Neural Processing Letters (2019) 50:189–226
https://doi.org/10.1007/s11063-019-10037-5

Semi-supervised One-Pass Multi-view Learning with Variable Features and Views

Changming Zhu[1,2] · **Duoqian Miao**[2]

Published online: 12 April 2019

Abstract
Traditional supervised multi-view learning machines aim to process multi-view data sets which consist of labeled instances from multiple views. While they cannot deal with semi-supervised data sets whose training instances consist of both labeled and unlabeled ones. Moreover, with the limitation of storage and process ability, some learning machines cannot process large-scale data sets. Furthermore, some instances maybe have missing features or views and traditional multi-view learning machines have no ability to process the data sets with variable features and views. Thus, this paper develops a semi-supervised one-pass multi-view learning with variable features and views (SOMVFV) so as to process the large-scale semi-supervised data sets with variable features and views. Related experiments on some supervised, semi-supervised, large-scale, and small-scale data sets validate the effectiveness of our proposed SOMVFV and we can get the following conclusions, (1) SOMVFV can process multiple kinds of special data sets; (2) compared with most learning machines used in our experiments, the better performance of SOMVFV is significant; (3) compared with missing views, missing features has a greater influence on the classification accuracy.

Keywords Semi-supervised multi-view learning · Variable views · Variable features · One-pass learning

✉ Changming Zhu
cmzhu@shmtu.edu.cn

Duoqian Miao
miaoduoqian@163.com

[1] College of Information Engineering, Shanghai Maritime University, Shanghai 201306, People's Republic of China

[2] Department of Computer Science and Technology, Tongji University, Shanghai 201804, People's Republic of China

1 Introduction

1.1 Background

Multi-view data set which consists of instances with multiple views has been paid more attention in recent years and each view indicates information of instances in a certain area. For example, there is a video data set and each video appears in multiple or different forms, i.e., visual, audio, and text. Then we regard each form as a view and this video data set as a multi-view one [1]. Moreover, for this multi-view data set, each view possesses multiple features. Take text view as an example. For each video, text view is a form for representation and text size, text color, text shape which reflect different information of text can be treated as the features of this view. In generally, for each view, we regard the number of features as the dimension of this view.

There are many multi-view data sets in real-world applications, for example, Cora [2] (used for text classification), IMDB [3] (used for movie classification), News Group [4] (used for text classification and text clustering), and Reuter [5] (used for document classification). The learning machine which is developed for processing these data sets is multi-view learning and in general, multi-view learning can be classified into some groups, for example, multi-view subspace learning methods [6,7], pre-fusion methods [8], late-fusion methods [9–11], disagreement-based methods [12–14], etc [15,16].

Although traditional multi-view learning has been widely used in multi-view clustering [17], handwritten digit recognition [18], human gait recognition [19,20], image recognition [21,22] and other fields [23–29], they still exist some problems.

1.2 The Problem of Traditional Multi-view Learning

First, in general, in real-world, instances of a data set can be divided into three parts. The first part is used for training a learning machine, the second part is used for validating the training results and adjusting the machine parameters, the third part is used for testing the effectiveness of the learning machine. In some cases, instances are divided into training part and testing part. Moreover, if the labels of instances are known, we call them labeled instances, otherwise, if the labels of them are unknown, we call them unlabeled instances. According to this definition, if all training or validation instances are labeled, we name this data set as supervised data set. If some training or validation instances are labeled and some of them are unlabeled, we name this data set as semi-supervised data set. In practice, traditional multi-view learning machines are always developed on the base of supervised data sets and those developed learning machines have no ability to process semi-supervised ones. This problem is treated as the first problem.

Second, most traditional multi-view learning machines are effective for small-scale data sets. In terms of these small-scale data sets, general storages can store all instances of a data set, for example, classical data set iris is a small-scale data set. While in real-world, with the coming of big-data age, more and more data sets are large-scale. For example, Youtube video data sets, news data sets, etc. Due to these data sets include hundreds of millions of instances, thus general storages have no ability to store all instances. Moreover, among those large-scale data sets, some of them are updated from time to time. Still take Youtube video and news data sets for example, as we know, videos and news can be uploaded by user whenever and wherever, thus sometimes, tens of thousands of videos and news are generated every minute and instances stored in storage can be varied with time lapse. For such large-scale

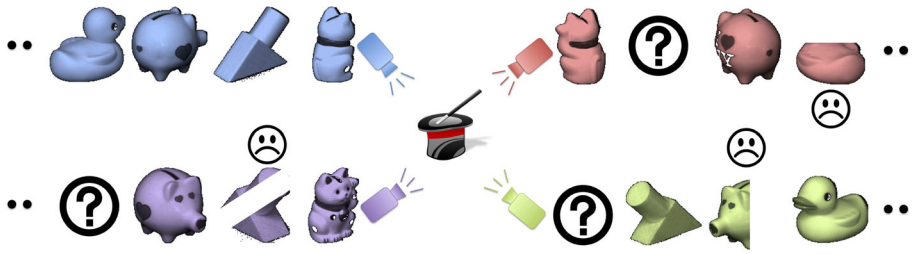


Fig. 1 Variable features and views in multi-view learning

or time-varying (i.e., frequent-updated) data sets, limited by the computation and storage ability, it is impossible for general storages to store all instances and these data sets are hard for traditional multi-view learning machines to process simultaneously since the traditional multi-view learning machines should use all instances for training and in many cases, the training instances are no change allowed. If we adopt traditional multi-view leaning machines to process those data sets, the performances will be reduced. Thus, having no ability to process large-scale and frequent-updated data sets is the second problem of traditional multi-view learning machines.

Third, in real-world applications, with the time lapse or some other factors, for example, the temporary failure of sensor or the man-made faults, some instances maybe loss some features or views. Please see Fig. 1 which is also given in [28]. In this figure, it takes the camera network as an example and multiple cameras capture the same scene from different angles at the same time. As [28] said, in common cases, multi-view learning machines adopt all information provided by these cameras for learning. However, some cameras could be temporarily out of action for natural or man-made reasons, thus some multi-view instances that are missing some views will be obtained (please see the question marks). Moreover, the cameras might be functional but could suffer from occlusions such that the views will have missing features (please see the crying expressions). In the worst case, missing views and missing features could simultaneously occur. For such a case, traditional multi-view learning machines have no ability to process the data sets with variable features and views and this becomes the third problem.

1.3 The Solutions to Problems of Traditional Multi-view Learning

According to above mentioned three kinds of problems of traditional multi-view learning machines, many scholars have developed some solutions in niche targeting.

First, in order to process semi-supervised multi-view data sets which are widely used in multi-view clustering [17], handwritten digit recognition [18], human gait recognition [19,20], image recognition [21,22], scholars have developed a series of semi-supervised multi-view learning machines [30]. For example, multi-view semi-supervised classification via adaptive regression (MVAR) [31], co-labeling [32], sparse Markov chain-based semi-supervised multi-instance multi-label method (Sparse-Markov) [33], semi-supervised multi-view hash model (SSMVH) [34], semi-supervised text classification with Universum learning (SSU) [35] are present popular used learning machines. The related experiments have validated that these semi-supervised multi-view learning machines possess better performances compared with the traditional supervised ones.

Second, for the large-scale and frequent-updated multi-view data sets, some online learning machines have been developed and one classical machine of them is one-pass multi-view (OPMV) [36] learning machine. For OPMV, it can go through the data only once and without storing the entire data set. Moreover, its model can be updated constantly with new instances coming. For example, according to [36], suppose the present OPMV is trained on the base of instances derived from 13:00 to 14:00, then it can be used to test the unlabeled instances derived from 13:00 to 14:00. For unlabeled instances derived from 14:00 to 14:10, we can also use the present OPMV to test, but the performance maybe not very good. Thus, we can adopt labeled instances derived from 14:00 to 14:10 to update the present OPMV and enhance the test performance. While OPMV is only feasible for supervised data sets and thus some scholars have developed a semi-supervised OPMV (SSOPMV) to process large-scale and frequent-updated semi-supervised multi-view data sets [37].

Third, in terms of the data sets with variable features and views, Xu et. al [28] have developed a multi-view learning with incomplete views (MVL-IV). For MVL-IV, it exploits the connections between multiple views and suggests that different views are generated from a shared subspace which makes the MVL-IV can estimate the incomplete views by integrating the information from the other observed views through this subspace. While MVL-IV aims to process data sets with variable views. What's more, some references concern missing features, including [38–41]. Moreover, Hou et. al have developed an one-pass learning with incremental and decremental features (OPID) [42] which not only considers variable features but also possesses the ability to process large-scale and frequent-updated data sets even though those data sets are single-view ones.

1.4 Proposal, Innovation, Motivation, and Contribution of SOMVFV

Although there are many learning machines have been developed to process the above three problems, to the best of our knowledge, there is no learning machine has been developed to process these problems simultaneously. Thus, this manuscript develops a semi-supervised one-pass multi-view learning with variable features and views (SOMVFV).

The innovation of SOMVFV is that it is the first time to propose a method to process semi-supervised, large-scale, frequent-updated, and variable features and views possessed data sets. Compared with the traditional semi-supervised or supervised multi-view learning machines, one-pass learning machines, and learning machines aiming to process data sets with missing views or features, SOMVFV possesses more application fields.

The motivation of SOMVFV is that it can be treated as a extended version of OPID and it has a high scalability and learning ability. In terms of the high scalability, if the processed data sets are small-scale, or are supervised, or are never-updated, or have full views and features, namely, without missing views and features, then SOMVFV can be degenerated the multi-view learning machines which are mentioned in Sect. 1.3. So we say SOMVFV has a high scalability. In terms of the high learning ability, since SOMVFV has an ability to process multiple kinds of special data sets including the semi-supervised ones, large-scale ones, frequent-updated ones, and ones with variable features and views, thus it can be found that SOMVFV has a high learning ability.

The contributions of SOMVFV are that (1) it has an ability to process semi-supervised one-pass multi-view data sets with variable features and views; (2) it has a better performance compared with a series of traditional multi-view learning machines; (3) it is feasible for many real-world applications.

1.5 Framework of Our Work

The rest of this paper is organized as below. Section 2 reviews the related work about SOMVFV. Section 3 shows the framework of the proposed SOMVFV. Sections 4 and 5 give the experiments about SOMVFV. Section 6 gives the conclusions and future work. Section 7 shows the appendix.

2 Related Work

According to the above contents, since SOMVFV is a learning machine to process semi-supervised one-pass multi-view data sets with variable features and views, thus in this section, we review the mechanisms of semi-supervised learning machines, one-pass learning machines, and ones with variable features or views.

2.1 Semi-supervised Learning Machine

With the complication of structures of multi-view data sets, multi-view learning machines have also gone through a series of changes. One classical series is semi-supervised learning machine which can be divided into several kinds including self-training, semi-supervised support vector machines, co-training, graph-based methods, and others [43–77].

In terms of the design of a traditional semi-supervised learning machine, there are two kinds of optimization procedures. For the first kind, at each iteration, one should first to adopt the labeled training instances to train the semi-supervised learning machine and get the decision function. Then, one should apply the decision function to classify the unlabeled training instances. If the predicted labels of some unlabeled training instances are same as their truth labels, one can add these unlabeled ones into the labeled training set. Then one adopts the updated labeled training instances to update the learning machine and its decision function. One should always carry out the iterations until all unlabeled training instances have been added into the labeled set or the iteration numbers attains to the maximum number. Finally, one finishes the procedure of optimizing a semi-supervised learning machine and the optimized learning machine can be used for testing the unlabeled test instances. For the second kind, the labeled training instances and the unlabeled ones are adopted simultaneously. Simply speaking, with this kind, the objective function includes both the labeled and unlabeled training instances and we can optimize and update the parameters of the learning machines with all labeled and unlabeled training instances used simultaneously.

2.2 One-Pass Learning Machine

With the coming of big-data age, one-pass leaning machine has been developed. This kind of learning machine aims to process large-scale and frequent-updated data sets. Up to the present, there are many one-pass learning machines have been developed and OPMV [36], SSOPMV [37], one-pass local online learning algorithm (LOL) [78], one-pass closed-form solution (OPML) [79], one-pass-throw-away class-wise learning (OPTACW) [80] are widely used ones.

In terms of an one-pass leaning machine, we always update the model once a new instance arrives. Simply speaking, the parameter of a model ω is related to the instances. When $(i+1)$-th instance arrives, the ω is written as $\omega(i+1)$ and $\omega(i+1) = f(\omega(i))$ where $f(x)$ is a function to update the parameter.

2.3 Multi-view Learning Machine with Variable Features or Views

As we know, due to the temporary failure of sensor or the man-made faults, with the time lapse, some instances maybe miss some features or views. Thus a lot of learning machines, especially, the multi-view ones with variable features or views are developed in recent years. For example, the previous mentioned MVL-IV [28] which aims to process instances with missing views, learning machines given in references [38–41] which aim to process instances with missing features, and OPID [42] which aims to process frequent-updated instances with variable features.

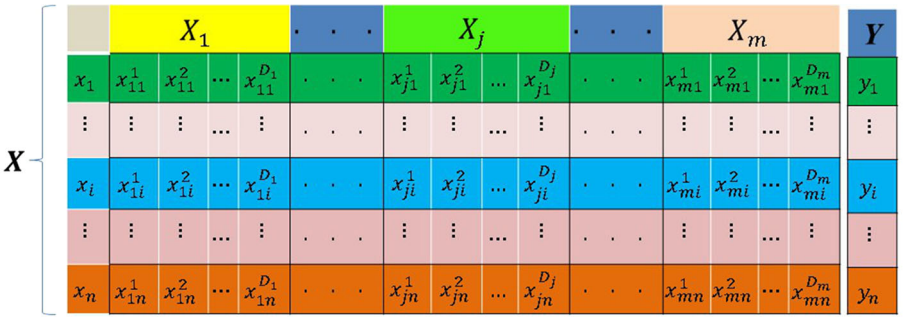
Among those methods, for MVL-IV, it assumes that different views are generated from a shared subspace and estimates the incomplete views by integrating the information from the other observed views through this subspace. Simply speaking, MVL-IV uses the low-rank assumption of the instance matrix to restore the missing views. For ones given in [38–41], they always adopt low-rank assumption and regularity to reconstruct the instance matrix to restore the missing features. The notion of them is similar with the one of MVL-IV. What's more, OPID [42] divides the features of instances into three parts, i.e., vanished features, survived features, and augmented features. Then it compresses important information of vanished features into functions of survived features and expand to include the augmented features. Moreover, OPID only needs to scan each instance once and does not need to store the whole data set.

3 Framework of SOMVFV

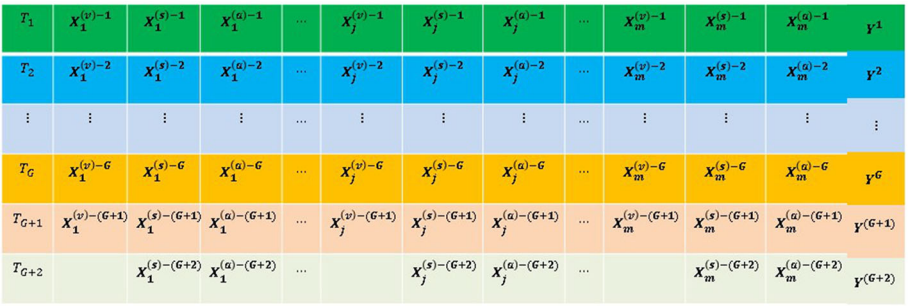
SOMVFV is a method for processing the large-scale semi-supervised multi-view data sets with variable features and views and the framework of SOMVFV is different from the ones of learning machines mentioned in Sect. 2. Now its framework is given below.

Suppose there is a large-scale multi-view data set X with n instances and m views (please see Fig. 2a) and X is also a semi-supervised binary-class data set. According to Fig. 2a, suppose x_{ji}^k denotes the k-th feature of j-th view of i-th instance, then $x_{ji} = (x_{ji}^1, x_{ji}^2, \cdots, x_{ji}^{D_j})$ denotes the j-th view of i-th instance, $x_i = (x_{1i}, x_{2i}, \cdots, x_{ji}, \cdots, x_{mi})$ denotes the i-th instance, and $X_j = \begin{pmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{ji} \\ \vdots \\ x_{jn} \end{pmatrix}$ denotes the j-th view where D_j is the dimension of j-th view.

For i-th instance, the label is y_i and $y_i \in \{+1, -1, NA\}$. $+1$ represents that the instance x_i belongs to class $+1$, -1 represents that x_i belongs to class -1, NA represents that the label of x_i is not given. Then $X = (X_1, X_2, \cdots, X_j, \cdots, X_m) = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix}$. Among X, l of them are labeled and u of them are unlabeled where $l + u = n$.



(a)



(b)

Fig. 2 Information of a large-scale multi-view data set and its changing at different time periods. In each time period, each view consists of three parts, i.e., vanished features, survived features, and augmented features

Then with the elapse of time, instances of X will be changed. Simply speaking, compared with the previous time period and in terms of information of features and views, some will be vanished at the next time period, some will be survived, and some are augmented in this time period. Thus, during $G+1$ known time periods, we can know the changing of X and at g-th time period, we can use $X_j^{(v)-g}$, $X_j^{(s)-g}$, and $X_j^{(a)-g}$ to denote the vanished features, survived features, and augmented features in j-th view respectively. The dimensions of them are $d_j^{(v)-g}$, $d_j^{(s)-g}$, and $d_j^{(a)-g}$ respectively. In other words, $x_{ji}^g = [x_{ji}^{(v)-g}, x_{ji}^{(s)-g}, x_{ji}^{(a)-g}]$. At the present time period, i.e., the $(G+2)$-th time period, since in practice, people always cannot predict which feature will be vanished, thus, at $(G+2)$-th time period, for X, only $X_j^{(s)-(G+2)}$ and $X_j^{(a)-(G+2)}$ are given (please see Fig. 2b). For the convenience of the declaration, we define the terms with the superscript g or $-g$ as the terms under g-th time period, for example, y_i^g represents the label of x_i at the g-th time period.

Then the aim of our developed SOMVFV is to use the instances at $(G+1)$-th time period to train a classifier and use it to test the labels of instances at $(G+2)$-th time period. Moreover, since the storage memory can only store the instances at one time period and in order to use the information given in the previous G time periods, we should try to summary the information into the $(G+1)$-th time period. Then the objective function of SOMVFV is given in Eq. (1) where $sign$ represents the sign function. For convenience, at g-th time period and for the j-th view, we define $\tilde{x}_{ji}^g = [x_{ji}^{(v)-g}, x_{ji}^{(s)-g}]$, $\dot{x}_{ji}^g = [x_{ji}^{(s)-g}]$, and $\bar{x}_{ji}^g = [x_{ji}^{(s)-g}, x_{ji}^{(a)-g}]$. Then the corresponding weights of the classifier are $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ respectively. Detailed discussion about Eq. (1) can be found in "Appendix".

$$\begin{aligned}\min L = \sum_{g=1}^{G}\Bigg\{\sum_{j=1}^{m}\Bigg[&\sum_{i=1}^{l}\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-y_i^g\right)^2+\sum_{i=1}^{l}\left(\dot{\omega}_j^g\dot{x}_{ji}^{g^T}-y_i^g\right)^2\\&+\sum_{i=1}^{l}\left(\bar{\omega}_j^g\bar{x}_{ji}^{g^T}-y_i^g\right)^2+\lambda_1\sum_{i'=1}^{u}\left(\tilde{\omega}_j^g\tilde{x}_{ji'}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji'}^{g^T}\right)^2\\&+\lambda_2\sum_{i'=1}^{u}\left(\dot{\omega}_j^g\dot{x}_{ji'}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji'}^{g^T}\right)^2+\lambda_3\sum_{i'=1}^{u}\left(\tilde{\omega}_j^g\tilde{x}_{ji'}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji'}^{g^T}\right)^2\\&+\lambda_4\sum_{i=1}^{l}\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji}^{g^T}\right)^2+\lambda_5\sum_{i=1}^{l}\left(\dot{\omega}_j^g\dot{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T}\right)^2\\&+\lambda_6\sum_{i=1}^{l}\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T}\right)^2+\rho\left(\left\|\tilde{\omega}_j^g\right\|_2^2+\left\|\dot{\omega}_j^g\right\|_2^2+\left\|\bar{\omega}_j^g\right\|_2^2\right)\Bigg]\Bigg\}\\ s.t.\quad &sign\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}\right)=sign\left(\dot{\omega}_j^g\dot{x}_{ji}^{g^T}\right)=sign\left(\bar{\omega}_j^g\bar{x}_{ji}^{g^T}\right)\end{aligned}\tag{1}$$

Then according to [36,37], the Lagrangian of Eq. (1) is given in Eq. (2).

$$\begin{aligned}\min L' = L&+\lambda_7\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji}^{g^T}\right)\\&+\lambda_8\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T}\right)+\lambda_9\left(\dot{\omega}_j^g\dot{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T}\right)\end{aligned}\tag{2}$$

In order to get the optimal $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$, we compute the partial derivative of L' with respect to them as below and the partial derivative of $\lambda_7(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji}^{g^T})+\lambda_8(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T})+\lambda_9(\dot{\omega}_j^g\dot{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T})$ with respect to $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ has been mixed into $\rho\tilde{\omega}_j^g$, $\rho\dot{\omega}_j^g$, and $\rho\bar{\omega}_j^g$.

$$\begin{aligned}\frac{\partial L'}{\partial\tilde{\omega}_j^g}=&2\sum_{i=1}^{l}\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-y_i^g\right)\tilde{x}_{ji}^g+2\lambda_1\sum_{i'=1}^{u}\left(\tilde{\omega}_j^g\tilde{x}_{ji'}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji'}^{g^T}\right)\tilde{x}_{ji'}^g\\&+2\lambda_3\sum_{i'=1}^{u}\left(\tilde{\omega}_j^g\tilde{x}_{ji'}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji'}^{g^T}\right)\tilde{x}_{ji'}^g+2\lambda_4\sum_{i=1}^{l}\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji}^{g^T}\right)\tilde{x}_{ji}^g\\&+2\lambda_6\sum_{i=1}^{l}\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T}\right)\tilde{x}_{ji}^g+\rho\tilde{\omega}_j^g\end{aligned}\tag{3}$$

$$\begin{aligned}\frac{\partial L'}{\partial\dot{\omega}_j^g}=&2\sum_{i=1}^{l}\left(\dot{\omega}_j^g\dot{x}_{ji}^{g^T}-y_i^g\right)\dot{x}_{ji}^g+2\lambda_1\sum_{i'=1}^{u}\left(\tilde{\omega}_j^g\tilde{x}_{ji'}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji'}^{g^T}\right)\left(-\dot{x}_{ji'}^g\right)\\&+2\lambda_2\sum_{i'=1}^{u}\left(\dot{\omega}_j^g\dot{x}_{ji'}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji'}^{g^T}\right)\dot{x}_{ji'}^g+2\lambda_4\sum_{i=1}^{l}\left(\tilde{\omega}_j^g\tilde{x}_{ji}^{g^T}-\dot{\omega}_j^g\dot{x}_{ji}^{g^T}\right)\left(-\dot{x}_{ji}^g\right)\\&+2\lambda_5\sum_{i=1}^{l}\left(\dot{\omega}_j^g\dot{x}_{ji}^{g^T}-\bar{\omega}_j^g\bar{x}_{ji}^{g^T}\right)\dot{x}_{ji}^g+\rho\dot{\omega}_j^g\end{aligned}\tag{4}$$

$$\frac{\partial L^{'}}{\partial \bar{\omega}_j^g} = 2\sum_{i=1}^{l}\left(\bar{\omega}_j^g \bar{x}_{ji}^{g^T} - y_i^g\right)\bar{x}_{ji}^g + 2\lambda_2\sum_{i^{'}=1}^{u}\left(\dot{\omega}_j^g \dot{x}_{ji^{'}}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji^{'}}^{g^T}\right)\left(-\bar{x}_{ji^{'}}^g\right) + 2\lambda_3\sum_{i^{'}=1}^{u}\left(\tilde{\omega}_j^g \tilde{x}_{ji^{'}}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji^{'}}^{g^T}\right)\left(-\bar{x}_{ji^{'}}^g\right) + 2\lambda_5\sum_{i=1}^{l}\left(\dot{\omega}_j^g \dot{x}_{ji}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji}^{g^T}\right)\left(-\bar{x}_{ji}^g\right) + 2\lambda_6\sum_{i=1}^{l}\left(\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji}^{g^T}\right)\left(-\bar{x}_{ji}^g\right) + \rho\bar{\omega}_j^g \tag{5}$$

Then we let $\frac{\partial L^{'}}{\partial \tilde{\omega}_j^g}$, $\frac{\partial L^{'}}{\partial \dot{\omega}_j^g}$, and $\frac{\partial L^{'}}{\partial \bar{\omega}_j^g}$ be 0, and we can get the optimal results of $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ [see the following Eqs. (6)–(8)]. Among these equations, the dimension of the identity matrix I is related to the ones of $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$.

$$\tilde{\omega}_j^g = \left\{2\sum_{i=1}^{l}\left(y_i^g + \lambda_4\dot{\omega}_j^g \dot{x}_{ji}^{g^T} + \lambda_6\bar{\omega}_j^g \bar{x}_{ji}^{g^T}\right)\tilde{x}_{ji}^g + 2\sum_{i^{'}=1}^{u}\left(\lambda_1\dot{\omega}_j^g \dot{x}_{ji^{'}}^{g^T} + \lambda_3\bar{\omega}_j^g \bar{x}_{ji^{'}}^{g^T}\right)\tilde{x}_{ji^{'}}^g\right\} \bullet \left\{2\sum_{i=1}^{l}\tilde{x}_{ji}^{g^T}\left(1+\lambda_4+\lambda_6\right)\tilde{x}_{ji}^g + 2\sum_{i^{'}=1}^{u}\tilde{x}_{ji^{'}}^{g^T}\left(\lambda_1+\lambda_3\right)\tilde{x}_{ji^{'}}^g + \rho I\right\}^{-1} \tag{6}$$

$$\dot{\omega}_j^g = \left\{\sum_{i=1}^{l} y_i^g \dot{x}_{ji}^g + \lambda_1\sum_{i^{'}=1}^{u}\tilde{\omega}_j^g \tilde{x}_{ji^{'}}^{g^T}\dot{x}_{ji^{'}}^g + \lambda_2\sum_{i^{'}=1}^{u}\bar{\omega}_j^g \bar{x}_{ji^{'}}^{g^T}\dot{x}_{ji^{'}}^g + \lambda_4\sum_{i=1}^{l}\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T}\dot{x}_{ji}^g + \lambda_5\sum_{i=1}^{l}\bar{\omega}_j^g \bar{x}_{ji}^{g^T}\dot{x}_{ji}^g\right\} \bullet \left\{\left(1+\lambda_4+\lambda_5\right)\sum_{i=1}^{l}\dot{x}_{ji}^{g^T}\dot{x}_{ji}^g + \left(\lambda_1+\lambda_2\right)\sum_{i^{'}=1}^{u}\dot{x}_{ji^{'}}^{g^T}\dot{x}_{ji^{'}}^g + \frac{\rho}{2}I\right\}^{-1} \tag{7}$$

$$\bar{\omega}_j^g = \left\{\sum_{i=1}^{l} y_i^g \bar{x}_{ji}^g + \lambda_2\sum_{i^{'}=1}^{u}\dot{\omega}_j^g \dot{x}_{ji^{'}}^{g^T}\bar{x}_{ji^{'}}^g + \lambda_3\sum_{i^{'}=1}^{u}\tilde{\omega}_j^g \tilde{x}_{ji^{'}}^{g^T}\bar{x}_{ji^{'}}^g + \lambda_5\sum_{i=1}^{l}\dot{\omega}_j^g \dot{x}_{ji}^{g^T}\bar{x}_{ji}^g + \lambda_6\sum_{i=1}^{l}\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T}\bar{x}_{ji}^g\right\} \bullet \left\{\left(1+\lambda_5+\lambda_6\right)\sum_{i=1}^{l}\bar{x}_{ji}^{g^T}\bar{x}_{ji}^g + \left(\lambda_2+\lambda_3\right)\sum_{i^{'}=1}^{u}\bar{x}_{ji^{'}}^{g^T}\bar{x}_{ji^{'}}^g + \frac{\rho}{2}I\right\}^{-1} \tag{8}$$

While since in each time period, the instances always arrive one by one, thus if people has not an enough storage memory, it cannot get the optimal results of $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ directly. In

order to get the optimal results, we update the $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ with an one-pass optimization method and get the results as below.

If at the g-th time period, when the $(i+1)$-th instance arrives and it is a labeled instance, then the update of $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ is given below where $\tilde{\omega}_{j(i+1)}^g$, $\dot{\omega}_{j(i+1)}^g$, and $\bar{\omega}_{j(i+1)}^g$ represent $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ at $(i+1)$-th instance arrives respectively.

$$\begin{aligned}\tilde{\omega}_{j(i+1)}^g &= \left\{-\frac{\partial L^{'}}{\partial \tilde{\omega}_{j(i)}^g} + (2y_{i+1}^g + 2\lambda_4 \dot{\omega}_{j(i)}^g \dot{x}_{j(i+1)}^{g^T} \right.\\ &\left. + 2\lambda_6 \bar{\omega}_{j(i)}^g \bar{x}_{j(i+1)}^{g^T}) \tilde{x}_{j(i+1)}^g \right\} \bullet \left\{2\tilde{x}_{j(i+1)}^{g^T}(1+\lambda_4+\lambda_6)\tilde{x}_{j(i+1)}^g + \rho I\right\}^{-1} \quad (9)\\ \dot{\omega}_{j(i+1)}^g &= \left\{-\frac{\partial L^{'}}{\partial \dot{\omega}_{j(i)}^g} + 2y_{i+1}^g \dot{x}_{j(i+1)}^g + 2\lambda_4 \tilde{\omega}_{j(i)}^g \tilde{x}_{j(i+1)}^{g^T} \dot{x}_{j(i+1)}^g \right.\\ &\left. + 2\lambda_5 \bar{\omega}_{j(i)}^g \bar{x}_{j(i+1)}^{g^T} \dot{x}_{j(i+1)}^g \right\} \bullet \left\{(1+\lambda_4+\lambda_5)\dot{x}_{j(i+1)}^{g^T} \dot{x}_{j(i+1)}^g + \frac{\rho}{2} I\right\}^{-1} \quad (10)\\ \bar{\omega}_{j(i+1)}^g &= \left\{-\frac{\partial L^{'}}{\partial \bar{\omega}_{j(i)}^g} + y_{i+1}^g \bar{x}_{j(i+1)}^g + \lambda_5 \dot{\omega}_{j(i)}^g \dot{x}_{j(i+1)}^{g^T} \bar{x}_{j(i+1)}^g \right.\\ &\left. + \lambda_6 \tilde{\omega}_{j(i)}^g \tilde{x}_{j(i+1)}^{g^T} \bar{x}_{j(i+1)}^g \right\} \bullet \left\{(1+\lambda_5+\lambda_6)\bar{x}_{j(i+1)}^{g^T} \bar{x}_{j(i+1)}^g + \frac{\rho}{2} I\right\}^{-1} \quad (11)\end{aligned}$$

With similar procedure, if the $(i+1)$-th instance is unlabeled, the update of $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ is given below.

$$\begin{aligned}\tilde{\omega}_{j(i'+1)}^g &= \left\{-\frac{\partial L^{'}}{\partial \tilde{\omega}_{j(i)}^g} + (2\lambda_1 \dot{\omega}_{j(i)}^g \dot{x}_{j(i'+1)}^{g^T} + 2\lambda_3 \bar{\omega}_{j(i)}^g \bar{x}_{j(i'+1)}^{g^T}) \tilde{x}_{j(i'+1)}^g \right\}\\ &\bullet \left\{2\tilde{x}_{j(i'+1)}^{g^T}(\lambda_1+\lambda_3)\tilde{x}_{j(i'+1)}^g + \rho I\right\}^{-1} \quad (12)\\ \dot{\omega}_{j(i'+1)}^g &= \left\{-\frac{\partial L^{'}}{\partial \dot{\omega}_{j(i)}^g} + \lambda_1 \tilde{\omega}_{j(i)}^g \tilde{x}_{j(i'+1)}^{g^T} \dot{x}_{j(i'+1)}^g + \lambda_2 \bar{\omega}_{j(i)}^g \bar{x}_{j(i'+1)}^{g^T} \dot{x}_{j(i'+1)}^g \right\}\\ &\bullet \left\{(\lambda_1+\lambda_2)\dot{x}_{j(i'+1)}^{g^T} \dot{x}_{j(i'+1)}^g + \frac{\rho}{2} I\right\}^{-1} \quad (13)\\ \bar{\omega}_{j(i'+1)}^g &= \left\{-\frac{\partial L^{'}}{\partial \bar{\omega}_{j(i)}^g} + \lambda_2 \dot{\omega}_{j(i)}^g \dot{x}_{j(i'+1)}^{g^T} \bar{x}_{j(i'+1)}^g + \lambda_3 \tilde{\omega}_{j(i)}^g \tilde{x}_{j(i'+1)}^{g^T} \bar{x}_{j(i'+1)}^g \right\}\\ &\bullet \left\{(\lambda_2+\lambda_3)\bar{x}_{j(i'+1)}^{g^T} \bar{x}_{j(i'+1)}^g + \frac{\rho}{2} I\right\}^{-1} \quad (14)\end{aligned}$$

According to the procedure of these G time periods, in a same time period, the dimensions of $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ are assumed be the same while in different time periods, the dimensions maybe changed. For example, from T_{g-1} to T_g, the dimensions are changed. Then in order to process instances in the g-th time period, we adopt the n-th instance x_n in $(g-1)$-th time period to update the weights. For example, if at the g-th time period, the 1-th instance x_1^g is labeled, then according to Eq. (9), $\tilde{\omega}_{j(1)}^g$ can be updated by Eq. (15). In this equation, if the dimensions of $\tilde{\omega}_{j(n)}^{g-1}$ and $\tilde{x}_{j1}^g$ are not same, we can expand their dimensions and the expended part can be fixed by value 0. For other weights and terms, the update method is same.

$$\tilde{\omega}_{j(1)}^{g} = \left\{ -\frac{\partial L^{'}}{\partial \tilde{\omega}_{j(n)}^{g-1}} + \left(2y_1^{g-1} + 2\lambda_4 \dot{\omega}_{jn}^{g-1} \dot{x}_{j1}^{g^T} + 2\lambda_6 \bar{\omega}_{jn}^{g-1} \bar{x}_{j1}^{g^T} \right) \tilde{x}_{j1}^{g} \right\} \bullet \left\{ 2\tilde{x}_{j1}^{g^T} (1 + \lambda_4 + \lambda_6) \tilde{x}_{j1}^{g} + \rho I \right\}^{-1} \tag{15}$$

After the procedure of G time periods, we can get $\tilde{\omega}_j^G$, $\dot{\omega}_j^G$, and $\bar{\omega}_j^G$. Then since these weights cover information of instances among the previous G time periods, thus we can use these weights to change the X_j^{G+1} to a compacted version. Concretely speaking, for $\tilde{x}_{ji}^{G+1}$, we define the $\tilde{z}_{ji}^{G+1} = \tilde{\omega}_j^G \tilde{x}_{ji}^{G+1^T}$ as the new representation of $\tilde{x}_{ji}^{G+1}$. For $\dot{x}_{ji}^{G+1}$, its new representation in $(G+1)$-th time period is $\dot{z}_{ji}^{G+1} = \dot{\omega}_j^G \dot{x}_{ji}^{G+1^T}$. For $\bar{x}_{ji}^{G+1}$, since at this time period, the augmented features are nothing to do with the previous G time periods, thus the new representation of $\bar{x}_{ji}^{G+1}$ in $(G+1)$-th time period is $\bar{z}_{ji}^{G+1} = [\dot{z}_{ji}^{G+1}, \bar{x}_{ji}^{G+1}]$. Then the dimensions of $\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, and $\bar{\omega}_j^{G+1}$ depend on the new representations of these features.

Then at the $(G+1)$-th time period, we optimize the following Eq. (16) so as to get the optimal $\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, and $\bar{\omega}_j^{G+1}$ where αs and μs are balance parameters and var represents the variance.

$$\min \alpha_1 A + \alpha_2 B + \alpha_3 C + \alpha_4 D \tag{16}$$

where

$$A = \sum_{j=1}^{m} \left[\frac{1}{2} \tilde{\omega}_j^{G+1} \tilde{\omega}_j^{G+1^T} + \mu_1 \sum_{i=1}^{l} ln \left(1 + e^{-y_i^{G+1} \tilde{z}_{ji}^{G+1} \tilde{\omega}_j^{G+1^T}} \right) \right] \tag{17}$$

$$B = \sum_{j=1}^{m} \left[\frac{1}{2} \dot{\omega}_j^{G+1} \dot{\omega}_j^{G+1^T} + \mu_2 \sum_{i=1}^{l} ln \left(1 + e^{-y_i^{G+1} \dot{z}_{ji}^{G+1} \dot{\omega}_j^{G+1^T}} \right) \right] \tag{18}$$

$$C = \sum_{j=1}^{m} \left[\frac{1}{2} \bar{\omega}_j^{G+1} \bar{\omega}_j^{G+1^T} + \mu_3 \sum_{i=1}^{l} ln \left(1 + e^{-y_i^{G+1} \bar{z}_{ji}^{G+1} \bar{\omega}_j^{G+1^T}} \right) \right] \tag{19}$$

$$D = \mu_4 \sum_{j=1}^{m} \left[\sum_{i^{'}=1}^{u} var \left(\tilde{\omega}_j^{G+1} \tilde{z}_{ji^{'}}^{G+1^T}, \dot{\omega}_j^{G+1} \dot{z}_{ji^{'}}^{G+1^T}, \bar{\omega}_j^{G+1} \bar{z}_{ji^{'}}^{G+1^T} \right) \right] \tag{20}$$

After that, the Eq. (16) can be rewritten as Eq. (21).

$$\begin{aligned} \min J = \sum_{j=1}^{m} \Bigg\{ & \frac{1}{2} \alpha_1 \tilde{\omega}_j^{G+1} \tilde{\omega}_j^{G+1^T} + \frac{1}{2} \alpha_2 \dot{\omega}_j^{G+1} \dot{\omega}_j^{G+1^T} + \frac{1}{2} \alpha_3 \bar{\omega}_j^{G+1} \bar{\omega}_j^{G+1^T} \\ & + \sum_{i=1}^{l} \Bigg[\alpha_1 \mu_1 ln \left(1 + e^{-y_i^{G+1} \tilde{z}_{ji}^{G+1} \tilde{\omega}_j^{G+1^T}} \right) + \alpha_2 \mu_2 ln \left(1 + e^{-y_i^{G+1} \dot{z}_{ji}^{G+1} \dot{\omega}_j^{G+1^T}} \right) \\ & + \alpha_3 \mu_3 ln \left(1 + e^{-y_i^{G+1} \bar{z}_{ji}^{G+1} \bar{\omega}_j^{G+1^T}} \right) \Bigg] \\ & + \alpha_4 \mu_4 \sum_{i^{'}=1}^{u} var \left(\tilde{\omega}_j^{G+1} \tilde{z}_{ji^{'}}^{G+1^T}, \dot{\omega}_j^{G+1} \dot{z}_{ji^{'}}^{G+1^T}, \bar{\omega}_j^{G+1} \bar{z}_{ji^{'}}^{G+1^T} \right) \Bigg\} \end{aligned} \tag{21}$$

Since at the $(G+1)$-th time period, instances also arrive in chronological order, thus we can also update $\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, and $\bar{\omega}_j^{G+1}$ with the following equations.

$$\tilde{\omega}_{j(i+1)}^{G+1} = \tilde{\omega}_{j(i)}^{G+1} - \frac{\partial J}{\partial \tilde{\omega}_{j(i)}^{G+1}} \tag{22}$$

$$\dot{\omega}_{j(i+1)}^{G+1} = \dot{\omega}_{j(i)}^{G+1} - \frac{\partial J}{\partial \dot{\omega}_{j(i)}^{G+1}} \tag{23}$$

$$\bar{\omega}_{j(i+1)}^{G+1} = \bar{\omega}_{j(i)}^{G+1} - \frac{\partial J}{\partial \bar{\omega}_{j(i)}^{G+1}} \tag{24}$$

where

$$\begin{aligned}\frac{\partial J}{\partial \tilde{\omega}_{j(i)}^{G+1}} = \sum_{j=1}^{m} \Bigg\{ & \alpha_1 \tilde{\omega}_{j(i)}^{G+1} \\ & + \alpha_1 \mu_1 \sum_{i=1}^{l'} \left[\frac{1}{1+e^{-y_i^{G+1} \tilde{z}_{ji}^{G+1} \tilde{\omega}_{j(i)}^{G+1^T}}} \left(-e^{-y_i^{G+1} \tilde{z}_{ji}^{G+1} \tilde{\omega}_{j(i)}^{G+1^T}} y_i^{G+1} \tilde{z}_{ji}^{G+1} \right) \right] \\ & + 2\alpha_4 \mu_4 \sum_{i'=1}^{u'} \left(\tilde{\omega}_{j(i)}^{G+1} \tilde{z}_{ji'}^{G+1^T} - \bar{\mu} \right) \tilde{z}_{ji'}^{G+1} \Bigg\}\end{aligned} \tag{25}$$

$$\begin{aligned}\frac{\partial J}{\partial \dot{\omega}_{j(i)}^{G+1}} = \sum_{j=1}^{m} \Bigg\{ & \alpha_2 \dot{\omega}_{j(i)}^{G+1} \\ & + \alpha_2 \mu_2 \sum_{i=1}^{l'} \left[\frac{1}{1+e^{-y_i^{G+1} \dot{z}_{ji}^{G+1} \dot{\omega}_{j(i)}^{G+1^T}}} \left(-e^{-y_i^{G+1} \dot{z}_{ji}^{G+1} \dot{\omega}_{j(i)}^{G+1^T}} y_i^{G+1} \dot{z}_{ji}^{G+1} \right) \right] \\ & + 2\alpha_4 \mu_4 \sum_{i'=1}^{u'} \left(\dot{\omega}_{j(i)}^{G+1} \dot{z}_{ji'}^{G+1^T} - \bar{\mu} \right) \dot{z}_{ji'}^{G+1} \Bigg\}\end{aligned} \tag{26}$$

$$\begin{aligned}\frac{\partial J}{\partial \bar{\omega}_{j(i)}^{G+1}} = \sum_{j=1}^{m} \Bigg\{ & \alpha_3 \bar{\omega}_{j(i)}^{G+1} \\ & + \alpha_3 \mu_3 \sum_{i=1}^{l'} \left[\frac{1}{1+e^{-y_i^{G+1} \bar{z}_{ji}^{G+1} \bar{\omega}_{j(i)}^{G+1^T}}} \left(-e^{-y_i^{G+1} \bar{z}_{ji}^{G+1} \bar{\omega}_{j(i)}^{G+1^T}} y_i^{G+1} \bar{z}_{ji}^{G+1} \right) \right] \\ & + 2\alpha_4 \mu_4 \sum_{i'=1}^{u'} \left(\bar{\omega}_{j(i)}^{G+1} \bar{z}_{ji'}^{G+1^T} - \bar{\mu} \right) \bar{z}_{ji'}^{G+1} \Bigg\}\end{aligned} \tag{27}$$

Moreover, for Eqs. (22)–(24), $i \geq 1$ and $\tilde{\omega}_{j(1)}^{G+1}$, $\dot{\omega}_{j(1)}^{G+1}$, $\bar{\omega}_{j(1)}^{G+1}$ can be gotten by Eq. (15). l' and u' represent the number of arrived labeled and unlabeled instances when the i-th instance arrives at $(G+1)$-th time period. What's more, in Eqs. (25)–(27), $\bar{\mu} = \frac{\tilde{\omega}_j^{G+1} \tilde{z}_{ji'}^{G+1^T} + \dot{\omega}_j^{G+1} \dot{z}_{ji'}^{G+1^T} + \bar{\omega}_j^{G+1} \bar{z}_{ji'}^{G+1^T}}{3}$. After the computation of Eqs. (22)–(27) with all n instances at $(G+1)$-th time period, we can get the optimal $\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, and $\bar{\omega}_j^{G+1}$.

Table 1 The framework of SOMVFV

Input:
The regularization parameters $\lambda > 0$;
The balance parameters $\alpha > 0$ and $\mu > 0$;
Weights for the 1-st time period for each view without any instance arrives $\tilde{\omega}_{j(0)}^1$, $\dot{\omega}_{j(0)}^1$, $\bar{\omega}_{j(0)}^1$;
The training instances X^g where $g = 1, 2, ..., (G+1)$;
The test instances X^{G+2}.
Output:$\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, $\bar{\omega}_j^{G+1}$, and class labels for X^{G+2}.
Initialize:each component of $\tilde{\omega}_{j(0)}^1$, $\dot{\omega}_{j(0)}^1$, and $\dot{\omega}_{j(0)}^1$ is $\frac{1}{d_j^{(v)-1}+d_j^{(s)-1}}$, $\frac{1}{d_j^{(s)-1}}$, $\frac{1}{d_j^{(s)-1}+d_j^{(a)-1}}$ respectively.
Training
for $g = 1:1:G$
for $j = 1:1:m$
for $i = 1:1:n$
update $\tilde{\omega}_{j(i)}^g$, $\dot{\omega}_{j(i)}^g$, and $\bar{\omega}_{j(i)}^g$ with instance x_{ji} arrives with Eqs. (3)∼(15);
end
end
end
Get $\tilde{\omega}_j^G = \tilde{\omega}_{j(n)}^G$, $\dot{\omega}_j^G = \dot{\omega}_{j(n)}^G$, and $\bar{\omega}_j^G = \bar{\omega}_{j(n)}^G$;
Get $\tilde{\omega}_{j(1)}^{G+1}$, $\dot{\omega}_{j(1)}^{G+1}$, $\bar{\omega}_{j(1)}^{G+1}$ by Eq. (15) and its similar versions;
for $j = 1:1:m$
for $i = 1:1:n$
Get $\tilde{\omega}_{j(i+1)}^{G+1}$, $\dot{\omega}_{j(i+1)}^{G+1}$, and $\bar{\omega}_{j(i+1)}^{G+1}$ by Eqs. (22)∼(27);
end
end
Let $\tilde{\omega}_j^{G+1} = \tilde{\omega}_{j(n)}^{G+1}$, $\dot{\omega}_j^{G+1} = \dot{\omega}_{j(n)}^{G+1}$, $\bar{\omega}_j^{G+1} = \bar{\omega}_{j(n)}^{G+1}$.
Test
Test X^{G+2} with $\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, $\bar{\omega}_j^{G+1}$ in each view.

Finally, in practice, at $(G+2)$-th time period, since we won't know which features will be vanished, thus according to Fig. 2b, for each view of X, only $X_j^{(s)-(G+2)}$ and $X_j^{(a)-(G+2)}$ are given. Then with the same operation given at $(G+1)$-th time period, due to the dimension of $\dot{z}_{ji}^{G+1}$ always equal or similar to the one of $\dot{x}_{ji}^{G+2}$, thus we first to add value 0 to $\dot{\omega}_j^{G+1}$ so as to make the dimension of it be equal to the one of $\dot{x}_{ji}^{G+2}$, then we use $\dot{z}_{ji}^{G+2} = \dot{x}_{ji}^{G+2}\dot{\omega}_j^{G+1^T}$ as the new representation of $\dot{x}_{ji}^{G+2}$. Then we define $\bar{z}_{ji}^{G+2} = [\dot{z}_{ji}^{G+2}, \bar{x}_{ji}^{G+2}]$. At last, we make the dimensions of $\bar{z}_{ji}^{G+2}$ and $\bar{\omega}_j^{G+1}$ be same and use $\bar{\omega}_j^{G+1}\bar{z}_{ji}^{G+2^T}$ to get the label of x_{ji}^{G+2}. Finally, the final label of x_i^{G+2} is $sign(\sum_{j=1}^m \bar{\omega}_j^{G+1}\bar{z}_{ji}^{G+2^T})$.

As a summary, the framework of our SOMVFV is given in Table 1.

4 Experiments

In order to validate the effectiveness of the proposed SOMVFV, we adopt some single-view, multi-view, large-scale, and small-scale data sets or learning machines for experiments and comparisons.

4.1 Experimental Setting

4.1.1 Data Set

According to Table 2, the used data sets in our work are different. Some of them are single-view, some of them are multi-view, some of them are large-scale, and some of them are

Table 2 The used data sets

	Small-scale	Large-scale
Single-view	Table 3	Table 4
Multi-view	Mfeat, Reuters, Corel [1]	Video, News [37]

Table 3 Used single-view small-scale data sets and details of them can be found in [81]

Order	Data sets	No. instances	No. features
1	AuC	690	14
2	BCW	699	9
3	GeD	1000	24
4	Glass	214	9
5	Heart	270	13
6	Iris	150	4
7	Letter	20,000	16
8	Liver	345	6
9	Pendigits	7494	16
10	PID	768	8
11	Satellite image	6435	36
12	Shuttle	58,000	9
13	Sonar	208	60
14	Thyroid	7200	21
15	Vowel	990	10
16	Waveform	5000	21
17	Waveform-noise	5000	40
18	Wine	178	13
19	BA	1372	4
20	TSE	5820	32
21	UKM	403	5
22	QSAR	1055	41

small-scale. By introducing different kinds of data sets, we can validate that our proposed SOMVFV is feasible for processing variable kinds of data sets.

What's more, for these data sets, we use different tables to show the information. In Table 3, AuC, BCW, GeD, Liver, BA, UKM, QSAR, PID, TSE represent Australian Card, Breast-Cancer-Wisconsin, German Data, Liver-disorders, Banknote Authentication, User Knowledge Modeling, QSAR biodegradation, Pima-Indians-Diabetes, Turkiye Student Evaluation respectively. Similarly, in Table 4, RLCP, GSADGM, PAMAP2, URL, YouTube-C, OR, Skin represent Record Linkage Comparison Patterns, Gas sensor array under dynamic gas mixtures, PAMAP2 Physical Activity Monitoring, URL Reputation, YouTube Comedy Slam Preference Data, Online Retail, Skin Segmentation respectively.

For the multi-view small-scale data sets, they are both used in [1] and Tables 5, 6, and 7 show the information of them respectively. In terms of these three data sets, (1) Mfeat consists of hand written digits (0–9) [82]. Each digit is a class and each instance consists of six views, i.e., Fourier coefficients of the character shapes (fou), profile correlations (fac), Karhunen-Love coefficients (kar), pixel averages in 2×3 windows (pix), Zernike moments(zer), and

Table 4 Used single-view large-scale data sets and details of them can be found in [81]

Order	Data sets	No. instances	No. features
23	HIGGS	11,000,000	28
24	HEPMASS	10,500,000	28
25	RLCP	5,749,132	12
26	SUSY	5,000,000	18
27	GSADGM	4,178,504	19
28	PAMAP2	3,850,505	52
29	URL	2,396,130	3,231,961
30	YouTube-C	1,138,562	3
31	OR	541,909	8
32	Skin	245,057	4

Table 5 Detailed information of Mfeat data set

View	No. instances	No. features	No. digits
fac	2000	216	10
fou	2000	76	10
kar	2000	64	10
pix	2000	240	10
zer	2000	47	10
mor	2000	6	10

Table 6 Detailed information of Reuters data set

View	No. documents	Vocabulary size
EN	18,758	21,513
FR	26,648	24,839
GR	29,953	34,279
SP	12,342	11,547
IT	24,039	15,506

Topic	No. documents	Per (%)
C15	18,816	16.84
CCAT	21,426	19.17
E21	13,701	12.26
ECAT	19,198	17.18
GCAT	19,178	17.16
M11	19,421	17.39

Table 7 Detailed information of Corel data set

View	No. instances	No. features	No. categories
Col-h	1000	32	10
Col-hl	1000	32	10
Col-m	1000	9	10
Coo-t	1000	16	10

morphological features (mor). (2) Reuters[1] consists of machine translated documents which are written in five different languages which are treated as five views [83,84]. These five languages are English (EN), French (FR), German (GR), Italian (IT), and Spanish (SP) and each document can be translated from one language to another language. Moreover, the documents are also categorized into six different topics, i.e., classes. (3) Corel[2] is extracted from a Corel image collection [82] and it consists of 68,040 photos from various categories. In our experiments, we randomly select 1000 photos from 10 categories and each category has 100 photos. The 10 categories, i.e., classes are C0-Africa, C1-Beach, C2-Building, C3-Buses, C4-Dinosaurs, C5-Elephants, C6-Flowers, C7-Horses, C8-Mountains and C9-Food. For this data set, four views are adopted. They are color histogram (abbr. Col-h), color histogram layout (abbr. Col-hl), color moments (abbr. Col-m), and co-occurrence texture (abbr. Coo-t). Each view represents a feature set.

In terms of the multi-view large-scale data sets Video and News, they are also frequent-updated. Video is the abbreviation of videos from YouTube and News is the abbreviation of news from Shanghai Media Group (SMG) which is reported every day. For Video, it consists of three views: visual, audio, and text. For News, it consists of four views: visual, audio, text, and language. Since it has been validated that for those large-scale and frequent-updated data sets, most of the instances are unlabeled due to labeling instances is a high-cost task, thus for these two frequent-updated data sets, we only adopt semi-supervised learning machines for experiments. Then for the experiments, we adopt the similar ways given in [37]. Concretely speaking, for Video, we select 100 art videos, 100 sports videos, 100 games videos, and 1000 unlabeled videos which are uploaded from 13:00 to 14:00, July 21, 2017 for training a semi-supervised learning machine and use the trained machine to label the videos uploaded from 14:00 to 14:20 July 21, 2017. Finally, we invite some staff to label the test videos by manual so as to judge whether the actual label is same as the predicted label and compute the performance of the learning machine. In order to not loss the generalization, we select more time periods for experiments. Information of the time period is given in Table 8. For News, the experimental way is same as the one for Video. But for News, we select 10 pieces of entertainment news, 10 pieces of sports news, 10 pieces of political news, and 100 unlabeled pieces of news in a time period. The sampling periods and test periods are also given in Table 8.

4.1.2 Learning Machine

Since the data sets have been divided into several kinds, thus we also adopt corresponding learning machines for comparisons. Table 9 shows the compared learning machines. In this table, name in bold indicates that the learning machine can process data sets with missing features further, name in bolditalic indicates that the learning machine can process data sets with missing views further, name in italic indicates that the learning machine can process data sets with both missing features and missing views further. In this table, SVM, CNN, FDROP, KARMA, LP, MV-LDA, MV-CCA, MV-LPP, MDIA-CNN, MEMR, MDA, LSDF, CSSSFS, SSOWMIL, MvSs-Zhu, MVML, AMVS represent support vector machine, convolutional neural network, tractable quadratic program for training robust classifiers, kernelized algorithm for risk-minimization with missing attributes, linear program, multi-view linear discriminant analysis, multi-view canonical correlation analysis, multi-view locality preserving

[1] http://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual%2C+Multiview+Text+Categorization+Test+collection

[2] http://archive.ics.uci.edu/ml/datasets/Corel+Image+Features

Table 8 Detailed information of sampling periods and test periods for Video and News

Day	Order	Sampling period	Test period	Order	Sampling period	Test period	Order	Sampling period	Test period
21/07/2017	1	13:00–14:00	14:00–14:20	2	14:30–15:30	15:30–15:50	3	16:00–17:00	17:00–17:20
22/07/2017	4	13:00–14:00	14:00–14:20	5	14:30–15:30	15:30–15:50	6	16:00–17:00	17:00–17:20
21/06/2017	7	13:00–14:00	14:00–14:20	8	14:30–15:30	15:30–15:50	9	16:00–17:00	17:00–17:20
22/06/2017	10	13:00–14:00	14:00–14:20	11	14:30–15:30	15:30–15:50	12	16:00–17:00	17:00–17:20

Table 9 The compared learning machines

Supervised	Non one-pass	One-pass
Single-view	SVM [85], CNN [77], **FDROP**[38], **KARMA** [41]	**OPID** [42], **LP** [39], LOL [78], OPML [79]
Multi-view	MV-LDA [86], MV-CCA [87], MV-LPP [88], ***MVL-IV*** [28],	
	MDIA-CNN [76], MEMR [75], *MDA* [73]	OPMV [36]
Semi-supervised	Non one-pass	One-pass
Single-view	LSDF [44], CSSSFS [53]	SSOWMIL [89]
Multi-view	MvSs-Zhu [22], MVML [18], co-graph [90], co-features [91], AMVS [92]	SSOPMV [37], *SOMVFV*

projections, multi-view dynamic image adaptive convolutional neural network, multi-view ensemble manifold regularization, multimodal deep autoencoder, locality sensitive discriminant feature, constraint scores for semi-supervised feature selection, semi-supervised online weighted multiple instance learning, multi-view semi-supervised learning proposed by Zhu, multiple-view multiple-learner, adaptive multi-view selection respectively.

4.1.3 Parameter Setting

In terms of the parameter settings of these learning machines except for our SOMVFV, we can refer to the related references. Then in terms of our SOMVFV, the parameter setting is given below. The regularization parameters λs (from λ_1 to λ_9) are selected from the set $\{0.1, 0.2, \ldots, 0.9\}$, the balance parameters αs (from α_1 to α_4) and μs (from μ_1 to μ_4) are selected from the set $\{2^{-4}, 2^{-3}, \ldots, 2^3, 2^4\}$, each component of $\tilde{\omega}^1_{j(0)}$, $\dot{\omega}^1_{j(0)}$, $\bar{\omega}^1_{j(0)}$ is 1.

What's more, since our SOMVFV can process data sets with variable features and views, thus in order to validate this point, for all the used data sets, we randomly remove some information of features or views in manual and use suffix (f) and (v) to represent that the data set misses some features or views respectively. For example, $Reuters$, $Reuters(f)$, $Reuters(v)$ represent the original Reuters, Reuters with missing features and Reuters with missing views. Sometimes, for experiments, we use $(f-x)$ and $(v-x)$ to denote the missing rate. For example, $Reuters(f-10)$ represents Reuters with missing 10% features. Indeed, for all used data sets, only Video and News are original semi-supervised. Thus, without loss of generality, for other supervised data sets, we also create the semi-supervised versions of them. Concretely speaking, we select 90% for training and the left 10% for test in random. Among the training set, we randomly select 25% training instances as the labeled ones and the left 75% instances are treated as the unlabeled training instances even though we know the truth labels (Indeed, more labeled training instances will bring a better performance in generally. But here, we only select 25% labeled ones just for validating the effectiveness of our SOMVFV in convenience). For convenience, we use $Dataset(u)$ for distinction. For example, $Reuters$ and $Reuters(u)$ represent the original supervised Reuters and the changed semi-supervised Reuters respectively. Of course, under such the definitions, we can use $Dataset(u/f/v/f-y-v-x)$ to represent the different forms of a data set. For example, $Reuters(u-f-10)$ indicates the semi-supervised Reuters with missing 10% features information and $Reuters(u-v-20-f-10)$ indicates the semi-supervised Reuters with missing 20% views information and 10% features information.

Furthermore, since one-pass learning machines can process frequent-updated data sets, thus when we process the unfrequent-updated data sets with these one-pass-related learning machines, we copy the whole data sets with 10 times and at each time, the information of instances are changed in random but the size is kept. We use the first 9 copies to train and the left copy to test.

In order to get the experimental results, we adopt 10-fold cross validation strategy and repeat the experiments for 10 times. Then the average results are given in the manuscript.

4.2 Classification Performance Comparison on Single-View Small-Scale Data Sets with Corresponding Learning Machines

We conduct the experiments on the given single-view small-scale data sets first. The data sets are shown in Table 3 and the used learning machines are selected from Table 9 according to the below different cases.

4.2.1 Case 1: Original Data Sets

We adopt SVM, FDROP, KARMA, and CNN for comparison so as to validate the effectiveness of the developed SOMVFV on the 22 single-view small-scale data sets. As we said before, FDROP and KARMA can process data sets with missing features, moreover, according to [38,41], they can also process complete single-view small-scale data sets. Related experimental results are given in case 1 of Table 10.

4.2.2 Case 2: Semi-supervised Data Sets

We adopt LSDF and CSSSFS for comparisons so that we can validate that our proposed SOMVFV is feasible for these 22 single-view small-scale data sets if they are semi-supervised. Case 2 of Table 10 shows the related experimental results.

4.2.3 Case 3: Original Data Sets with Missing Features

If these 22 original supervised single-view small-scale data sets miss some features, thus in order to judge that whether our SOMVFV can process supervised single-view small-scale data sets with missing features or not, we use FDROP and KARMA for comparisons. Corresponding experimental results are given in case 3 of Table 10 and we suppose these data sets miss 10% features.

4.2.4 Case 4: Semi-supervised Data Sets with Missing Features

If these 22 single-view small-scale data sets are semi-supervised and moreover, they miss some features, for example, 10% features. Then according to Table 9, since there is no other learning machine can process this case, thus we only use our SOMVFV for experiments just to validate that SOMVFV can process semi-supervised single-view small-scale data sets with missing features. Case 4 of Table 10 shows the experimental results.

Table 10 Classification performance (%) comparisons for single-view small-scale data sets with corresponding learning machines on different cases

Data set	Case 1					Case 4
	SVM	FDROP	KARMA	CNN	SOMVFV	SOMVFV
AuC	68.08	71.26	84.42	81.53	**84.78**	78.88
BCW	95.79	97.34	97.80	97.71	**98.71**	92.66
GeD	75.33	76.32	78.60	75.45	**79.34**	75.46
Glass	63.81	67.06	69.43	70.66	**74.65**	67.10
Heart	86.17	88.02	87.27	84.32	**88.90**	86.76
Iris	88.41	93.42	92.29	91.60	**97.33**	91.53
Letter	85.44	87.02	88.07	93.62	**94.65**	90.30
Liver	73.90	74.19	75.35	**75.79**	75.65	68.21
Pendigits	87.70	91.77	96.46	94.07	**96.91**	94.73
PID	72.11	75.95	76.21	75.66	**77.73**	73.89
Satellite Image	73.28	82.88	76.92	76.15	**83.65**	79.82
Shuttle	73.89	85.44	77.65	80.83	**85.95**	78.68
Sonar	66.40	68.40	70.82	72.12	**76.92**	72.73
Thyroid	81.03	84.62	90.84	**93.06**	92.13	85.97
Vowel	49.25	49.36	49.28	46.23	**49.59**	46.74
Waveform	76.24	80.34	80.06	**81.72**	80.91	75.71
Waveform-noise	72.66	75.76	84.53	83.34	**86.37**	81.06
Wine	89.50	96.65	93.55	92.34	**96.67**	90.52
BA	81.72	83.17	82.26	78.97	**84.90**	80.05
TSE	74.85	79.08	75.29	77.20	**79.13**	69.91
UKM	70.24	71.72	78.80	75.80	**81.93**	75.77
QSAR	70.72	72.06	70.97	74.73	**77.57**	70.73
Avg.	76.21	79.63	80.77	80.59	**83.84**	78.51

Data set	Case 2			Case 3		
	LSDF	CSSSFS	SOMVFV	FDROP	KARMA	SOMVFV
AuC	75.55	77.45	**83.65**	65.15	79.55	**79.88**
BCW	92.05	88.14	**96.34**	97.00	89.43	**98.13**
GeD	77.31	73.19	**79.24**	71.69	76.58	**76.70**
Glass	69.26	67.09	**73.14**	66.46	64.29	**69.14**
Heart	78.89	80.29	**87.19**	80.95	87.16	**87.98**
Iris	91.67	91.88	**96.48**	89.31	84.68	**91.57**
Letter	91.00	84.40	**92.33**	83.08	84.70	**93.33**
Liver	73.49	72.73	**74.40**	67.68	70.58	**71.56**
Pendigits	87.66	89.01	**96.87**	85.24	95.25	**96.52**
PID	73.07	73.87	**75.92**	74.17	73.34	**76.51**
Satellite Image	75.34	82.67	**83.16**	78.35	70.83	**80.37**
Shuttle	84.11	84.46	**84.89**	80.05	75.05	**82.51**
Sonar	76.82	72.71	**76.85**	63.93	68.08	**75.35**
Thyroid	80.42	87.92	**89.23**	83.06	90.22	**90.32**

Table 10 continued

Data set	Case 2			Case 3		
	LSDF	CSSSFS	SOMVFV	FDROP	KARMA	SOMVFV
Vowel	45.73	47.86	**48.98**	48.14	46.88	**48.75**
Waveform	78.28	76.71	**78.88**	78.03	75.98	**78.79**
Waveform-noise	83.03	81.62	**85.81**	70.62	77.21	**81.17**
Wine	89.28	89.71	**94.09**	92.57	85.15	**94.91**
BA	81.75	79.52	**83.48**	78.10	82.14	**84.21**
TSE	69.11	74.62	**76.72**	71.57	70.47	**72.99**
UKM	75.38	76.43	**80.13**	69.67	74.77	**79.54**
QSAR	74.83	72.99	**76.09**	68.72	70.21	**73.39**
Avg.	78.37	78.42	**82.45**	75.62	76.93	**81.07**

In each case which has compared learning machines, the best performance is given in bold

4.2.5 Experimental Results Derived from Table 10

From this figure, it is found that (1) in terms of the original complete supervised single-view small-scale data sets, our SOMVFV performs best and compared with the classical and state-of-the-art learning machine SVM, SOMVFV gets a higher classification accuracy with at least 7% enhancement; (2) when the complete single-view small-scale data sets are semi-supervised, SOMVFV still performs best and compared with case 1, the decreased classification accuracy is less than 2% in average; (3) if the used data sets miss 10% features, the classification accuracy of SOMVFV, FDROP, and KARMA are both decreased. While compared with FDROP and KARMA, SOMVFV still outperforms them and the decline ratio is smaller; (4) SOMVFV has a good ability to process single-view small-scale data sets when they are semi-supervised and miss features. Simply speaking, compared with the other three cases, when the used data sets are semi-supervised and miss 10% features, the classification accuracy of SOMVFV will be decreased due to the loss of useful discriminant information. While in terms of the classification accuracy, the decline ratio is about $2\% \sim 6\%$ in generally.

4.3 Classification Performance Comparison on Single-View Large-Scale Data Sets with Corresponding Learning Machines

Here, experiments on single-view large-scale data sets which are shown in Table 4 are given. In terms of these 10 single-view large-scale data sets, we adopt the same cases given in Sect. 4.2.

4.3.1 Case 1: Original Data Sets

We adopt OPID, LP, LOL, and OPML for comparisons so that we can validate the effectiveness of the developed SOMVFV on the 10 single-view large-scale data sets. As we said before, all these four compared learning machines can process data sets with missing features, moreover, according to their corresponding references, they can also process complete single-view large-scale data sets. Here, we use case 1 of Table 11 to show the classification performance comparison results on the original 10 supervised single-view large-scale data sets with OPID, LP, LOL, OPML, and SOMVFV used.

Table 11 Classification performance (%) comparisons for single-view large-scale data sets with corresponding learning machines on different cases

Data set	Case 1					Case 2	
	OPID	LP	LOL	OPML	SOMVFV	SSOWMIL	SOMVFV
HIGGS	74.40	64.58	71.44	73.62	**78.12**	72.69	**73.43**
HEPMASS	68.74	61.77	65.85	67.62	**72.91**	64.29	**67.62**
RLCP	79.42	68.04	72.73	74.40	**81.23**	79.36	**80.64**
SUSY	81.21	77.93	79.77	80.66	**83.07**	74.08	**75.11**
GSADGM	73.09	68.81	75.06	75.56	**76.64**	71.07	**74.64**
PAMAP2	63.91	54.78	58.10	60.18	**64.12**	55.97	**60.21**
URL	90.94	82.10	83.81	86.47	**92.31**	86.79	**87.03**
YouTube-C	63.63	58.49	59.58	59.70	**65.32**	60.51	**62.73**
OR	47.83	44.46	45.70	47.55	**51.02**	48.29	**50.38**
Skin	75.67	72.69	75.60	78.41	**79.58**	73.16	**78.29**
Avg.	71.88	65.37	68.76	70.42	**74.43**	68.62	**71.01**

Data set	Case 3					Case 4
	OPID	LP	LOL	OPML	SOMVFV	SOMVFV
HIGGS	71.96	61.40	69.79	70.03	**77.73**	73.23
HEPMASS	65.75	57.25	65.06	65.42	**71.05**	65.71
RLCP	79.28	65.80	71.16	71.84	**80.21**	76.42
SUSY	79.34	73.33	77.34	76.72	**82.78**	70.63
GSADGM	70.44	63.16	72.76	74.06	**74.12**	71.34
PAMAP2	61.29	54.23	56.99	57.90	**62.57**	58.24
URL	86.47	78.80	80.19	82.42	**90.29**	85.58
YouTube-C	**63.57**	53.72	57.59	59.48	63.33	58.66
OR	47.74	42.38	44.36	46.80	**50.90**	49.30
Skin	74.60	66.27	71.58	76.66	**78.69**	74.25
Avg.	70.04	61.63	66.68	68.13	**73.17**	68.34

In each case which has compared learning machines, the best performance is given in bold

4.3.2 Case 2: Semi-supervised Data Sets

Like what we have done in Sect. 4.2.2, if the used 10 single-view large-scale data sets are semi-supervised, then we adopt SSOWMIL for comparison. Please see case 2 of Table 11.

4.3.3 Case 3: Original Data Sets with Missing Features

If these original 10 single-view large-scale data sets miss some features, for example, 10% features, we still use OPID, LP, LOL, OPML, and SOMVFV for experiments so as to see whether these learning machines can process supervised single-view large-scale data sets with missing features or not. Case 3 of Table 11 shows the results.

4.3.4 Case 4: Semi-supervised Data Sets with Missing Features

If the used 10 single-view large-scale data sets are semi-supervised and miss 10% features, since other compared learning machines in Table 9 cannot process this case, so here we only use our proposed SOMVFV for experiments and just to validate that SOMVFV can process this case. Case 4 of Table 11 shows the results.

4.3.5 Experimental Results Derived from Table 11

From this figure, it is found that (1) similarly with what we have gotten in Sect. 4.2.5, our proposed SOMVFV can process single-view large-scale data sets no matter they are complete, features missing, supervised, or semi-supervised; (2) compared with OPID, LP, LOL, and OPML, when we use these learning machines to process data sets with missing features, the classification accuracy of SOMVFV has least influence. In other words, in terms of classification accuracy, the decline ratio of SOMVFV is least; (3) when we process semi-supervised single-view large-scale data sets with missing features, the decline ratio of classification accuracy for SOMVFV is about $3-7\%$ in generally.

4.4 Classification Performance Comparison on Multi-view Small-Scale Data Sets with Corresponding Learning Machines

We also adopt multi-view small-scale data sets Mfeat, Reuters, and Corel [1] for experiments. The following cases are considered.

4.4.1 Case 1: Original Data Sets

For these three complete supervised multi-view small-scale data sets, we use MV-LDA, MV-CCA, MV-LPP, MVL-IV, MDIA-CNN, and MEMR for comparisons. Here, according to [28], MVL-IV can not only process supervised data sets with missing views but also the complete supervised multi-view small-scale data sets. Then case 1 of Table 12 shows the experimental results.

4.4.2 Case 2: Semi-supervised Data Sets

If these three data sets are semi-supervised, then we use MvSs-Zhu, MVML, co-graph, co-features, and AMVS for comparisons. Please see case 2 of Table 12.

4.4.3 Case 3: Original Data Sets with Missing Features

Suppose with the temporary failure of sensor or the man-made faults, in terms of these three data sets, some instances loss 10% features, then due to in Table 9, there is no other compared learning machine can process this case, so we only use our proposed SOMVFV for experiments and validate that SOMVFV has an ability to process this case. Case 3 of Table 12 shows the related experiments. Indeed, for the most below cases in this subsection, we encounter the same situation, i.e., we only use SOMVFV for experiments.

4.4.4 Case 4: Semi-supervised Data Sets with Missing Features

If the used three multi-view small-scale data sets are semi-supervised and miss 10% features, then case 4 of Table 12 shows that SOMVFV can process this case.

4.4.5 Case 5: Original Data Sets with Missing Views

If these three multi-view small-scale data sets miss 20% views, then we use MVL-IV for comparison. Case 5 of Table 12 shows the related experiments.

4.4.6 Case 6: Semi-supervised Data Sets with Missing Views

In this case, we use SOMVFV to process the semi-supervised Mfeat, Reuters, and Corel with missing 20% views and case 6 of Table 12 shows the related experiments.

4.4.7 Case 7: Original Data Sets with Missing Features and Missing Views

Here, we use MDA and SOMVFV to process such a case, i.e., supervised Mfeat, Reuters, and Corel with missing 10% features and 20% views. Case 7 of Table 12 gives the experimental results.

4.4.8 Case 8: Semi-supervised Data Sets with Missing Features and Missing Views

When Mfeat, Reuters, and Corel are semi-supervised and miss 10% features and 20% views, we use SOMVFV to process such a case and case 8 of Table 12 shows the related experimental results.

4.4.9 Experimental Results Derived from Table 12

From this figure, it is found that (1) our proposed SOMVFV can process multi-view small-scale data sets with multiple variable cases and SOMVFV performs best compared with MV-LDA, MV-CCA, MV-LPP, MVL-IV, MDIA-CNN, MEMR, MvSs-Zhu, MVML, co-graph, co-features, AMVS, MDA; (2) compared with MVL-IV, when these three multi-view small-scale data sets miss 20% views, the classification accuracy of SOMVFV decreases fewer; (3) although the performance of SOMVFV is weakened when Mfeat, Reuters, and Corel miss 10% features or 20% views, the decline ratio of classification accuracy is acceptable and about $1\% \sim 7\%$ in generally; (4) if semi-supervised data sets miss features and views simultaneous, the performance of SOMVFV is worst compared with other cases including missing features, miss views, or supervised. According to the experimental results, in case 8, the performance of SOMVFV is weakened with 7.23%, 3.48%, and 9.24% for Mfeat, Reuters, and Corel respectively compared with the performance of SOMVFV in case 1.

4.5 Classification Performance Comparison on Multi-view Large-Scale Data Sets with Corresponding Learning Machines

Furthermore, we adopt multi-view large-scale data sets Video and News [37] to show the effectiveness of our developed SOMVFV. Although in Table 9, we show that OPMV is a

Table 12 Classification performance (%) comparisons for multi-view small-scale data sets with corresponding learning machines on different cases

Data set	Case 1						
	MV-LDA	MV-CCA	MV-LPP	MVL-IV	MDIA-CNN	MEMR	SOMVFV
Mfeat	72.90	81.00	76.67	83.93	73.54	79.86	**84.18**
Reuters	81.40	79.18	84.40	81.57	85.25	89.76	**92.64**
Corel	77.15	79.28	78.95	80.10	78.29	78.25	**81.97**
Avg.	77.15	79.82	80.01	81.87	79.02	82.62	**86.26**

Data set	Case 2						Case 3
	MvSs-Zhu	MVML	Co-graph	Co-features	AMVS	SOMVFV	SOMVFV
Mfeat	71.13	77.50	76.32	77.12	75.41	**78.69**	83.51
Reuters	89.99	89.44	82.28	87.22	83.89	**91.31**	91.01
Corel	77.49	79.78	81.29	73.83	80.77	**81.50**	79.17
Avg.	79.54	82.24	79.96	79.39	80.02	**83.83**	84.56

Data set	Case 4	Case 5		Case 6	Case 7		Case 8
	SOMVFV	MVL-IV	SOMVFV	SOMVFV	MDA	SOMVFV	SOMVFV
Mfeat	78.43	80.13	**83.37**	76.93	**83.73**	82.52	76.95
Reuters	87.60	74.97	**90.22**	88.34	81.76	**89.88**	89.16
Corel	77.48	75.82	**80.35**	78.84	72.54	**77.07**	72.73
Avg.	81.17	76.97	**84.64**	81.37	79.35	**83.15**	79.61

In each case which has compared learning machines, the best performance is given in bold

Table 13 Classification performance (%) comparisons for multi-view large-scale data sets with corresponding learning machines on different cases

Data set	Case 1			Case 2	Case 3	Case 4
	OPMV	SSOPMV	SOMVFV	SOMVFV	SOMVFV	SOMVFV
Video	78.85	85.00	**92.28**	86.43	92.06	86.12
News	86.46	89.50	**96.67**	90.31	94.37	89.63
Avg.	82.66	87.25	**94.47**	88.37	93.21	87.87

In each case which has compared learning machines, the best performance is given in bold

supervised multi-view one-pass learning machine, thus it has no ability to process semi-supervised data sets. While for the fairly comparison, we still use OPMV for experiments. But during the procedure of OPMV, all unlabeled training instances are used for test and they won't take part in the training process. In other words, when we adopt OPMV for experiments, only labeled training instances are used for training.

4.5.1 Case 1: Original Semi-supervised Data Sets

For the original semi-supervised multi-view large-scale data sets Video and News, we use OPMV, SSOPMV, and SOMVFV for experiments. Case 1 of Table 13 shows the related experimental results.

4.5.2 Case 2: Original Semi-supervised Data Sets with Missing Features

Since other compared learning machines in Table 9 cannot process semi-supervised multi-view large-scale data sets with missing features or views, thus for this case and the following two cases, we only use SOMVFV for experiments. Case 2 of Table 13 shows the performance of SOMVFV when semi-supervised Video and News miss 10% features.

4.5.3 Case 3: Original Semi-supervised Data Sets with Missing Views

Case 3 of Table 13 shows the performance of SOMVFV when semi-supervised Video and News miss 20% views.

4.5.4 Case 4: Original Semi-supervised Data Sets with Missing Features and Missing Views

Case 4 of Table 13 shows the performance of SOMVFV when semi-supervised Video and News miss 10% features and 20% views.

4.5.5 Experimental Results Derived from Table 13

From this figure, it is found that (1) SOMVFV performs best and it has a good ability to process semi-supervised multi-view large-scale data sets; (2) if semi-supervised multi-view large-scale data sets Video and News miss 10% features, the performance of SOMVFV is weakened with a larger percentage compared with the case that Video and News miss 20%

views. In other words, missing features has a greater influence compared with missing views in terms of classification accuracy of SOMVFV. This conclusion will be validated by the following experiments given in 5.3; (3) when Video and News miss 10% features and 20% views simultaneous, the performance of SOMVFV is similar with the one in case 2 and this also indicates that missing features has a greater influence.

5 Further Discussion

5.1 Significance Analysis

In this part, we give the significance analysis so as to validate that our SOMVFV is significant better than other compared learning machines. The used analysis method is Friedman-Nemenyi statistical test which is described in [93]. Friedman test is used to analyze if the differences between all compared learning machines on multiple data sets are significant or not while Nemenyi one is used to analyze if the differences between two compared learning machines on multiple data sets are significant or not. In generally, the differences always indicate the ones in the average classification accuracies. Details of Friedman-Nemenyi statistical test can be found in [93] and here, we only give the results in a simple way.

In order to carry out the Friedman-Nemenyi statistical test, according to classification accuracy, we give the ranks of learning machines on different single-view data sets (for convenience, we only adopt the single-view data sets for description). Please see Fig. 3 and in this figure, the last row 'Avg' indicates the average ranks of a learning machine on all used data sets. What's more, in this figure, we also give some indexes related to Friedman-Nemenyi statistical test. In order to show the average ranks clearly, we use Table 14 to show the results. Here, in the figure and table, case 1 indicates results about single-view small-scale while case 2 indicates results about single-view large-scale.

According to this figure, (1) for the case 1, since $F_F = 12.5921 > F_{0.05}(6, 126) = 1.6404$ and $F_F = 12.5921 > F_{0.10}(6, 126) = 1.4694$, thus we reject the null-hypothesis [85], i.e., the differences between SVM, FDROP, KARMA, CNN, LSDF, CSSSFS, and SOMVFV on multiple data sets are significant. Then since $CD_{0.05} = 1.9208$, $CD_{0.10} = 1.7541$,

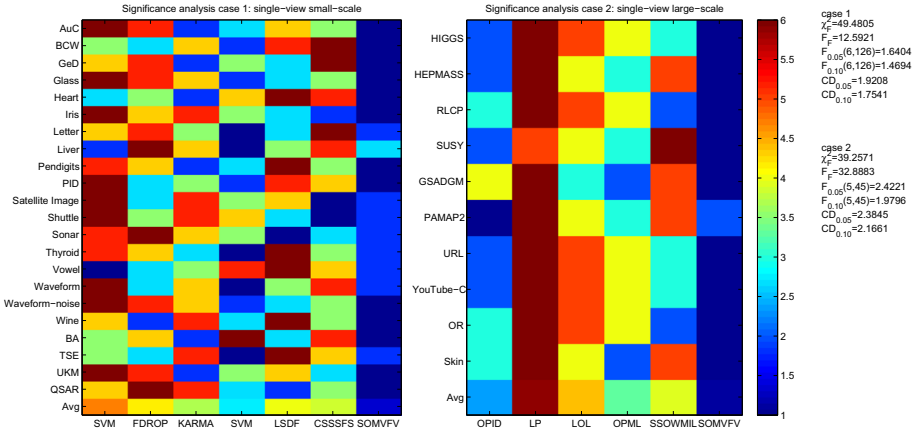


Fig. 3 Rank comparisons on single-view data sets

Table 14 Average ranks of learning machines on different single-view data sets

Case 1	SVM	FDROP	KARMA	CNN	LSDF	CSSSFS	SOMVFV
Avg	5.41	4.77	4.23	3.09	4.59	4.45	1.45
Case 2	OPID	LP	LOL	OPML	SSOWMIL		SOMVFV
Avg	2.40	5.90	4.40	3.30	3.90		1.10

$3.09 < 1.45 + 1.9208 < 4.23$, and $3.09 < 1.45 + 1.7541 < 4.23$, thus compared with SVM, FDROP, KARMA, LSDF, and CSSSFS, the differences between one of them and SOMVFV on multiple data sets are significant while compared with CNN, the difference between CNN and SOMVFV on multiple data sets is not significant; (2) for the case 2, since $F_F = 32.8883 > F_{0.05}(5, 45) = 2.4221$ and $F_F = 32.8883 > F_{0.10}(5, 45) = 1.9796$, thus we also reject the null-hypothesis [85], i.e., the differences between OPID, LP, LOL, OPML, SSOWMIL, and SOMVFV on multiple data sets are significant. Then since $CD_{0.05} = 2.3845$, $CD_{0.10} = 2.1661$, $3.30 < 1.10+2.3845 < 3.90$, and $2.40 < 1.10+2.1661 < 3.30$, thus the differences between LP (LOL, SSOWMIL) and SOMVFV on multiple data sets are significant while the ones between OPID (OPML) and SOMVFV on multiple data sets are not significant at some cases.

According to the above experimental results, we can see that on most cases, our SOMVFV is significant better than other compared learning machines in average.

5.2 Influence of Parameter

As we said before, in SOMVFV, there are many parameters should be adjusted. For example, regularization parameters λs (from λ_1 to λ_9), balance parameters αs (from α_1 to α_4) and μs (from μ_1 to μ_4). As we know, different parameters always bring different classification accuracies. Thus here, we discuss the influence of parameters. Since according to the experimental results which are not shown in this manuscript, we know the different values between λs (αs, μs) bring less influence on classification accuracies, thus here, we suppose different λs (αs, μs) are set the same value. Furthermore, for each kind of data sets, we select one data set for the description of the experimental results. Figure 4 shows the influence of parameters for SOMVFV and for convenience, we only give the experimental results when $\mu = 2^{-2}$, $\mu = 2^0$, and $\mu = 2^2$ since for other settings of μ, we can get the similar results.

According to this figure, it is found that the influence of α is more heavy. It may be caused by the fact that α is employed to ensemble the learning machines and it is more direct to have influence on the final result. This reason and phenomenon are same as the ones given in [42].

5.3 Influence of Rates of Missing Features or Missing Views

In our above experiments, we suppose the data sets miss 10% features or 20% views. Indeed, as we know, more missing features or views will decrease the classification accuracies of the learning machines with a larger decline ratio. Thus here, we discuss the influence of rates of missing features or missing views. For convenience, for the case of missing features and views, we select data set Video and our proposed SOMVFV for experiments; for the case of missing features only, we select data set AuC and learning machines SOMVFV, FDROP,

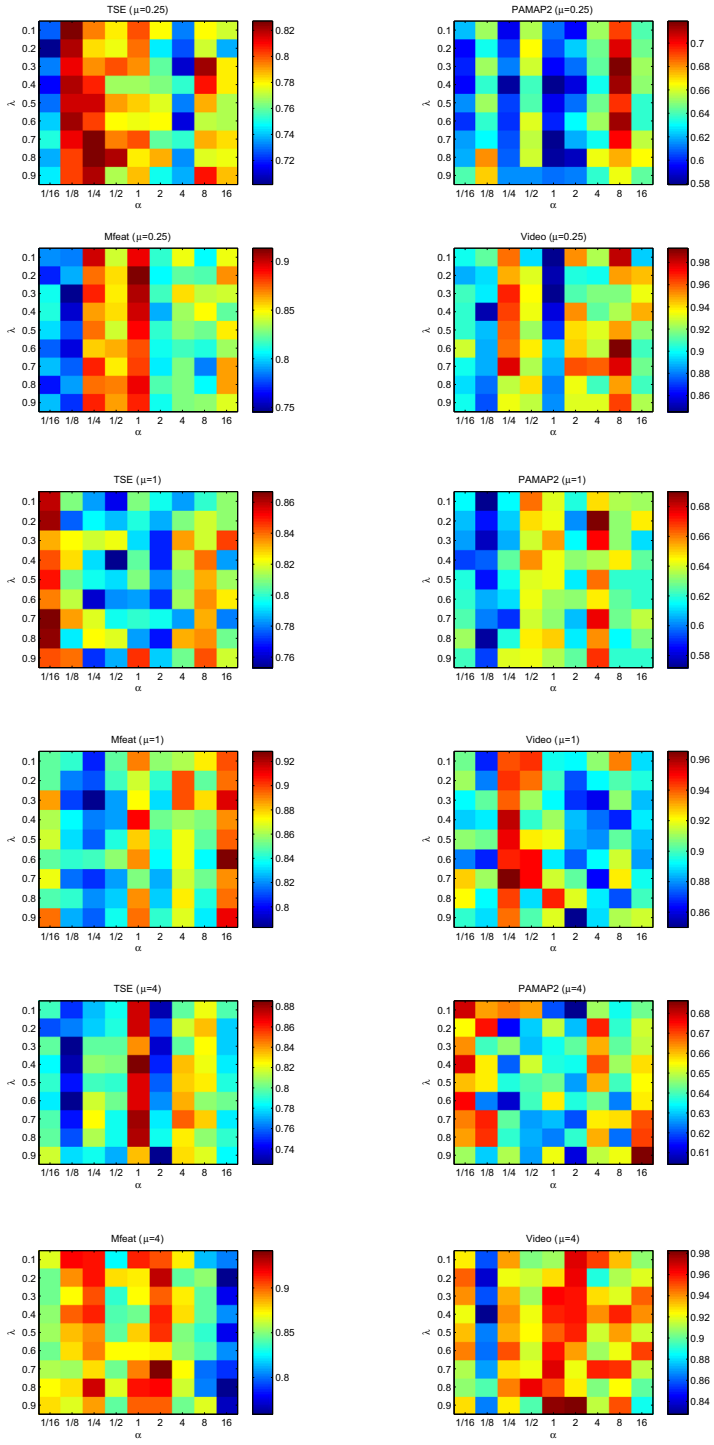


Fig. 4 Classification accuracy comparison with different parameters for SOMVFV on four classical data sets

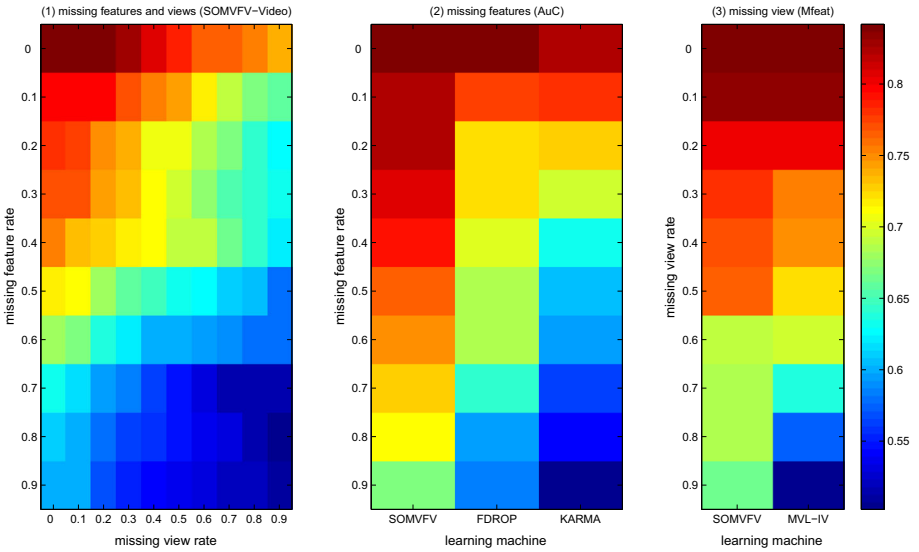


Fig. 5 Classification accuracy comparison with different rates of missing features or missing views

and KARMA for experiments; for the case of missing views only, we select data set Mfeat and learning machines SOMVFV and MVL-IV for experiments.

Figure 5 shows the related experimental results and according to this figure, it is found that (1) compared with missing views, missing features has a greater influence on the classification accuracies of the learning machines. In other words, if the rate of missing views is fixed, a higher rate of missing features makes the classification accuracy of a learning machine decrease with a faster speed. While if the rate of missing features is fixed, the classification accuracy of a learning machine decreases with a slow speed when the rate of missing views is higher; (2) when the rate of missing features or views is higher, the classification accuracy of a learning machine is worse.

5.4 Computational Time Analysis

Here, we show the computational time of the proposed and compared learning machines. Tables 15, 16, 17, and 18 show the results on different kinds of data sets and in these tables, for each data set, the computational time of the first compared learning machine is set to be 1.00 just for convenience.

According to these tables, it is found that (1) for small-scale data sets, our proposed SOMVFV costs less computational time compared with most learning machines. Indeed, different from with compared learning machines, our SOMVFV updates and optimizes the weights of classifiers with going through the data only once and without storing the entire data set. This operation can reduce the computational time at a great extent; (2) for large-scale data sets, especially the multi-view ones, our proposed SOMVFV only costs a litter more or less computational time. Thus according to the experimental results, we can say that our proposed SOMVFV brings a better classification performance and won't lead to a more computational time.

Table 15 Computational time analysis for single-view small-scale data sets

Data set	SVM	FDROP	KARMA	CNN	LSDF	CSSSFS	SOMVFV
AuC	1.00	0.78	0.39	1.22	0.74	0.65	0.69
BCW	1.00	0.77	0.42	1.17	0.86	0.84	0.68
GeD	1.00	0.79	0.37	1.19	1.01	0.73	0.79
Glass	1.00	0.82	0.41	1.19	0.98	0.85	0.71
Heart	1.00	0.84	0.41	1.17	1.03	0.77	0.64
Iris	1.00	0.76	0.38	1.21	0.82	0.90	0.65
Letter	1.00	0.79	0.43	1.19	0.96	0.85	0.87
Liver	1.00	0.80	0.42	1.14	0.92	0.77	0.83
Pendigits	1.00	0.82	0.39	1.14	0.90	0.96	0.61
PID	1.00	0.82	0.42	1.18	0.79	0.90	0.71
Satellite image	1.00	0.76	0.41	1.23	0.98	0.80	0.76
Shuttle	1.00	0.84	0.42	1.19	0.90	0.79	0.81
Sonar	1.00	0.84	0.40	1.20	0.99	0.75	0.76
Thyroid	1.00	0.81	0.41	1.29	0.83	0.83	0.74
Vowel	1.00	0.84	0.40	1.17	0.91	0.62	0.91
Waveform	1.00	0.87	0.41	1.25	0.79	0.77	0.70
Waveform-noise	1.00	0.80	0.38	1.20	1.00	0.96	0.77
Wine	1.00	0.84	0.41	1.27	0.78	1.01	0.68
BA	1.00	0.81	0.39	1.16	0.78	0.69	0.58
TSE	1.00	0.76	0.39	1.18	0.84	0.77	0.73
UKM	1.00	0.82	0.37	1.22	1.01	0.77	0.61
QSAR	1.00	0.76	0.41	1.15	0.82	0.81	0.71
Avg.	1.00	0.81	0.40	1.19	0.89	0.81	0.73

Table 16 Computational time analysis for single-view large-scale data sets

Data set	OPID	LP	LOL	OPML	SSOWMIL	SOMVFV
HIGGS	1.00	1.38	1.28	0.89	1.10	1.00
HEPMASS	1.00	1.21	1.26	0.91	1.02	0.95
RLCP	1.00	1.15	1.26	0.93	1.08	1.03
SUSY	1.00	1.34	1.23	0.93	0.94	0.94
GSADGM	1.00	1.19	1.18	0.97	1.03	1.00
PAMAP2	1.00	1.17	1.31	0.91	1.08	0.99
URL	1.00	1.35	1.22	0.86	1.06	0.95
YouTube-C	1.00	1.31	1.22	0.87	1.01	0.95
OR	1.00	1.35	1.21	0.89	1.03	1.06
Skin	1.00	1.10	1.28	0.91	0.98	0.98
Avg.	1.00	1.26	1.25	0.91	1.03	0.98

Table 17 Computational time analysis for multi-view small-scale data sets

Data set	MV-LDA	MV-CCA	MV-LPP	MVL-IV	MvSs-Zhu	MVML	Co-graph
Mfeat	1.00	0.80	0.79	0.75	0.78	0.75	0.81
Reuters	1.00	0.78	0.76	0.81	0.81	0.76	0.80
Corel	1.00	0.81	0.78	0.81	0.79	0.76	0.84
Avg.	1.00	0.80	0.78	0.79	0.79	0.76	0.82

Data set	Co-features	AMVS	MDIA-CNN	MEMR	MDA	SOMVFV
Mfeat	0.82	0.80	1.19	0.85	0.75	0.63
Reuters	0.78	0.75	1.12	0.75	0.76	0.73
Corel	0.75	0.82	1.18	0.83	0.83	0.63
Avg.	0.79	0.79	1.16	0.81	0.78	0.66

Table 18 Computational time analysis for multi-view large-scale data sets

Data set	OPMV	SSOPMV	SOMVFV
Video	1.00	0.94	1.04
News	1.00	1.11	1.05
Avg.	1.00	1.03	1.05

6 Conclusions and Future Work

In this manuscript, we develop a semi-supervised one-pass multi-view learning with variable features and views (SOMVFV). The developed SOMVFV aims to compress important information of vanished features into functions of survived features and expand to include the augmented features when an or some instances arrive. Then SOMVFV can adopt the information of features and views in the previous time periods to train and test instances at the present time period. In order to validate the effectiveness of the developed SOMVFV, we use 22 small-scale single-view data sets, 10 single-view large-scale data sets, 3 multi-view small-scale data sets, and 2 multi-view large-scale data sets for experiments. Then we further discuss significance analysis, influence of parameter, influence of rates of missing features or missing views, computational time. According to the experiments, it is found that (1) our developed SOMVFV can process multiple kinds of data sets no matter they are small-scale, large-scale, single-view, multi-view, supervised, semi-supervised, complete, and feature or views missing; (2) SOMVFV is significant better than other compared learning machines in average; (3) compared with λ and μ, the influence of α is more heavy; (4) compared with missing views, missing features has a greater influence on the classification accuracies of the learning machines.

Although SOMVFV has a good performance compared with the classical semi-supervised learning machines, one-pass learning machines, and ones with variable features or views, but we still have some issues should be considered in the future work. First, as we know, the higher the rate of missing features or views is, the worse the classification accuracy of a learning machine is. So in the future work, we will try to search a good method to fix or restore more missing features or views. Second, the model of SOMVFV is still complicated, thus we want to simplify it in the future work.

7 Appendix

7.1 Further Discussion About the Objective Function of SOMVFV

Here, we will give the further discussion about Eq. (1) which is the objective function of SOMVFV.

According to OPID [42], OPMV [36], and SSOPMV [37], we know $f(x) = \omega x^T$ can be regarded as a classifier and this classifier is used to predict the label of instance x where ω is the weight of this classifier.

Then according to the background of our work and what we have said in Sect. 3, we should process a large-scale multi-view data set with m views and n instances. What's more, since information of this data will be changed with the elapse of time, thus suppose at the current time period, we have collected instances from G time periods. Then at g-th time period, i-th labeled instance of j-th view for the data set, i.e., x_{ji}^g can be represented as $x_{ji}^g = [x_{ji}^{(v)-g}, x_{ji}^{(s)-g}, x_{ji}^{(a)-g}]$ where $x_{ji}^{(v)-g}$, $x_{ji}^{(s)-g}$, and $x_{ji}^{(a)-g}$ indicate the vanished features, survived features, and augmented features of x_{ji}^g respectively. y_i^g is the real label of i-th instance for the data set at g-th time period if this instance is a labeled one. After that, we recombine x_{ji}^g into three parts, i.e., $\tilde{x}_{ji}^g = [x_{ji}^{(v)-g}, x_{ji}^{(s)-g}]$, $\dot{x}_{ji}^g = [x_{ji}^{(s)-g}]$, and $\bar{x}_{ji}^g = [x_{ji}^{(s)-g}, x_{ji}^{(a)-g}]$ and for these three parts, we define three corresponding classifiers so as to get the label of each part. The corresponding weights of these classifiers are $\tilde{\omega}_j^g$, $\dot{\omega}_j^g$, and $\bar{\omega}_j^g$ respectively. Similarly, for an unlabeled instance $x_{ji'}^g$, the meaning of these corresponding terms are similar.

Then according to the above contents, in Eq. (1), we can see for a labeled instance x_{ji}^g, $\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T}$, $\dot{\omega}_j^g \dot{x}_{ji}^{g^T}$, $\bar{\omega}_j^g \bar{x}_{ji}^{g^T}$ indicate the predicted labels of $\tilde{x}_{ji}^g$, $\dot{x}_{ji}^g$, $\bar{x}_{ji}^g$. So $(\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T} - y_i^g)^2$, $(\dot{\omega}_j^g \dot{x}_{ji}^{g^T} - y_i^g)^2$, $(\bar{\omega}_j^g \bar{x}_{ji}^{g^T} - y_i^g)^2$ mean the differences between their predicted labels and the real labels in terms of $\tilde{x}_{ji}^g$, $\dot{x}_{ji}^g$, $\bar{x}_{ji}^g$. Then $(\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T} - \dot{\omega}_j^g \dot{x}_{ji}^{g^T})^2$, $(\dot{\omega}_j^g \dot{x}_{ji}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji}^{g^T})^2$, $(\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji}^{g^T})^2$ represent the differences between predicted labels of $\tilde{x}_{ji}^g$, $\dot{x}_{ji}^g$, $\bar{x}_{ji}^g$. What's more, for an unlabeled instance $x_{ji'}^g$, $\tilde{\omega}_j^g \tilde{x}_{ji'}^{g^T}$, $\dot{\omega}_j^g \dot{x}_{ji'}^{g^T}$, and $\bar{\omega}_j^g \bar{x}_{ji'}^{g^T}$ indicate the predicted labels of the three recombined parts of $x_{ji'}^g$ and $(\tilde{\omega}_j^g \tilde{x}_{ji'}^{g^T} - \dot{\omega}_j^g \dot{x}_{ji'}^{g^T})^2$, $(\dot{\omega}_j^g \dot{x}_{ji'}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji'}^{g^T})^2$, $(\tilde{\omega}_j^g \tilde{x}_{ji'}^{g^T} - \bar{\omega}_j^g \bar{x}_{ji'}^{g^T})^2$ represent the differences between each other for these predicted labels. Moreover, $\left|\left|\tilde{\omega}_j^g\right|\right|_2^2$, $\left|\left|\dot{\omega}_j^g\right|\right|_2^2$, $\left|\left|\bar{\omega}_j^g\right|\right|_2^2$ indicate the complexities of the corresponding classifiers and for the labeled training instances, the constraint conditions $sign(\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T}) = sign(\dot{\omega}_j^g \dot{x}_{ji}^{g^T}) = sign(\bar{\omega}_j^g \bar{x}_{ji}^{g^T})$ should be satisfied due to for these instances, their predicted labels should be same with different recombined parts adopted.

According to the above definitions and explanations, the minimization of the objective function, i.e., Eq. (1) implies that when we adopt instances to train classifiers, we should try to realize the following aims. First, for the labeled instances, we should make the differences between the predicted labels and the real labels be smaller so that the prediction errors of instances can be smaller no matter which recombined part is adopted. Second, for the training instances including the labeled ones and unlabeled ones, we should make the differences between predicted labels of these recombined parts be smaller so that the prediction results

are similar with different recombined parts adopted. Third, in terms of different recombined parts, we should make the complexities of corresponding classifiers be smaller as far as possible. In terms of this objective function, in an ideal case, each labeled instance can be classified correctly under different recombined parts cases and each recombined part should bring a same predicted label for each instance. What's more, in such an ideal case, complexity of each corresponding classifier is zero, then the value of objective function is zero under ideal case.

7.2 How Can Eq. (1) Achieve Classification

As we said before, $f(x) = \omega x^T$ can be regarded as a classifier which predicts the label of instance x and ω is the weight of this classifier. Then according to the further discussion about Eq. (1), we know that during the procedure of training, we can adopt $(\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T} - y_i^g)^2$, $(\tilde{\omega}_j^g \tilde{x}_{ji}^{g^T} - \dot{\omega}_j^g \dot{x}_{ji}^{g^T})^2$, etc. to achieve classification for training instances. What's more, after we optimize the Eq. (1), we can get the initial weights $\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, and $\bar{\omega}_j^{G+1}$ for the corresponding classifiers at $(G+1)$-th time period. Then we can use instances collected at $(G+1)$-th time period to optimize the values of $\tilde{\omega}_j^{G+1}$, $\dot{\omega}_j^{G+1}$, and $\bar{\omega}_j^{G+1}$. Finally, for the test instance x_i^{G+2} which is collected at $(G+2)$-th time period, we can use $sign(\sum_{j=1}^{m} \bar{\omega}_j^{G+1} \bar{z}_{ji}^{G+2^T})$ to achieve classification and get the predict label of x_i^{G+2} where $\bar{z}_{ji}^{G+2^T}$ is a new representation of x_i^{G+2}.

7.3 Why We Adopt Linear Model in SOMVFV

From our work and Eq. (1), we can see that we only adopt linear model (i.e., $f(x) = \omega x^T$) in SOMVFV. Indeed, the reasons why we adopt linear model are based on the following three points.

First, in many references [36,37,42], adopting linear models for elaboration is more convenient and simpler than adopting nonlinear ones.

Second, adopting nonlinear models can process more complex nonlinear separable problems. But adopting nonlinear models for elaboration should always to adopt kernel functions and this brings some inconvenience and verbose contents. Limited by the length of this paper and for the beautification of typesetting, adopting linear model for elaboration here is feasible.

Third, in OPID [42] which is a basic of our SOMVFV, it also adopts linear model for elaboration, thus in our work, we adopt the same mode and also adopt linear model.

7.4 Why the Adopted Linear Model Can Produce Significant Performance Improvement

As we know, in our real-world applications, most classification tasks should process nonlinear separable problems and nonlinear models always outperform linear ones. But according to the experimental results, it is found that our proposed SOMVFV which adopts linear models can also produce significant performance improvement. Indeed, different from the compared learning machines, according to the framework of our SOMVFV, it can process multiple kinds of data sets no matter they are small-scale, large-scale, single-view, multi-view, supervised, semi-supervised, complete, and feature or views missing. Thus, this makes our SOMVFV get a significant performance improvement even though we adopt linear model.

Acknowledgements This work is supported by National Natural Science Foundation of China under Grant numbers 61602296 and 41701523, Natural Science Foundation of Shanghai under Grant number 16ZR1414500 and authors would like to thank their supports.

References

1. Xu YM, Wang CD, Lai JH (2016) Weighted multi-view clustering with feature selection. Pattern Recognit 53:25–35
2. McCallum A, Nigam K, Rennie J, Seymore K (2000) Automating the construction of internet portals with machine learning. Inf Retr 3(2):127–163
3. Bisson G, Grimal C (2012) Co-clustering of multi-view datasets: a parallelizable approach. In: Proceedings of the 12th IEEE international conference on data mining (ICDM'12), pp 828–833
4. Hussain S, Grimal C, Bisson G (2010) An improved co-similarity measure for document clustering. In: Proceedings of 9th international conference on machine learning and applications (ICMLA' 10), pp 190–197
5. Amini M, Usunier N, Goutte C (2009) Learning from multiple partially observed views—an application to multilingual text categorization. Adv Neural Inf Process Syst 22:28–36
6. Hardoon DR, Szedmak S, Shawe-Taylor J (2004) Canonical correlation analysis: an overview with application to learning methods. Neural Comput 16(12):2639–2664
7. Sharma A, Kumar A, Daume H, Jacobs DW (2012) Generalized multiview analysis: a discriminative latent space. In: IEEE Conference on Computer Vision and Pattern Recognition, vol 157, pp 2160–2167
8. Gönen M, Alpaydin E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
9. Ye G, Liu D, Jhuo IH, Huan J (2012) Robust late fusion with rank minimization. In: Computer vision and pattern recognition, pp 3021–3028
10. Fang YX, Zhang HJ, Ye YM, Li XT (2014) Detecting hot topics from twitter: a multiview approach. J Inf Sci 40(5):578–593
11. Zhang HJ, Liu G, Chow TWS, Liu WY (2011) Textual and visual content-based anti-phishing: a Bayesian approach. IEEE Trans Neural Netw 22(10):1532–1546
12. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Eleventh conference on computational learning theory, pp 92–100
13. Wang W, Zhou ZH (2010) Multi-view active learning in the non-realizable case. Adv Neural Inf Process Syst 23:2388–2396
14. Zhou ZH, Li M (2007) Semi-supervised learning with very few labeled training examples. In: Proceeding of the 22nd AAAI conference on artificial intelligence, pp 675–680
15. Zhao P, Jiang Y, Zhou ZH (2017) Multi-view matrix completion for clustering with side information. In: Advances in knowledge discovery and data mining, pp 403–415
16. Ye HJ, Zhan DC, Miao Y, Jiang Y, Zhou ZH (2015) Rank consistency based multi-view learning: a privacy-preserving approach. In: ACM international on conference on information and knowledge management, pp 991–1000
17. Tzortzis G, Likas A (2012) Kernel-based weighted multi-view clustering. In: 2012 IEEE 12th international conference on data mining, pp 675–684
18. Sun SL, Zhang QJ (2011) Multiple-view multiple-learner semi-supervised learning. Neural Process Lett 34:229–240
19. Deng MQ, Wang C, Chen QF (2016) Human gait recognition based on deterministic learning through multiple views fusion. Pattern Recognit Lett 78(C):56–63
20. Tang JJ, Li DW, Tian YJ, Liu DL (2018) Multi-view learning based on nonparallel support vector machine. Knowl-Based Syst 158:94–108
21. Wu F, Jing XY, You XG, Yue D, Hu RM, Yang JY (2016) Multi-view low-rank dictionary learning for image classification. Pattern Recognit 50:143–154
22. Zhu SH, Sun X, Jin DL (2016) Multi-view semi-supervised learning for image classification. Neurocomputing 208:136–142
23. Sun S, Xie X X, Dong C (2018) Multiview learning with generalized eigenvalue proximal support vector machines. IEEE Trans Cybern PP(99):1–10
24. Sun ZR, Cai YX, Wang SJ, Wang CD, Zheng YQ, Chen YH, Chen YC (2018) Multi-view intact space learning for tinnitus classification in resting state EEG. Neural Process Lett. https://doi.org/10.1007/s11063-018-9845-1
25. Li JX, Zhang B, Lu GM, Zhang D (2019) Generative multi-view and multi-feature learning for classification. Inf Fusion 45:215–226

26. Zhao Y, You XG, Yu SJ, Yu C, Yuan W, Jing XY, Zhang TP, Tao DC (2018) Multi-view manifold learning with locality alignment. Pattern Recognit 78:154–166
27. Sindhwani V, Rosenberg DS (2008) An RKHS for multi-view learning and manifold co-regularization. In: International conference on machine learning, ACM, pp 976–983
28. Xu C, Tao DC, Xu C (2015) Multi-view learning with incomplete views. IEEE Trans Image Process 24(12):5812–5825
29. Wang HY, Wang X, Zheng J, Deller JR, Peng HY, Zhu LQ, Chen WG, Li XL, Liu RJ, Bao HJ (2014) Video object matching across multiple non-overlapping camera views based on multi-feature fusion and incremental learning. Pattern Recognit 47(12):3841–3851
30. Sheikhpour R, Sarram MA, Gharaghani S, Chahooki MAZ (2017) A survey on semi-supervised feature selection methods. Pattern Recognit 64:141–158
31. Tao H, Hou CP, Nie FP, Zhu JB, Yi DY (2017) Scalable multi-view semi-supervised classification via adaptive regression. IEEE Trans Image Process 26(9):4283–4296
32. Xu XX, Li W, Xu D, Tsang IW (2016) Co-labeling for multi-view weakly labeled learning. IEEE Trans Pattern Anal Mach Intell 38(6):1113–1125
33. Han C, Chen J, Wu QY, Mu S, Min HQ (2015) Sparse markov chain-based semi-supervised multi-instance multi-label method for protein function prediction. J Bioinform Comput Biol 13(5):1543001
34. Zhang CH, Zheng WS (2017) Semi-supervised multi-view discrete hashing for fast image search. IEEE Trans Image Process 26(6):2604–2617
35. Liu CL, Hsaio WH, Lee CH, Chang TH, Kuo TH (2016) Semi-supervised text classification with universum learning. IEEE Trans Cybern 46(2):462–473
36. Zhu Y, Gao W, Zhou ZH (2015) One-pass multi-view learning. J Mach Learn Res 30:1–16
37. Zhu CM, Wang Z, Zhou RG, Wei L, Zhang XF, Ding Y (2018) Semi-supervised one-pass multi-view learning. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3654-3
38. Globerson A, Roweis S (2006) Nightmare at test time: robust learning by feature deletion. In: International conference on machine learning, pp 353–360
39. Dekel O, Shamir O (2008) Learning to classify with missing and corrupted features. In: International conference on machine learning, pp 216–223
40. Teo CH, Globerson A, Roweis ST, Smola AJ (2007) Convex learning with invariances. In: Conference on neural information processing systems, pp 1489–1496
41. Hazan E, Livni R, Mansour Y (2015) Classification with low rank and missing data. In: International conference on machine learning, pp 257–266
42. Hou CP, Zhou ZH (2018) One-pass learning with incremental and decremental features. IEEE Trans Pattern Anal Mach Intell 40:2776–2792. https://doi.org/10.1109/TPAMI.2017.2769047
43. Cheng H, Deng W, Fu C, Wang Y, Qin Z (2011) Graph-based semi-supervised feature selection with application to automatic spam image identification. In: Proceedings of the computer science for environmental engineering and ecoinformatics, pp 259–264
44. Zhao J, Lu K, He X (2008) Locality sensitive semi-supervised feature selection. Neurocomputing 71:1842–1849
45. Doquire G, Verleysen M (2011) Graph Laplacian for semi-supervised feature selection in regression problems. In: Cabestany J, Rojas I, Joya G (eds) Advances in computational intelligence. IWANN 2011. Lecture notes in computer science. Springer, Berlin, pp 248–255
46. Doquire G, Verleysen M (2013) A graph Laplacian based approach to semi-supervised feature selection for regression problems. Neurocomputing 121:5–13
47. Chen LCL, Huang RHR, Huang WHW (2010) Graph-based semi-supervised weighted band selection for classification of hyperspectral data. In: Proceedings of the international conference on audio, language and image processing, pp 1123–1126
48. Yang M, Chen Y, Ji G (2010) Semi_fisher score: a semi-supervised method for feature selection. In: Proceedings of the international conference on machine learning and cybernetics, pp 527–532
49. Lv S, Jiang H, Zhao L, Wang D, Fan M (2013) Manifold based fisher method for semisupervised feature selection. In: Proceedings of the 10th international conference on fuzzy systems and knowledge discovery, pp 664–668
50. Yang W, Hou C, Wu Y (2011) A semi-supervised method for feature selection. In: Proceedings of the international conference on computer and information science and technology, pp 329–332
51. Liu Y, Nie F, Wu J, Chen L (2013) Efficient semi-supervised feature selection with noise insensitive trace ratio criterion. Neurocomputing 105:12–18
52. Liu Y, Nie F, Wu J, Chen L (2010) Semi-supervised feature selection based on label propagation and subset selection. In: Proceedings of the international conference on computer and information application, pp 293–296

53. Kalakech M, Biela P, Macaire L, Hamad D (2011) Constraint scores for semi-supervised feature selection: a comparative study. Pattern Recognit Lett 32:656–665
54. Benabdeslem K, Hindawi M (2011) Constrained Laplacian score for semi-supervised feature selection. In: Proceedings of the machine learning and knowledge discovery in databases, pp 204–218
55. Zhao Z, Liu H (2007) Semi-supervised feature selection via spectral analysis. In: Proceedings of the 7th SIAM international conference data mining, pp 641–646
56. Song X, Zhang J, Han Y, Jiang J (2016) Semi-supervised feature selection via hierarchical regression for web image classification. Multimed Syst 22(1):41–49
57. Ma Z, Nie F, Yang Y, Uijlings JRR, Sebe N, Hauptmann AG (2012) Discriminating joint feature analysis for multimedia data understanding. IEEE Trans Multimed 14(6):1662–1672
58. Shi C, Ruan Q, An G (2014) Sparse feature selection based on graph Laplacian for web image annotation. Image Vis Comput 32:189–201
59. Ma Z, Yang Y, Nie F, Uijlings J, Sebe N (2011) Exploiting the entire feature space with sparsity for automatic image annotation. In: Proceedings of the 19th ACM multimedia conference, pp 283–292
60. Barkia H, Elghazel H, Aussem A (2011) Semi-supervised feature importance evaluation with ensemble learning. In: Proceedings of the international conference on data mining, pp 31–40
61. Ren J, Qiu Z, Fan W, Cheng H, Yu PS, Philip SY (2008) Forward semi-supervised feature selection. In: Proceedings of the advances in knowledge discovery and data mining, pp 970–976
62. Bellal F, Elghazel H, Aussem A (2012) A semi-supervised feature ranking method with ensemble learning. Pattern Recognit Lett 33:1426–1433
63. Han Y, Park K, Lee YK (2011) Confident wrapper-type semi-supervised feature selection using an ensemble classifier. In: Proceedings of the 2011 2nd artificial intelligence, management science and electronic commerce, pp 4581–4586
64. Ang JC, B HH, Nuzly H, Hamed A, Haron H, Hamed HNA (2015) Semi-supervised SVM-based feature felection for cancer classification using microarray gene expression data. In: International conference on industrial, vol 9101, pp 468–477
65. Yang L, Wang L (2007) Simultaneous feature selection and classification via semisupervised models. In: Proceedings of the third international conference on natural computation, pp 646–650
66. Dai K, Yu HY, Li Q (2013) A semisupervised feature selection with support vector machine. J Appl Math 2013(1):1–11
67. Xu Z, King I, Lyu MRT, Jin R (2010) Discriminative semi-supervised feature selection via manifold regularization. IEEE Trans Neural Netw 21(7):1033–47
68. Han Y, Yang Y, Yan Y, Ma Z, Sebe N, Member S (2015) Semisupervised feature selection via spline regression for video semantic recognition. IEEE Trans Neural Netw Learn Syst 26:252–264
69. Zhang J, Yu J, Tao DC (2018) Local deep-feature alignment for unsupervised dimension reduction. IEEE Trans Image Process 27(5):2420–2432
70. Zhang YX, Pal S, Coates M, $\ddot{U}$stebay D (2019) Bayesian graph convolutional neural networks for semi-supervised classification. In: The thirty-third AAAI conference on artificial intelligence (AAAI-19), pp 1–8
71. Yu J, Rui Y, Tao DC (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032
72. Yu J, Tao DC, Wang M, Rui Y (2015) Learning to rank using user clicks and visual features for image retrieval. IEEE Trans Cybern 45(4):767–779
73. Hong CQ, Yu J, Wan J, Tao DC, Wang M (2015) Multimodal deep autoencoder for human pose recovery. IEEE Trans Image Process 24(12):5659–5670
74. Hong CQ, Yu J, Tao DC, Wang M (2015) Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. IEEE Trans Ind Electron 62(6):3742–3751
75. Hong CQ, Yu J, You J, Chen XH, Tao DC (2015) Multi-view ensemble manifold regularization for 3D object recognition. Inf Sci 320:395–405
76. Xiao Y, Chen J, Wang YC, Cao ZG, Zhou JTY, Bai X (2019) Action recognition for depth video using multi-view dynamic images. Inf Sci 480:287–304
77. Ting FF, Tan YJ, Sim KS (2019) Convolutional neural network improvement for breast cancer classification. Expert Syst Appl 120:103–115
78. Zhou ZZ, Zheng WS, Hu JF, Xu Y, You J (2016) One-pass online learning: a local approach. Pattern Recognit 51:346–357
79. Li WB, Gao Y, Wang L, Zhou LP, Huo J, Shi YH (2018) OPML: a one-pass closed-form solution for online metric learning. Pattern Recognit 75:302–314
80. Junsawang P, Phimoltares S, Lursinsap C (2016) A fast learning method for streaming and randomly ordered multi-class data chunks by using one-pass-throw-away class-wise learning concept. Expert Syst Appl 63:249–266

81. Blake CL, Newman DJ, Hettich S, Merz CJ (2012) UCI repository of machine learning databases. http://archive.ics.uci.edu/ml/index.php
82. Asuncion A, Newman D (2007) UCI machine learning repository. http://archive.ics.uci.edu/ml/datasets/Multiple+Features
83. Amini MR, Usunier N, Goutte C (2009) Learning from multiple partially observed views-an application to multilingual text categorization. In: Neural information processing systems (NIPS), pp 28–36
84. http://multilingreuters.iit.nrc.ca/ReutersMultiLingualMultiView.htm
85. Vapnik V (1998) Statistical learning theory. Wiley, Hoboken
86. Iosifidis A, Tefas A A, Nikolaidis N, Pitas I (2012) Multi-view human movement recognition based on fuzzy distances and linear discriminant analysis. Comput Vis Image Underst 116(3):347–360
87. Rupnik J, Shawe-Taylor J (2010) Multi-view canonical correlation analysis. In: Proceeding of Slovenian KDD conference on data mining data warehouses, pp 1–4
88. Yin X, Huang Q, Chen X (2011) Multiple view locality preserving projections with pairwise constraints. Commun Syst Inf Technol 100:859–866
89. Wang ZH, Yoon S, Xie SJ, Lu Y, Park DS (2016) Visual tracking with semi-supervised online weighted multiple instance learning. Vis Comput 32(3):307–320
90. Du YT, Li Q, Cai ZM, Guan XH (2013) Multi-view semi-supervised web image classification via co-graph. Neurocomputing 122:430–440
91. Gu P, Zhu QS, Zhang C (2009) A multi-view approach to semi-supervised document classification with incremental Naive Bayes. Comput Math Appl 57(6):1030–1036
92. Yang ZK, Liu Z, Liu SY, Min L, Meng WT (2014) Adaptive multi-view selection for semi-supervised emotionrecognition of posts in online student community. Neurocomputing 144:138–150
93. Zhu CM (2016) Improved multi-kernel classification machine with Nyström approximation technique and Universum data. Neurocomputing 175:610–634