Multi-view classifier based on Probabilistic Collaborative Representation and Latent Representation

Jian-wei Liu[1], Hui-dan Zhao[1], Run-kun Lu[1],Xiong-lin Luo[1]
[1] Department of Automation, China University of Petroleum, Beijing 102249
E-mail: liujw@cup.edu.cn
E-mail: 2805232563@qq.com E-mail:zsylrk@gmail.com
E-mail: luoxl@cup.edu.cn

Abstract: Multi-view is quite more effective at improving the training of model than merely using single view. However, most existing multi-view learning algorithms only either pay attention to consistency or complementary principle among views, not making full use of multi-view data. Due to its high complexity, algorithm considering both complementarity and consistency has limited ability to process large-scale data. On the basis of Probabilistic Collaborative Representa-tion based Classifier (ProCRC), we propose Probabilistic Collaborative Representation based Classifier for Multi-View (ProCRC-MV), which jointly maximizes the likelihood that a test example belongs to the co-subspace of each class. Learning subspace in the process of collaborative representation, considering consistency and complementarity concur-rently, ProCRC-MV can achieve promising classification performance. Meanwhile, it has low computational complexity, fast running speed, and can still maintain good performance when dealing with large-scale data. ProCRC-MV has the ability for subspace learning based on self-representation, so we combain latent representation learning for better search-ing subspace with ProCRC-MV to construct a novel classifier called LProCRC-MV, the ability of LProCRC-MV to process complex data is further enhanced comparing with ProCRC-MV.

Key Words: Probabilistic Collaborative Representation;Multi-View Learning; Complementarity and Consistency;Subspace Learning, Latent Representation

1 Introduction

In some practical problems, one thing can be described in many different ways from different perspectives, constituting multi-view of the thing. For instance, we can classify web pages according to the information contained in the web pages them-selves, or that contained in the hyperlinks linked to the web pages to classify them. Recently, multi-view learning has been received more and more attention in machine learning domain because of its convincing performance and high representation capabilities. Currently, representative multi-view learning algorithms can be mainly divided into the following categories: semi-supervised methods represented by co-training [1] and co-regularization [14], supervised learning methods represented by SVM-2K [6]and multiple kernel learning [9], subspace learning methods represented by CCA [4]and multi-view subspace learning with supervision [19] and multi-view deep learning([13]).

Despite these successes, however, there are two principal limitations in previous methods: 1) algorithm either focus on consistency or complementarity among views, which leads to poor performance. 2) algorithm which considers of consistency and complementarity of multiple views simultaneously, will impose increasing memory and computation burden, which is both expensive and complicated to be implemented and has limited ability to process large-scale data.

Concretely speaking, the consistency of multi-view refers to the characteristics in each view to describe the same attribute for one target object. The complementarity of multi-view means that one view describes some attribute of the target object, while the other views fail to show the attribute. Neglecting either consistency or complementarity will fail to make full use of the underlying information hidden in multi-view data. Some previous methods only focus on consistency among views, such as co-training and subspace learning approaches, especially CCA. Others only focus on complementarity among views, such as multiple kernel learning. The performance of these algorithms is generally relatively ordinary. Nevertheless, considering consistency together with complementarity, such as deep neural network, looking for consistent and unique latent representation for each view at the same time, the model is general-ly overcomplex, with limited ability to process large-scale data ,and prone to suffer from over-fitting.

To address above issues, we propose supervised Probabilistic Collaborative Representation based Classifier for Multi-View (ProCRC-MV) on the basis of Probabilis-tic Collaborative Representation based Classifier(ProCRC)[3] in this paper, which addresses practical and theoretical shortcomings discussed above and we show that it leads to improved performance on several tasks. ProCRC-MV takes into account consistency together with complementarity of multi-view, and has the ability of self-representation based subspace learning. Our experimental results indicate that

---

This work is supported by the Science Foundation of China University of Petroleum Beijing(2462018QZDX02)

978-1-7281-5855-6/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: China University of Petroleum. Downloaded on March 12,2021 at 00:38:03 UTC from IEEE Xplore. Restrictions apply.

ProCRC-MV costs less computational time and memory, demonstrating its better classification performance than some existing algorithms. To further improve the classification performance of our proposed model, we combine latent representation learning for better subspace searching with ProCRC-MV constructing a novel classifier, we dub it as LProCRC-MV, which further enhance the performance of ProCRC-MV in dealing with complex data.

ProCRC is inspired by CRC [21]. In ProCRC, all examples in training set are used to compose the dictionary for collaborative representation, these examples are called base vectors. Every example in training set can be represented by all base vectors. The probability of test examples in each class of cooperative subspace can be expressed and calculated. Examples belong to the class with the highest probability calculated. ProCRC has obvious probability explanation, and its performance is superior to SRC [18], CRC and many widely used classifiers such as SVM in many visual classification tasks. Collaborative representation is mostly used in image processing, but it has the capable of achieving amazing results in various fields like signal processing [8]. This paper is ingeniously applies cooperative representation to multi-view learning. ProCRC-MV employs all base vectors in training set and all base vectors in each class to represent a test example cooperatively. Then it will check which class is better for example reconstruction, the example belongs to the category that reconstruction error is smallest. When calculating reconstruction error, ProCRC-MV would take into account the influence of all features. For the attributes from different views that represent the same information, the change rule should be same. It can increase or decrease the representation weight before the base vector at the same time, that is to say, model considers consistency among views. For the unique features of each view, model will also focus on, because these features may mainly affect reconstruction error. ProCRC-MV will constantly balance representation weight to minimize reconstruction error, that is, considering consistency and complementarity of multi-view together.

However, in actual machine learning scenarios, we often encounter more complex data with high dimension, feature redundancy, entangling among various classes and existing data damage.In order to further improve the ability of ProCRC-MV to process complex data, latent representation learning for better searching subspace [20] is combined with ProCRC-MV. Learning latent representation can remove redundant information and noise in data, decrease the number of dimension of examples. Futhermore, while obtaining more representative representations, the consistency and complementarity properties of multi-view are still retained, which is better for finding suitable faithful subspace. ProCRC-MV itself has the ability of subspace learning based on self-representation, which is the main reason why ProCRC-MV works. By Combining ProCRC-MV with latent representation learning, we construct a new classifier LProCRC-MV, which is more conducive to cooperative representation and further improves classification performance to process complex data.

The work done in this paper is as follows:

(1) we construct supervised multi-view classifier called ProCRC-MV considering consistency in company with complementarity, ProCRC-MV has an outstanding classification performance, even when dealing with large-scale data set.

(2) Although ProCRC-MV takes into account complementarity and consistency at the same time, the complexity of ProCRC-MV is far less than other state-of-the-art algorithms that both leverage consistency and complementarity factors when dealing with large-scale data sets. ProCRC-MV has faster computing speed and need less memory in training process which can be seen in section 3.

(3) ProCRC-MV has subspace learning ability due to self-representation mechanism. In an effort to further boost up classification performance, we integrate ProCRC-MV with latent representation learning for more compact, and faithful subspace learning, and create a new classifier which we denote it as LProCRC-MV, which is more conducive to collaborative representation and has better classification performance in dealing with complex multi-view data.

(4) We have demonstrated ProCRC-MV's promising performance in contrast with other state-of-art multi-view classification approaches on a variety of real-world datasets. Due to incorporate more compact, and faithful latent representation learning, LProCRC-MV achieve the better classification performance compared with ProCRC-MV, which is verified in experimental comparison section .

2 ProCRC-MV and LProCRC-MV

2.1 Probabilistic Collaborative Representation based Classifier (ProCRC)

In ProCRC [3], the probabilities of a test example in co-subspace of each class can be expressed and calculated, and it belongs to the class with the highest probability. Suppose we have a training set $X = [X_1, \cdots, X_K]$, where X_k, $k \in \{1, \cdots, K\}$ is the data matrix of class k, and each column of X_k is an example vector, l_X represent the label set corresponding to X. Let S represent linear cooperative subspace spanned by all examples in X, X is defined as the dictionary for self-representation, each example in it is base vector. For each example x in S, it can be represented by a linear combination of all base vectors: $x = X\alpha$, α is representation weight matrix, the corresponding label for each example x is denoted as $l(x)$. Although all $X\alpha$ fall in S, the labels of these points are different, some may belong to $l(x)$, others not. Different points have different probabilities $P(l(x) \in l_X)$ for whether they belong to the label or not, which is related to the norm of α. $P(l(x) \in l_X)$ will be larger if the norm of α is smaller. An intuitive choice is to use Gaussian function to represent probability

$$P(l(x) \in l_X) \propto \exp(-c \|\alpha\|_2^2) \tag{1}$$

where c is a constant.

For the example y outside S, because the subspace spanned by examples in X should be composed of many planes, y may not be on these planes, but may be very close to these planes and has the same label $l(x)$ as the points on the plane. Therefore it can't be accurately co-represented

Authorized licensed use limited to: China University of Petroleum. Downloaded on March 12,2021 at 00:38:03 UTC from IEEE Xplore. Restrictions apply.

by $X\alpha$ like x. In order to get the label of y, we can find a data point x in S and calculate the probability with

$$P(l(y) \in l_X) = P(l(y) = l(x)|l(x) \in l_X) \cdot P(l(x) \in l_X) \tag{2}$$

The key idea of ProCRC is that $P(l(y) = l(x)|l(x) \in l_X)$ can be derived from the similarity between x and x. we

$$
\begin{bmatrix} y_{1,n_y} \\ y_{2,n_y} \\ \vdots \\ y_{m,n_y} \\ \vdots \\ y_{M,n_y} \end{bmatrix} = [x_1, x_2, \cdots, x_{n_X}, \cdots, x_{N_X}] \begin{bmatrix} \hat{\alpha}_{1,n_y} \\ \hat{\alpha}_{2,n_y n_y} \\ \vdots \\ \hat{\alpha}_{n_X,n_y} \\ \vdots \\ \hat{\alpha}_{N_X,n_y} \end{bmatrix}
$$

$$
= \hat{\alpha}_{1,n_y} \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{m,1} \\ \vdots \\ x_{M,1} \end{bmatrix} + \cdots + \hat{\alpha}_{n_X,n_Y} \begin{bmatrix} x_{1,n_X} \\ x_{2,n_X} \\ \vdots \\ x_{m,n_X} \\ \vdots \\ x_{M,n_X} \end{bmatrix} + \cdots + \hat{\alpha}_{N_X,n_y} \begin{bmatrix} x_{1,N_X} \\ x_{2,N_X} \\ \vdots \\ x_{m,N_X} \\ \vdots \\ x_{M,N_X} \end{bmatrix} \tag{15}
$$

self-representation based subspace learning method is preferred. However, subspace learning is often affected by the properties of raw features, such as high dimension, feature redundancy, entangling among various classes and existing data damage. These redundant features have irregular values in the same class, which makes the error of self-representation relatively large. The general formulation of subspace learning based on self-representation is

$$
\min_{Z} L(X, XZ) + \lambda \quad (Z) \tag{16}
$$

where $L(\cdot)$ and $\quad(\cdot)$ represent reconstruction loss function and regularization term respectively. $\lambda > 0$ is used to balance loss function and regularization term.

We can see that ProCRC-MV has the ability of learning subspace based on self-representa

-tion from its objective function, which is reasonable why ProCRC-MV has better classification performance as can be seen in section 3. However, ProCRC-MV suffers from limitation in processing complex data which can be shown in section 3. In order to further make examples more conducive to learning subspace, we incorporate the approach proposed in [20] to learn latent representation of multi-view, reducing redundancy in raw features and disentangling the relationship among them, then we utilize latent representations as input of ProCRC-MV to learn classifier, we dub it as ProCRC-MV based on Latent Representation (LProCRC-MV). [20] assumes that different views are originated from one latent representation, which contain underly-ing information and complete properties of data. Finding latent representation of multi-view can remove redundancy features and disentangle in multi-view data set without losing consistency and complementarity and reduce the number of dimension of the data.

Due to learning latent representation of multi-view in LProCRC-MV is unsupervised learning process, and to simplify symbolic representation as simple as possible, we overload the symbol X, below X represents the whole collection of training set $X = \{[x_i^1, \cdots, x_i^V]\}_{i=1}^{N_X}$ with N_X examples and V views and test set $Y = \{[y_i^1, \cdots, y_i^V]\}_{i=1}^{N_Y}$ with N_Y examples and V views. By doing so, the model can be prevented from over-fitting, which is caused by some noise in training set learned, or examples appear in test set that can't be found in training set.

Subspace learning process based on latent representation is as follows. Given a data set $X = \{[x_i^1, \cdots, x_i^V]\}_{i=1}^{N}$, with N examples and V views for each example. We assume that these data are generated from the same latent representation , as shown in Figure 1. where $P = \{p^1, \cdots, p^V\}$ is

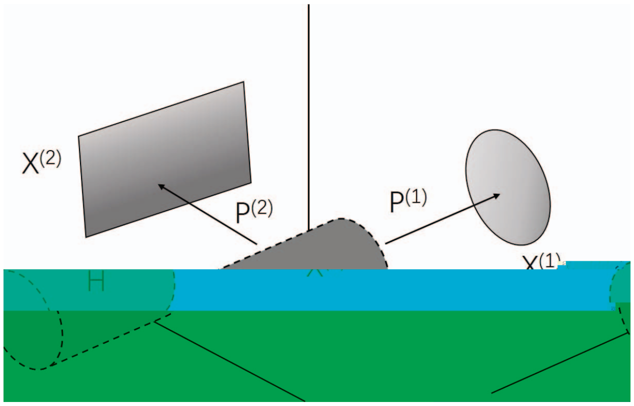

Figure 1: Demonstration of latent representation learning process.

a project mapping matrix of latent subspace. The example's representation can be formulated as following

$$
x_i^{(v)} = P^{(v)} h_i + e_i^{(v)} \tag{17}
$$

where $e_i^{(v)}$) is the sentati.72ecol-abonsiX below form

$$\begin{array}{l}\min\limits_{P,H,Z,E_V,E_S} \|E_V\|_{2,1} + \lambda_1 \|E_S\|_{2,1} + \lambda_2 \|Z\|_2 \\ s.t. X = PH + E_V, H = HZ + E_S, PP^T = I\end{array} \tag{21}$$

In order to make learned latent representation to be more advantageous to subspace learning of ProCRC-MV, we use L_2 norm of Z as regularization term, which is similar to the subspace learning method proposed in [5].

In order to maintain the integrity of the content of the paper, we summarize the optimization process for objective function (21) as follows, for detail content, please see [20] and references therein. (21) can be solved by Lagrange multiplier, and the loss function is rewritten into the form of ALM problem [11]:

$$\begin{array}{c} L(P, H, Z, E_V, E_S, J) \\ = \|E\|_{2,1} + \lambda \|J\|_2 + \quad (W_1, X - PH - E_V) \\ + \quad (W_2, H - HZ - E_S) + \quad (W_3, J - Z) \\ s.t. E = [E_V; E_S];\ PP^T = I \end{array} \tag{22}$$

where J is an auxiliary variable used to replace Z during calculation. W_1, W_2, W_3 are also auxiliary variable of ALM. [20] defines $\quad (C, D) = \frac{\mu}{2} \|D\|_F^2 + \langle C, D\rangle$, and $\langle A, B\rangle = tr(A^T B)$, $\mu > 0$ is penalty parameter, C is Lagrange multiplier. According to ADM optimization strategy [11], the loss function is divided into several subproblems, and then all variables are optimized by cyclic updating. The detail processes for Multi-View Latent Representation Learning (MV-LRL) can be divided into six stages:

(1) Optimization of mapping matrix P

$$\begin{array}{l} P^* = \arg\min_P \quad (W_1, X - PH - E_V) \\ s.t. PP^T = I \end{array} \tag{23}$$

According to Theorem 1 in [16], for objective function $\min_R \|Q - GR\|_F^2$, $s.t. R^T R$
$= RR^T = I$, the result of optimization is $R = UV^T$, where U and V are left and right singular values of $G^T Q$. we can rewrite (23) as follow

$$\begin{array}{l} P^* = \arg\min_P \quad (W_1, X - PH - E_V) \\ = \arg\min_P \frac{\mu}{2} \|X - PH - E_V + W_1/\mu\|_F^2 \\ = \arg\min_P \frac{\mu}{2} \|(X + W_1/\mu - E_V) - PH\|_F^2 \\ = \arg\min_P \frac{\mu}{2} \left\|(X + W_1/\mu - E_V)^T - H^T P^T\right\|_F^2 \end{array} \tag{24}$$

Using the results of Theorem 1 in [16], thus we have $(P^*)^T = UV^T$, where U and V are left and right singular values of $H(X + W_1/\mu - E_V)^T$.

(2) Optimization of latent matrix H

$$\begin{array}{l} H^* = \arg\min_H \quad (W_1, X - PH - E_V) \\ + \quad (W_2, H - HZ - E_S) \end{array} \tag{25}$$

Deriving the gradient of objective function with respect to H, and set gradient to 0, and the following formula is obtained

$$AH + HB = C \tag{26}$$

where

$$A = \mu P^T P \tag{27}$$

$$B = \mu(ZZ^T - Z - Z^T + I) \tag{28}$$

$$\begin{array}{l} C = (P^T W_1 + W_2(Z^T - I)) \\ +\mu(P^T X + E_S^T - P^T E_V - E_S Z^T) \end{array} \tag{29}$$

note that equality (26) is Sylvester equation [22], then solving equation (26) ,we can obtain H^*.

(3) Solution of subspace learning matrix Z

The parts of loss function which is related to solving Z

$$\begin{array}{l} Z^* = \arg\min_Z \quad (W_3, J - Z) \\ + \quad (W_2, H - HZ - E_S) \end{array} \tag{30}$$

correspondingly, the following result can be obtained

$$\begin{array}{l} Z^* = (H^T H + I)^{-1}[(J + H^T H - H^T E_S) \\ +(W_3 + H^T W_2)/\mu] \end{array} \tag{31}$$

(4) Updating formula of reconstruction error E

$$\begin{array}{c} E^* = \arg\min_E \|E\|_{2,1} + \quad (W_1, X - PH - E_V) \\ + \quad (W_2, H - HZ - E_S) \\ = \arg\min_E \frac{1}{\mu} \|E\|_{2,1} + \frac{1}{2} \|E - G\|_F^2 \end{array} \tag{32}$$

where G is constructed by vertically concatenating $X - PH - W_1/\mu$ and $H - HZ + W_2/\mu$. Relevant optimization algorithms are derived from [12].

(5) Solution of matrix J

$$\begin{array}{l} J^* = \arg\min{}_J \lambda \|J\|_* + \quad (W_3, J - Z) \\ = \frac{\lambda}{\mu} \|J\|_* + \frac{1}{2} \|J - (Z - W_3/\mu)\|_F^2 \end{array} \tag{33}$$

This is a low rank optimization problem, which can be solved by SVT method [2].

(6) Update multiplier

$$\begin{array}{c} W_1 = W_1 + \mu(X - PH - E_V) \\ W_2 = W_2 + \mu(H - HZ - E_S) \\ W_3 = W_3 + \mu(J - Z) \end{array} \tag{34}$$

The learning process of latent representation is shown in Algorithm1.

Algorithm 1 learning process of latent representation

```
Input: Multi-view data set: $X = \{[x_i^1, \cdots, x_i^V]\}_{i=1}^N$, the dimension $k$ of $H$ and Hyperparameter $\lambda$
Initialize $P = 0$, $E_V = 0$, $E_S = 0$, $J = Z = 0$, $W_1 = 0$,$W_2 = 0$, $W_3 = 0$, $\mu = 10^{-6}$, $\rho = 1.2$, $\varepsilon = 10^{-4}$, $\max_\mu = 10^6$, initialize $H$ with random value.
repeat
    Update variables $P$, $H$, $Z$, $E_V$, $E_S$, $J$
    Update variables $W_1, W_2, W_3$
    Update parameter $\mu$ according to $\mu = \min(\rho\mu; \max_\mu)$
    Check convergence conditions: $\|X - PH - E_V\|_\infty < \varepsilon$, $\left\|H - PH - E_{\setminus S}\right\|_\infty < \varepsilon$ and $\|J - Z\|_\infty < \varepsilon$
until converged
Output: $P$, $H$, $Z$ and $E$
```

Every matrix is updated in turn, and optimal solution of all matrices is finally reached. After learning latent representation H, Hcan be divided into training and test set, then we utilize latent representations as input of ProCRC-MV to learn classifier, which is our proposed LProCRC-MV.

Authorized licensed use limited to: China University of Petroleum. Downloaded on March 12,2021 at 00:38:03 UTC from IEEE Xplore. Restrictions apply.

3 Experimental results

3.1 ProCRC-MV

Before classification, multi-view data set should be preprocessed. We merge all views in $X = \{[x_i^1, \cdots, x_i^V]\}_{i=1}^{N_X}$ together corresponding to each example x_i , and get a training set $X^{M \times N_X} = [x_1, \cdots, x_{n_X}, \cdots, x_{N_X}]$, $n_X \in \{1, \cdots, N_X\}$. We also merge all views in $Y = \{[y_i^1, \cdots, y_i^V]\}_{i=1}^{N_Y}$ together corresponding to each example y_i, and get a test set $Y^{M \times N_Y} = [y_1, \cdots, y_{n_Y}, \cdots, y_{N_Y}]$, $n_Y \in \{1, \cdots, N_Y\}$, where M represents the number of dimension of examples. Finally, we feed $X^{M \times N_X}$, $Y^{M \times N_Y}$ and corresponding label sets $\{l_i\}_{i=1}^{N_X}$, $\{l_i\}_{i=1}^{N_Y}$ to ProCRC-MV for classification.

This paper utilizes seven real-world multi-view datasets to verify the performance of ProCRC-MV. In particular, Cornell and Texas are subset of WebKB. Table 1 list the characteristics of these datasets, where V is the number of views, K is the number of classes.

Table 1: Characteristics of datasets

Data Set	Instances	V	Dimension Number	K
YaleFace	256	2	2016 for all	8
Leaves	96	3	64 for all	6
ORL	400	3	3304/4096/6750	40
BBC	685	4	4633/4659/4684/4665	5
Cornell	195	2	585/1703	5
Texas	187	2	561/1703	5
Wisconsin	265	2	795/1703	5

We evaluate the performance of ProCRC-MV on classification tasks by comparing it with several baseline multi-view learning algorithms, including DICS [15], multiNMF [10], GMVNMF [17] and MVCC [7]. For fair comparison, we choose the parameters of all algorithms within the range that author suggested.

DICS is a multi-view learning algorithm based on NMF, exploring discriminative and non-discriminative information among different views and generating corresponding features for classification from all subspaces.

MultiNMF is an NMF-based multi-view clustering algorithm, which can get compatible clustering results among multiple views.

GMVNMF is a multi-view feature extraction framework based on NMF, which combines the local geometric structure information of each view. The extracted features take into account the internal relevance among views and are further used to generate clustering results.

MVCC is a multi-view clustering method based on conceptual factorization with local manifold regularization, getting a consistent representation of multiple views.

The experimental results are shown in Table 2 and Table 3. All datasets are divided into training and testing data in a ratio of 0.8:0.2 for ProCRC-MV, because it does not need validation set. The ratio of other algorithms' training, verification and test set is 0.6:0.2:0.2.

We can see the accuracy and F1 score of ProCRC-MV are higher than other algorithms for almost all data set except for Wisconsi. For these data sets, ProCRC-MV has better classification performance and is more stable, because it takes into account consistency and complementarity of multi-view data, and has ability of subspace learning based on self-representation. YaleFace and ORL are human face data sets, and ProCRC-MV was originally proposed for face recognition. For these data sets, different views are more similar in form, such as straight face and side face, photos with different light intensity, color and black-and-white photos. They are generally pixel data, unlike web pages. ProCRC-MV can easily build a comprehensive and low-rank dictionary for face images, therefore the classification effect is better. BBC is a motion picture set, which also has the characteristics of pictures. For classification of image datasets, the stability of the model is also good enough. Leaves is relatively simple, and each algorithm achieves better classification effect on Leaves, ProCRC-MV achieves the state-of-art result.

For more complex data sets, Cornell, Texas and Wisconsin, ProCRC-MV can also get better classification accuracy than other algorithms, which is inferior for Wisconsin. However, F1-Score is relatively low because of the high dimension of examples and the relatively complex relationship among classes in these datasets. Cornell and Texas contain webpages from four universities, and the corresponding labels are classified as professors, students, projects or other webpages. It is difficult for ProCRC-MV to establish a comprehensive and low rank dictionary, because high dimension, feature redundancy and entangling among various classes in raw data sets. To further enhance the ability of ProCRC-MV to process complex data, in the next section we combine ProCRC-MV with latent representation learning that is more conducive to subspace searching based on self-representation.

In the same hardware environment for training, the training time of ProCRC-MV is about 0.002s, and other algorithms on small data set are about 2 minutes, while on large data set they are more than 10 minutes, or even several hours. ProCRC-MV also uses far less storage space than other algorithms.

3.2 LProCRC-MV

We first input multi-view data set $X = \{[x_i^1, \cdots, x_i^V]\}_{i=1}^{N}$ into MV-LRL, note that X is the whole collection of training set $X = \{[x_i^1, \cdots, x_i^V]\}_{i=1}^{N_X}$ and test set $Y = \{[y_i^1, \cdots, y_i^V]\}_{i=1}^{N_Y}$. MV-LRL is an unsupervised learning algorithm. After learning latent representation H by MV-LRL, H will be divided into training and test set, and then classified by LProCRC-MV. The classification results of ProCRC-MV, LProCRC-MV and other algorithms are compared in Table 4 and Table 5.

We can see from Table 4 and Table 5 the accuracy of LProCRC-MV and F1-score are not as good as ProCRC in image data sets, because the raw image data sets can be constructed a comprehensive and low-rank dictionary, which can achieve good classification result. After using LProCRC-MV, the number of dimension of latent representation can be set artificially, and the selected number of dimension of latent representation is less than the genuine one, which makes the obtained latent representation lose

Authorized licensed use limited to: China University of Petroleum. Downloaded on March 12,2021 at 00:38:03 UTC from IEEE Xplore. Restrictions apply.

Table 2: Accuracy of different algorithms(%)

Algorithm	YaleFace	Leaves	ORL	BBC	Cornell	Texas	Wisconsin
DICS	88.9±3.4	97.6±2.3	92.6±5.4	90.4±2.0	72.5±5.5	76.7±5.4	85.0±4.7
MultiNMF	63.8±4.0	95.0±0.2	89.2±1.9	72.8±0.5	49.5±7.4	69.2±4.3	51.4±3.8
GMVNMF	50.1±2.6	95.4±0.1	57.5±0	37.8±1.2	41.4±1.7	58.0±1.7	53.0±1.5
MVCC	33.5±7.0	100±0	81.8±1.9	95.5±2.8	60.7±5.1	65.1±5.0	64.5±2.5
ProCRC	98.6±1.3	100±0	98.6±1.3	96.1±1.7	79.5±6.4	77.3±6.1	78.5±5.0

Table 3: F1-score of different algorithms(%)

Algorithm	YaleFace	Leaves	ORL	BBC	Cornell	Texas	Wisconsin
DICS	87.9±3.8	97.9±2.6	92.65±4.2	89.1±2.3	59.4±8.7	61.8±11.5	69.4±9.7
MultiNMF	62.5±4.7	93.9±1.3	87.0±3.0	71.2±0.1	32.9±6.2	49.1±5.0	37.8±3.7
GMVNMF	50.9±2.5	96.1±0.1	55.5±0	32.3±3.2	26.1±1.5	51.1±1.4	40.0±2.5
MVCC	32.2±5.9	100±0	84.2±2.2	93.4±6.4	42.8±4.6	53.6±8.5	.8±2.7
ProCRC	98.7±1.2	100±0	98.7±1.2	96.0±2.0	66.0±8.4	60.3±7.5	66.0±8.4

Table 4: Accuracy of different algorithms(%)

Algorithm	YaleFace	Leaves	ORL	BBC	Cornell	Texas	Wisconsin
DICS	88.9±3.4	97.6±2.3	92.6±5.4	90.4±2.0	72.5±5.5	76.7±5.4	85.0±4.7
MultiNMF	63.8±4.0	95.0±0.2	89.2±1.9	72.8±0.5	49.5±7.4	69.2±4.3	51.4±3.8
GMVNMF	50.1±2.6	95.4±0.1	57.5±0	37.8±1.2	41.4±1.7	58.0±1.7	53.0±1.5
MVCC	33.5±7.0	100±0	81.8±1.9	95.5±2.8	60.7±5.1	65.1±5.0	64.5±2.5
ProCRC	98.6±1.3	100±0	98.6±1.3	96.1±1.7	79.5±6.4	77.3±6.1	78.5±5.0
L- ProCRC	95.5±7.0	100±0	98.8±0.5	94.9±1.5	80.8±4.4	79.6±5.1	86.8±4.8

Table 5: F1-score of different algorithms(%)

Algorithm	YaleFace	Leaves	ORL	BBC	Cornell	Texas	Wisconsin
DICS	87.9±3.8	97.9±2.6	92.65±4.2	89.1±2.3	59.4±8.7	61.8±11.5	69.4±9.7
MultiNMF	62.5±4.7	93.9±1.3	87.0±3.0	71.2±0.1	32.9±6.2	49.1±5.0	37.8±3.7
GMVNMF	50.9±2.5	96.1±0.1	55.5±0	32.3±3.2	26.1±1.5	51.1±1.4	40.0±2.5
MVCC	32.2±5.9	100±0	84.2±2.2	93.4±6.4	42.8±4.6	53.6±8.5	50.8±2.7
ProCRC	98.7±1.2	100±0	98.7±1.2	96.0±2.0	66.0±8.4	60.3±7.5	66.0±8.4
L- ProCRC	95.4±7.7	100±0	98.0±0.8	95.0±0.3	69.6±6.1	65.5±4.9	63.8±6.3

partial information. For simple data sets such as Leaves, LProCRC-MV can also achieve good classification results. For relatively complex data sets, Cornell, Texas and Wisconsin, LProCRC-MV has been outperformed in both accuracy and F1-score. Especially accuracy has significant promotion, and the stability of the model has also been improved to a certain extent, because LProCRC-MV can find latent representation of multi-view, resulting in removing redundancy and disentangling in multi-view data set without losing consistency and complementarity and reduce the number of dimension of the data, which is more conducive to learning the subspace of the data.

Attention should be paid to the setting of hyperparameters and the number of dimension of H. In order to get better classification performance, we can use validation set to see which algorithm works better for ProCRC-MV and LProCRC-MV, but the computation and memory cost may increase for LProCRC-MV.

4 Conclusions and Discussion

we propose supervised ProCRC-MV to deal with multi-view classification problem, which jointly maximizes probability distribution over K co-subspaces that a test example belongs to. Because it comprehensively takes into account consistency and complementarity of multi-view in the process of collaborative representation, and has the ability of self-representation-based subspace learning, state-of-art classification performance is obtained. Furthermore, ProCRC-MV can avoid memory and computation burden, dislike other algorithms considering both complementarity and consistency among views, it displays promising result when processing large-scale data.

In order to further improve the ability for dealing with complex data, we integrate ProCRC-MV with latent representation learning for better searching subspace, we construct a novel classifier called LProCRC-MV. After learning latent representation, redundancy and entangling information are removed from examples, and the number of dimension of the data is reduced while retaining consistency and complementarity among views, which is beneficial to subspace learning based on self-representation. The ability of LProCRC-MV to process complex data is further enhanced comparing with ProCRC-MV.