumed by the current MSL methods, which in better accordance with our non-i.i.d. structure.

In this experiment we utilize the F-measure as the quantitative metric for performance evaluation, which is calculated as follows:

\[
\text{F-measure} = 2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

where \(\text{precision} = \frac{|S_f \cap S_{gt}|}{|S_f|}\) and \(\text{recall} = \frac{|S_f \cap S_{gt}|}{|S_{gt}|}\). \(S_f\) and \(S_{gt}\) denoted the support sets of the foreground calculated from the competing methods and the groundtruth one, respectively. For all competing methods, we choose many different threshold values to make the F-measure as large as possible. The larger the F-measure, the more accurate the foreground object is detected by the corresponding method.

The videos employed in this experiment contain different scenes, which involve not only simple static (such as Bootstrapping, Aperture, Escalator, Lobby) but also complex dynamic (such as Camouflage, SwitchLight, WavingTrees and WaterSurface) backgrounds. For videos with static backgrounds, we simply set the rank as 1 for all competing methods. However, for videos with dynamic backgrounds, we set the rank ranging from 1 to 5 and then determined the optimal rank parameter according to the F-Measures of all competing methods. It should be noted that in most cases all of the utilized methods attain best performance under the same rank setting, and their performances are thus fairly tested under

<table>
<thead>
<tr>
<th>Video</th>
<th>PLSSS MSL JIVE CSRL MSL NIID-MSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campus</td>
<td>0.64 0.73 0.54 0.63 0.63 0.72</td>
</tr>
<tr>
<td>Escalator</td>
<td>0.66 0.68 0.68 0.69 0.70 0.75</td>
</tr>
<tr>
<td>Fountain</td>
<td>0.57 0.82 0.48 0.58 0.56 0.83</td>
</tr>
<tr>
<td>Curtain</td>
<td>0.75 0.88 0.46 0.82 0.83 0.90</td>
</tr>
<tr>
<td>Hall</td>
<td>0.60 0.71 0.57 0.60 0.62 0.75</td>
</tr>
<tr>
<td>Lobby</td>
<td>0.93 0.91 0.88 0.91 0.91 0.92</td>
</tr>
<tr>
<td>ShoppingMall</td>
<td>0.85 0.68 0.81 0.84 0.84 0.84</td>
</tr>
<tr>
<td>WaterSurface</td>
<td>0.81 0.82 0.80 0.82 0.82 0.82</td>
</tr>
<tr>
<td>Mean</td>
<td>0.72 0.78 0.65 0.74 0.74 0.82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Video</th>
<th>PLSSS MSL JIVE CSRL MSL NIID-MSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrapping</td>
<td>0.70 0.72 0.73 0.73 0.73 0.73</td>
</tr>
<tr>
<td>Camouflage</td>
<td>0.71 0.71 0.73 0.71 0.71 0.74</td>
</tr>
<tr>
<td>Aperture</td>
<td>0.85 0.96 0.57 0.86 0.86 0.96</td>
</tr>
<tr>
<td>SwitchLight</td>
<td>0.53 0.67 0.48 0.59 0.60 0.68</td>
</tr>
<tr>
<td>TimeOfDay</td>
<td>0.46 0.76 0.51 0.66 0.65 0.76</td>
</tr>
<tr>
<td>WavingTrees</td>
<td>0.64 0.61 0.62 0.61 0.63 0.91</td>
</tr>
<tr>
<td>Mean</td>
<td>0.65 0.74 0.61 0.69 0.70 0.80</td>
</tr>
</tbody>
</table>
the same optimal rank parameter. More specifically, we set the rank parameter of all competing methods as 3 for videos of Camouflage, SwitchLight, TimeOfDay in Table IV and videos of Campus, Fountain, Hall in Table V, and set the rank as 5 for the other videos. Note that we regard the absolute value of the residual term $|X^\alpha - L^\alpha R|$ as the foreground to calculate the F-Measure.

Table IV and Table V list the F-measure value calculated from all the competing methods of the two groups of datasets, respectively. It is easy to see that out proposed NIID-MSL achieves the best performance on average. More concretely, NIID-MSL obtains comparable or better result for the videos with simple static background, such as Bootstrapping, Apertu, Escalator, ShoppingMall, etc. This is because the latent subspace prior, which is strong enough to extract the background, equipped with a relatively robust loss (such as the $L_1$ loss in MSL, Cauchy loss in MISL) naturally leads to a good performance. However, for videos with complex dynamic background, such as waving tree in the WavingTrees and flickering computer screen in the Camouflage, the superiority of the NIID-MSL is more prominent.

Furthermore, we take the Wallflower dataset to compare the running times of all competing methods. The results are listed in Table VI. The evaluation was performed in MATLAB R2018a on a computer with a twelve-core Inter(R) Core(TM) i7-8700K CPU @3.7GHz and 16GB RAM. From the table, it can be seen that comparing with the other methods, the computational cost of the proposed method is at moderate level overall. The FLSSS, JIVE and CSRL method run faster than the MSL, MISL and NIID-MSL methods, since they are constructed on the basis of $L_2$ loss, which is easy to handle during optimization. The MSL and MISL methods both introduce robust loss into the multi-view subspace learning: the former adopts $L_1$ loss and the latter uses Cauchy loss. Our proposed NIID-MSL, which is also focused on the robust issue, has a similar running speed compared with MSL and is evidently faster than MISL. Considering its better robust performance in all experiments, it should be reasonable to say that our proposed method is relatively efficient.

### D. Background subtraction on multi-view data

In this subsection, our method is applied to the problem of background subtraction on videos captured from multi-view cameras. Two multi-view video sequences including one outdoor scene\(^7\) (Fig. 9) and one indoor scene\(^8\) (Fig. 10) are employed, which are captured by 3 cameras located at different positions targeting at the similar scene. Each surveillance video contains thousands of frames. Due to the memory limitation of our computer, we only extracted about 200 frames (taking one from every five frames) to compose our evaluation data.

The FLSSS, MSL, CSRL, JIVE, MISL and our proposed NIID-MSL were then implemented on the dataset for the task. We first ran each method on multi-view videos to get the low-rank components as the background. Then we obtained the foreground by calculating the absolute values of differences between the original frames and the estimated backgrounds. The rank was set as 3 for all competing methods for fair comparison. The small rank setting is due to the fact that the tested videos are captured from statistic scenes, and their backgrounds are almost fixed and with strong correlation.

The results of the subtracted backgrounds and foregrounds on several typical frames by all competing methods are shown in Fig. 9 and Fig. 10 for easy comparison. It can be observed that all the competing methods can extract the background from the videos with slight difference in visualization, while the background images achieved by our method preserve relatively more elaborate details. Especially, for the outdoor scene video (Fig. 9), in which some people walked around the pictured scene, our proposed NIID-MSL method can effectively separate them absolutely as foreground so as to get perfect background. As a comparison, other competing methods all extract the background in a relatively coarse manner with varying degrees of human shadow, since their simple i.i.d. noise assumption does not finely fit the noise situations in both of the three practical scenarios. Furthermore, in order to quantify the comparison result, we carefully annotated the foreground mask of one typical frame in each dataset, and calculated the F-Measure of each view. The detailed F-Measure values are shown in Tables IX and X, and the visualization corresponds to Fig. 9 and Fig. 10, respectively. It is easy to see the superiority of our proposed method, both in quantitative metrics and visualization.

A more interesting observation is that, our NIID-MSL method is able to discover several modalities of the foreground information, through using GMM to fit the residuals in each

\(^7\)http://cs.binghamton.edu/~mrldata/pets2009
\(^8\)https://cvlab.epfl.ch/data/data-pom-index-php/
Fig. 9. From left to right: original video frames of 3 views of the Pets09 dataset, background extracted by FLSSS, MSL, JIVE, CSRL, MISL and NIID-MSL, together with their extracted noise images. For NIID-MSL, there are 4 Gaussian components to fit the noise at the top-level of HDP, and 3, 3 and 4 of which appear in the second level to noise of View 1, View 2 and View 3, respectively. The number below the foreground subtracted by the NIID-MSL method is the variance of the Gaussian components in the corresponding views.

Fig. 10. From left to right: original video frames of 3 views of the Lab dataset, background extracted by FLSSS, MSL, JIVE, CSRL, MISL and NIID-MSL, together with their extracted noise images. For NIID-MSL, there are 5 Gaussian components at the top-level of HDP, and 2, 4 and 4 of which appear in the second level to characterize the noise of View 1, View 2 and View 3, respectively. The number below the foreground subtracted by NIID-MSL method is the variance of the Gaussian components in the corresponding views.

The components of GMM with different variances reflect the noises with different scales and extents. What’s more, the GMM noise extracted from each view share some components of the top-level GMM of HDP by our method. Thus the variances of the Gaussian components in different views may or may not be the same, which depicts the correlations and distinctiveness of the noise in different views. Here we list the variance of the Gaussian components of each view in these two groups of experiments in Table VII and Table VIII, respectively. Besides, each component of GMM noise is visualized
TABLE IX
F-MEASURE VALUE COMPARISON OF ALL COMPETING METHODS ON PETS09 DATASET.

<table>
<thead>
<tr>
<th>Video</th>
<th>Methods</th>
<th>PLSS</th>
<th>MSL</th>
<th>JIVE</th>
<th>CSRL</th>
<th>MSL</th>
<th>NIID-MSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>View1</td>
<td>0.64</td>
<td>0.74</td>
<td>0.63</td>
<td>0.67</td>
<td>0.65</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>View2</td>
<td>0.65</td>
<td>0.87</td>
<td>0.67</td>
<td>0.75</td>
<td>0.71</td>
<td>0.88</td>
<td></td>
</tr>
<tr>
<td>View3</td>
<td>0.71</td>
<td>0.87</td>
<td>0.64</td>
<td>0.70</td>
<td>0.65</td>
<td>0.87</td>
<td></td>
</tr>
</tbody>
</table>

TABLE X
F-MEASURE VALUE COMPARISON OF ALL COMPETING METHODS ON LAB DATASET.

<table>
<thead>
<tr>
<th>Video</th>
<th>Methods</th>
<th>PLSS</th>
<th>MSL</th>
<th>JIVE</th>
<th>CSRL</th>
<th>MSL</th>
<th>NIID-MSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>View1</td>
<td>0.75</td>
<td>0.82</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>0.84</td>
<td></td>
</tr>
<tr>
<td>View2</td>
<td>0.73</td>
<td>0.77</td>
<td>0.65</td>
<td>0.77</td>
<td>0.77</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>View3</td>
<td>0.53</td>
<td>0.82</td>
<td>0.64</td>
<td>0.56</td>
<td>0.57</td>
<td>0.79</td>
<td></td>
</tr>
</tbody>
</table>

in Fig. 9 and Fig. 10 and their corresponding variance is also marked. These results apparently illustrate and substantiate the better noise fitting capability of our proposed method.

VII. CONCLUSION

We have proposed a new MSL method by sharing noise among different views under the Bayesian framework. Compared with most of the current MSL methods, which assume a simple i.i.d. noise distribution (e.g., Gaussian or Laplacian) on all views of data, our method simultaneously considers the distinctiveness and correlation of noises among different views of data and models the noise of each view as a DPGMM with its specific parameters. It can perform the MSL task with better noise fitting capability in practical applications and shows a more robust performance in the presence of complex noise. The effectiveness of our method is substantiated by experiments implemented on synthetic and real datasets. This study paves the path for modeling complex noises in machine learning, and the generalization of such noise modeling idea to other practical problems like multi-view text clustering/classification should be a meaningful research direction for future research.

ACKNOWLEDGEMENT

This research was supported by National Key R&D Program of China (2018YFB1004300) and China NSFC projects (61661160611, 11690011, 61603292, 61721002, U1811461).

REFERENCES

Sanping Zhou received the M.E. degree from Northwestern Polytechnical University, Xi'an, China, in 2015. He is currently pursuing the Ph.D. degree in Institute of Artificial Intelligence and Computing Technology, Xi'an Jiaotong University. His current research interests include medical image segmentation, deep learning, and probabilistic programming.


