Multi-view and Multi-label Method with Three-Way Decision-Based Clustering

Changming Zhu[1(✉)], Lin Ma[1], Panhong Wang[1], and Duoqian Miao[2]

[1] Shanghai Maritime University, Shanghai 201306, China
cmzhu@shmtu.edu.cn
[2] Tongji University, Shanghai 201804, China

Abstract. In real-world multi-view multi-label clustering and some classification tasks, instances and the corresponding clusters has at least three kinds of relationships, belong-to definitely, not belong-to definitely, and uncertain. Some learning machines consider only two of them, for example, belong-to definitely and not belong-to definitely. Moreover, three-way decision-based clustering (TDC) strategy is a good method to make the belongingness of instances to a cluster depend on the probabilities of uncertain instances belonging to core regions. Thus in our work, we take the notion of classical multi-view multi-label learning machines as the basic and introduce TDC so as to develop a multi-view and multi-label method with three-way decision-based clustering (MVML-TDC) and consider the relationships between instances and clusters. Experimental results validate that MVML-TDC achieves a better average performance and an acceptable running time.

Keywords: Three-way decision-based clustering · Multi-view learning · Multi-label learning

1 Introduction

1.1 Background and Problems

Since multi-view multi-label data sets exist in real-world applications widely, thus some related tasks are put forward. Among them, two tasks are of a general nature. One is clustering and the other is classification. Classical multi-view multi-label clustering methods include connectivity constrained clustering algorithm for traffic segmentation [1] and classical multi-view multi-label classification methods include latent semantic aware multi-view multi-label learning (LSA-MML) [2], latent multi-view subspace clustering (LMSC) [3]. Although there are many learning methods developed to process these tasks, there is a key problem should to be solved. As we know, in real-world applications, instances and the clusters (or sub classes) might have gradual relationships. Namely, there are three relationships between an instance and a cluster, namely, belong-to definitely, not belong-to definitely, and uncertain. In most of the existing studies,

© Springer Nature Switzerland AG 2020
Y. Peng et al. (Eds.): PRCV 2020, LNCS 12306, pp. 69–80, 2020.
https://doi.org/10.1007/978-3-030-60639-8_6

a cluster is represented by a single set. Any set naturally divides the space into two regions. Instances belong to the cluster if they are elements of the set, otherwise they do not. Here, only two relationships are considered, no matter in hard clustering or in soft clustering. They are typically based on two-way (i.e., binary) decisions. Then for the third relationship, which means the instance may or may not belong to the cluster, one cannot make decisions based on the presently obtained knowledge, i.e, two-way decisions, but they can make further decisions once more information becomes available. This method is referred to as three-way decisions. According to the present learning methods, they don't adopt three-way decisions. Thus, the improvement of performance is limited.

1.2 Objectives

In order to solve the above this problem, we develop a multi-view and multi-label method with three-way decision-based clustering (MVML-TDC). First, we adopt an three-way decision-based clustering (TDC) which is one kind of three-way decisions developed by Yu et al. [4] to improve clustering performance and produce the group partition of the data set. Second, we construct a multi-view and multi-label model with the introduction of TDC.

1.3 Novelty and Contributions

The novelty of MVML-TDC is that in the field of multi-view multi-label learning, it is the first attempt for the combination of multi-view multi-label learning and three-way decisions. The proposed method can improve the classification and clustering performances simultaneously.

The contributions of MVML-TDC are (1) it has a better ability to process multi-view multi-label data sets; (2) it won't add too much running time and moves forward research of multi-view multi-label learning.

2 Three-Way Decision-Based Clustering [4]

Suppose there is a data set $X = \{x_1, ..., x_i, ..., x_n\}$ and it should be clustered with g clusters, namely, $C = \{C_1, ..., C_m, ..., C_g\}$ and for each cluster C_m, it can be represented by $C_m = (Co(C_m), Fr(C_m))$ further. Here, $Co(C_m) = CoreRegion(C_m)$ and $x \in CoreRegion(C_m)$ represents that x belongs to cluster C_m definitely. Meanwhile, $Fr(C_m) = FringeRegion(C_m)$ and $x \in FringeRegion(C_m)$ represents that x might belong to cluster C_m. Then, $Tr(C_m) = X - Co(C_m) - Fr(C_m) = TrivialRegion(C_m)$ and $x \in TrivialRegion(C_m)$ represents that x does not belong to cluster C_m definitely. Moreover, for each cluster, we have $X = Co(C_m) \bigcup Fr(C_m) \bigcup Tr(C_m)$, $Co(C_m) \bigcap Fr(C_m) = \varnothing$, $Co(C_m) \bigcap Tr(C_m) = \varnothing$, $Tr(C_m) \bigcap Fr(C_m) = \varnothing$. Thus, $C = \{(Co(C_1), Fr(C_1)), ..., (Co(C_g), Fr(C_g))\}$. What's more, we let $Y = \{y_1, ..., y_i, ..., y_n\}$ indicates the label matrix and y_i represents the label of x_i. Then TDC is carried out as a sequence of the following steps.

2.1 Pairwise Constraints

According to [4], pairwise constraints offer typical prior information for semi-supervised clustering. In their work, they introduce must-link (positive association) and cannot-link (negative association) to reflect the constraint relations between the data points, i.e., instances. Must-link constraint requires that the two instances must belong to the same cluster, and this relation is denoted by $ML = \{(x_i, x_j)|y_i = y_j, \quad for \quad i \neq j, \quad x_i, x_j \in X, \quad y_i, y_j \in Y\}$. Cannot-link constraint requires that the two objects must belong to different clusters, and this relation is denoted by $CL = \{(x_p, x_q)|y_p = y_q, \quad for \quad p \neq q, \quad x_p, x_q \in X, \quad y_p, y_q \in Y\}$. Then for instances $x_i, x_j, x_g \in X$, [5] said that must-link constraint shows the following transitivity properties for instances.

$$\begin{aligned} &(x_i, x_j) \in ML \quad \& \quad (x_j, x_g) \in ML \\ &\Rightarrow (x_i, x_g) \in ML \\ &(x_i, x_j) \in ML \quad \& \quad (x_j, x_g) \in CL \\ &\Rightarrow (x_i, x_g) \in CL \end{aligned} \tag{1}$$

Then in [4], we set a matrix $R \in \mathbb{R}^{n \times n}$ to store the constraint pairs and the update of R is given below. First, R is initialized as $\varnothing$. Then, when a pair of instances are must-link constraint relation, namely $(x_i, x_j) \in ML$, the corresponding value of element in R is updated to 1; when a pair of instances are cannot-link constraint relation, namely $(x_i, x_j) \in CL$, the corresponding value of element in R is updated to 0. At the end of each iteration, we update R according to the response of expert and the transitivity properties of Eq. (1).

Next, one updates the consensus similarity matrix $W^\star$ in the following manner. Here, $W^\star = (Z^\star + (Z^\star)^T)/2$ where $Z^\star$ is a consensus low-rank matrix which is derived from X. The method to get $Z^\star$ can be refer to [4].

$$\begin{aligned} &if \quad (x_i, x_j) \in ML, \quad then \quad w_{ij} = w_{ji} = 1 \\ &if \quad (x_i, x_j) \in CL, \quad then \quad w_{ij} = w_{ji} = 0 \end{aligned} \tag{2}$$

2.2 Initialize Core Regions Construction

When we obtain the pairwise constraints of X, they should initialize core regions construction. The method to realize the initialization is farthest-first traversal scheme and this scheme aims to select the core instances. Core instances indicate the ones which locate on the fringe of a cluster and contain more information than instances in the center of a cluster. The basic idea of farthest-first traversal of a set of instances is to find K instances such that they are far from each other. Details are given below.

First, we let the original X be the $CandidateSet$ and l be the count of the number of constructed core regions. Initial value of l is 0. Second, for each cluster

$(Co(C_m), Fr(C_m))$, it is initialized as $\varnothing$, namely, $Co(C_m) = \varnothing$ and $Fr(C_m) = \varnothing$. Third, we first select a starting instance x from X at random and put x into the $Co(C_1)$. Fourth, we choose the next instance to be farthest from the untraversed set $CandidateSet$ by Eq. (3). Being specific, $AllCo$ is the set of all core instances, namely, $AllCo = \bigcup_{p=1}^{l} Co(C_p)$. According to min-max criterion, the distance between x and $AllCo$ is $d(x, AllCo(p)) = \min\limits_{y \in AllCo(p)} ||x - y||$. Then, the farthest one is determined as follows.

$$\begin{aligned} x &\leftarrow arg \max_{x \in CandidateSet} d(x, AllCo) \\ &= arg \max_{x \in CandidateSet, y \in AllCo(p)} (\min ||x - y||) \end{aligned} \tag{3}$$

After that, we should decide whether x and an instance $x_i \in Co(C_p)$ $(1 \leq p \leq l)$ are in the same cluster. They make pair-wise queries through the form as: do instances x and x_i belong to the same cluster? If the ML constraint is satisfied, x is assigned to $Co(C_m)$ and it should be removed from $CandidateSet$. If no one ML constraint is satisfied after traversing all core regions, a new core region $Co(C_{l+1})$ is constructed and assign x to the new core region $Co(C_{l+1})$. With the above procedure, we can divide the $CandidateSet$ into g clusters.

2.3 Extend Core Regions and Construct Fringe Regions

Once we initialize the core regions construction, they should extend these core regions and construct fringe regions. Concretely speaking, let $N(x)$ be a set of q neighbor instances of x. Then we extend the core regions by observing the relationship between the x (which is an unlabeled instance) and x_i (which is a labeled random instance from $Co(C_m)$) with the following three-way decision rules. Namely, if x is the neighbor of x_i and x_i is also the neighbor of x, then we can say that x is much similar with x_i and they should both belong to the core region. If x is the neighbor of x_i, but x_i is not the neighbor of x, we say they are not similar and x should belong to the fringe region. Otherwise, if x is not the neighbor of x_i, x should belong to the $Tr(C_m)$.

$$\begin{aligned} &if \quad (x \in N(x_i)) \wedge (x_i \in N(x)), \\ &\quad then \quad Co(C_m) = Co(C_m) \cup \{x\} \\ &if \quad (x \in N(x_i)) \wedge (x_i \notin N(x)), \\ &\quad then \quad Fr(C_m) = Fr(C_m) \cup \{x\} \\ &if \quad (x \notin N(x_i)), \\ &\quad then \quad Tr(C_m) = Tr(C_m) \cup \{x\} \end{aligned} \tag{4}$$

2.4 Select the Most Informative Instance $x^\star$ from Fringe Regions

Then we adopt the active learning strategy to improve the performance of clustering. The objective of this strategy is to select the most informative instance $x^\star$

from fringe regions. Concretely speaking, we measure the uncertainty of instance based on the similarity firstly. Namely, on the base of consensus similarity matrix $W^\star$, let $w_{.j}$ denote the similarity between x and x_j and $U = \bigcup_{m=1}^{g} Fr(C_m)$ denote all uncertain instances currently, then they adopt the following formula to estimate the probability of an uncertain instance x belonging to a core region $Co(C_m)$ where $|Co(C_m)|$ is the number of instances in the core region $Co(C_m)$.

$$P(x \in Co(C_m)) = \frac{\frac{1}{|Co(C_m)|}\sum_{x_j \in Co(C_m)} w_{.j}}{\sum_{s=1}^{g} \frac{1}{|Co(C_s)|}\sum_{x_j \in Co(C_s)} w_{.j}} \tag{5}$$

Second, we use Eq. (6) to measure the uncertainty of an instance by the entropy.

$$H(x) = -\frac{1}{g}\sum_{t=1}^{g}(P(x \in Co(C_m))log_2 P(x \in Co(C_m))) \tag{6}$$

where $x \in U$.

Then, the most informative instance $x^\star$ is selected by Eq. (7).

$$x^\star = arg\max_{x \in U} H(x) \tag{7}$$

2.5 Construct Pairwise Query

Finally, we construct pairwise query with the following method. First, they sort the clusters by $P(x^\star \in Co(C_m))$ in descending order where $1 \leq m \leq g$. Second, for each cluster, they select one instance x_i from $Co(C_m)$ in random and query the constraint relationship between $x^\star$ and x_i. If $(x^\star, x_i) \in ML$, then $Co(C_m) = Co(C_m) \cup \{x^\star\}$. At last, they adopt Eq. (1) to update the matrix R and Eq. (2) to update the matrix $W^\star$.

With the above five steps, we can cluster X into g clusters. Details of TDC can be found in [4].

3 Multi-view and Multi-label Method with Three-Way Decision-Based Clustering

3.1 Data Preparation

Suppose there is a data set $X = \{x_1, ..., x_i, ..., x_n\}$ with v views and $X^1, ..., X^j, ..., X^v$ is a data matrix of each view. Here, $i \in [1, n]$ and $j \in [1, v]$. For jth view, i.e., $X^j \in \mathbb{R}^{d_j \times n}$ where d_j is the feature dimension of jth view, it consists of information from n instances, namely, $X^j = \{x_1^j, ..., x_i^j, ..., x_n^j\}$ and $x_i^j = \{x_{i1}^j, ..., x_{ip}^j, ..., x_{id_j}^j\} \in \mathbb{R}^{d_j \times 1}$ represents the information of jth view for ith instance and x_{ip}^j represents pth feature of ith instance in the jth view where $p \in [1, d_j]$. For ith instance, i.e, $x_i \in \mathbb{R}^{d \times 1}$ can be represented by $x_i = \{x_i^{1T}, ..., x_i^{jT}, ..., x_i^{vT}\}^T$ where $d = \sum_{j=1}^{v} d_j$. Thus X can also be rewritten

as $X = \begin{pmatrix} X^1 \\ \vdots \\ X^j \\ \vdots \\ X^v \end{pmatrix} \in \mathbb{R}^{d\times n}$. Furthermore, X is also a multi-label data set and in different views, an instance always possesses different labels, thus suppose $y_i^j \in \mathbb{R}^{l_j\times 1}$ is a label vector of ith instance in the jth view and each component of y_i^j indicates the label of x_i^j for the corresponding class. l_j represents that at jth view, instances have l_j classes. If the rth component of y_i^j, namely, $y_{ir}^j = 1$, it means x_i^j belongs to rth class definitely. If $y_{ir}^j = -1$, this indicates that x_i^j does not belong to rth class definitely. If $y_{ir}^j = 0$, this means whether x_i^j belongs to rth class or not is not available. Then $y_i = \{{y_i^1}^T, ..., {y_i^j}^T, ..., {y_i^v}^T\}^T$ represents the label of ith instance, $Y^j = \{y_1^j, ..., y_i^j, ..., y_n^j\} \in \mathbb{R}^{l_j\times n}$ represents the label matrix of jth view, and we let $Y = \begin{pmatrix} Y^1 \\ \vdots \\ Y^j \\ \vdots \\ Y^v \end{pmatrix} \in \mathbb{R}^{l\times n}$ indicates the label matrix for X where $l = \sum_{j=1}^{v} l_j$. Furthermore, since in many cases, instances have two kinds of labels, one is predicted labels and the other is real labels. So here, we let Y, Y^j, y_i^j, y_i represent the predicted ones and $\widetilde{Y}$, $\widetilde{Y^j}$, $\widetilde{y_i^j}$, $\widetilde{y_i}$ represent the real ones. Definitions of $\widetilde{\star}$ is similar with the $\star$.

3.2 Framework of MVML-TDC

On the base of TDC, our proposed MVML-TDC is carried out with the following method.

According to the above contents, we know if $\widetilde{Y}$ is low-rank, it can be written as the low-rank decomposition, i.e., $\widetilde{Y} = UV$, where $U \in \mathbb{R}^{l\times k}$, $V \in \mathbb{R}^{k\times n}$, and $rank(\widetilde{Y}) = k < l$. U has a function to project the original labels to the latent label space while V can be treated as the latent labels that are more compact and more semantically abstract than the original labels. For $\widetilde{Y^j}$, it can also be decomposed by $\widetilde{Y^j} = U^j V^j$ where $U^j \in \mathbb{R}^{l\times k_j}$, $V^j \in \mathbb{R}^{k_j\times n}$, and $rank(\widetilde{Y^j}) = k_j < l$. Since in real-world applications, due to the lack of manpower or making equipment failure, labels are only partially observed and some instances maybe lost a few labels in some views, so we always want to minimize the reconstruction error on the observed labels, i.e., $\min_{U,V,U^j,V^j} ||\mathrm{II}_{\Omega}(Y - UV)||_F^2 + \sum_{j=1}^{v} \left|\left|\mathrm{II}_{\Omega^j}(Y^j - U^j V^j)\right|\right|_F^2$. Here, $||\star||_F^2$ represents the square of Frobenius norm for $\star$, Ω (Ω^j) consists of indices of the observed

labels in Y (Y^j), $[||\Pi_\Omega(A)||]_{ij} = A_{ij}$ if $(i,j) \in \Omega$, and 0 otherwise (similar to Ω^j case).

After that, we adopt a linear mapping $W \in \mathbb{R}^{d\times k}$ ($W^j \in \mathbb{R}^{d_j\times k_j}$) to map instances to the latent labels and W (W^j) is learned by $\min_{W,V,W^j,V^j} \left|\left|V - W^T X\right|\right|_F^2 + \sum_{j=1}^{v} \left|\left|V^j - {W^j}^T X^j\right|\right|_F^2$.

Moreover, in order to introduce the local label correlation, we divide the data set into several groups with TDC. Namely, for X, it is partitioned into g groups, i.e., $X = \{X_1, X_2, ..., X_g\}$ and each part $X_m \in \mathbb{R}^{d\times n_m}$ where n_m is the number of instances in X_m. Then under j-th view, X^j is also divided into g^j groups, i.e., $X^j = \{X_1^j, X_2^j, ..., X_{g^j}^j\}$ and m-th group of X^j is $X_m^j \in \mathbb{R}^{d_j\times n_m^j}$. Then since the prediction on instance x_i is $sign(f(x_i))$ where $f(x_i) = UW^T x_i \in \mathbb{R}^{l\times 1}$, so $F_0 = [f(x_1), f(x_2), ..., f(x_n)] = UW^T X$ represents the classifier output matrix of X. Similarly, $F_0^j = U^j {W^j}^T X^j$, $F_m = UW^T X_m$, $F_m^j = U^j {W^j}^T X_m^j$ represent the classifier output matrices of X^j, X_m, X_m^j respectively. The dimensions of F_0, F_0^j, F_m, F_m^j are $l \times n$, $l \times n$, $l \times n_m$, $l \times n_m^j$ respectively.

Then, on the base of X, X_m, X^j, X_m^j and their corresponding observed label matrices, we compute the label correlation matrices. Take X as instance, $S_0 = \{[S_0]_{pq}\}$ denotes global label correlation matrix and $[S_0]_{pq} = \frac{y_{p,:} y_{q,:}^T}{||y_{p,:}||||y_{q,:}||}$ represents the global label correlation of p-th label with respect to q-th label and $y_{p,:}$ is the p-th row of Y. Then we let L_0 be the Laplacian matrix of S_0. Similarly, for X_m, $S_m = \{[S_m]_{pq}\}$ is the corresponding local label correlation matrix and L_m is its Laplacian matrix. Then, under j-th view, for X^j and X_m^j $S_0^j = \{[S_0^j]_{pq}\}$ and $S_m^j = \{[S_m^j]_{pq}\}$ are the corresponding global label correlation matrix and local label correlation matrix, L_0^j and L_m^j are their corresponding Laplacian matrices. Dimensions of S_0, L_0, S_m, L_m, S_0^j, L_0^j, S_m^j, L_m^j are both $l \times l$.

According to the above definitions, we want the classifier outputs can be closer if two labels are more positively correlated and then, we should minimize $tr(F_0^T L_0 F_0) + \sum_{m=1}^{g} tr(F_m^T L_m F_m) + \sum_{j=1}^{v} (tr({F_0^j}^T L_0^j F_0^j) + \sum_{m=1}^{g^j} tr({F_m^j}^T L_m^j F_m^j))$ where $tr(A)$ represents the trace of A.

Furthermore, refer to LSA-MML and LMSC [2,3] which introduce a consensus multi-view representation to encode the complementary information from different views, we adopt the same way. Suppose P is a latent representation matrix (i.e., consensus multi-view representation), B^j is the basic matrix corresponding to j-th view, then $\sum_{j=1}^{v} \left|\left|X^j - B^j P\right|\right|_F^2$ searches a comprehensive multi-view representation and $\sum_{j\neq t} IND(B^j, B^t)$ is used to measure the independence between different views where $IND(B^j, B^t) = -HSIC(B^j, B^t)$ and $HSIC$ is a Hilbert-Schmidt independence criterion estimator [2].

So according to the above contents, our goal is to solve the following optimization problem.

$$\begin{aligned}
&\min_{U,W,V,U^j,W^j,V^j} ||\Pi_\Omega(Y - UV)||_F^2 \qquad (8)\\
&+ \lambda_0 \left|\left|V - W^T X\right|\right|_F^2 + \lambda_1 \Re(U, V, W, U^j, V^j, W^j, P, B^j)\\
&+ \sum_{j=1}^{v} (\lambda_2 \left|\left|\Pi_{\Omega^j}(Y^j - U^j V^j)\right|\right|_F^2 + \lambda_3 \left|\left|V^j - {W^j}^T X^j\right|\right|_F^2)\\
&+ \lambda_4 tr(F_0^T L_0 F_0) + \lambda_5 \sum_{m=1}^{g} tr(F_m^T L_m F_m)\\
&+ \sum_{j=1}^{v} (\lambda_6^j tr({F_0^j}^T L_0^j F_0^j) + \lambda_7^j \sum_{m=1}^{g^j} tr({F_m^j}^T L_m^j F_m^j))\\
&+ \lambda_8 \sum_{j=1}^{v} \left|\left|X^j - B^j P\right|\right|_F^2 + \lambda_9 \sum_{j \neq t} IND(B^j, B^t)
\end{aligned}$$

where λs are tradeoff parameters, λ^js are tradeoff parameters corresponding to j-th views, $\Re(U, V, W, U^j, V^j, W^j, P, B^j) = ||U||_F^2 + ||V||_F^2 + ||| W|_F^2 + \left|\left|U^j\right|\right|_F^2 + \left|\left|V^j\right|\right|_F^2 + \left|\left|W^j\right|\right|_F^2 + ||P||_F^2 + \left|\left|B^j\right|\right|_F^2$ is the regularizer.

3.3 Solution

In order to solve Eq. (8), we adopt alternating optimization. Namely, in each iteration, we update one of the variables in $\{Z_m, U, V, W, Z_m^j, U^j, V^j, W^j, P, B^j\}$ with gradient descent and leave the others fixed. After we get the ∇_A where $A \in \{Z_m, U, V, W, Z_m^j, U^j, V^j, W^j, P, B^j\}$, we can use $A := A - \eta \nabla_A$ to update A where η is the step size. Once we get the optimal results of these parameters, we can use $UW^T X$ to compute the classifier outputs for X. For X^j and the group partitions C_m^j and C_m, the outputs can be produced with the corresponding optimal matrices including U^js, W^js and others.

3.4 Computational Complexity

As in [4], the computational complexity of TDC is $O(ng^2) + O(Frng) + max(O(Fr), O(g))$ where Fr is the number of instances in all fringe regions. Then according to [6], the computational complexity of a classical multi-view multi-label learning machine is $O(Gn^2)$ where G is a constant. Thus, the computational complexity should be $O(ng^2) + O(Frng) + max(O(Fr), O(g)) + O(Gn^2) = O(Sn^2)$ where S is another constant.

4 Experiments

In order to demonstrate the performance of the developed MVML-TDC, we consider some benchmark data sets for experiments and related experimental results are shown in Subsect. 4.2.

4.1 Experimental Setting

Data Set. In our experiments, three kinds of data sets are adopted. The first kind is 3 multi-view data sets including Pascal VOC 2007 (VOC)[1], MIR-Flickr (MIR)[2], and 3Source[3]. The second kind is 2 multi-label data sets including Arts and Business which are also adopted in [7–9]. The third kind is a multi-view multi-label data set, i.e., NUS-WIDE [10,11].

Compared Method. Since three kinds of data sets are adopted in our experiments, thus for the fair comparison, we also adopt three kinds of learning methods for comparisons. They are multi-view learning methods, multi-label ones, and multi-view multi-label ones. For the multi-view ones, multiple-view multiple-learner semi-supervised learning method (MVMLSS) [12], LMSC [3], multi-view low-rank dictionary learning (MLDL) [13] are adopted. For the multi-label ones, label-specific features and local pairwise label correlation based multi-label learning (LF-LPLC) [14], multi-label classification machine with hierarchical embedding (MLCHE) [15], multi-label learning with global and local label correlation (GLOCAL) [7] are adopted. For the multi-view multi-label ones, multi-view based multi-label propagation (MVMLP) [11], semi-supervised dimension reduction for multi-label and multi-view learning (SSDR-MML) [16], LSA-MML [2] are adopted.

Parameter Setting. For the compared methods, the parameter settings of them can be found in the respective references. Then for the proposed MVML-TDC, the setting of TDC can refer to [4]. For others, we can refer to [6]. Although it seems we should adjust too many parameters, but [6] has validated that only ones which corresponds to global and local label correlations have a larger influence on the performance of the learning machine. Thus, we won't show the influence of these parameters and the ones corresponding to global and local label correlations are set to be 10^{-3}.

4.2 Experimental Results

AUC and Precision. We adopt AUC and precision to show the effectiveness of MVML-TDC for the classification tasks. In general, a higher AUC or a higher precision brings a better classification performance. Table 1 and Table 2 give the average AUC and precision respectively for the test sets for each data set. In these tables, •/◦ indicates that MVML-TDC is significantly better/worse than the corresponding method (pairwise t-tests at 95% significance level). The best average AUC or precision for each data set is shown in bold. / represents no result since the related method cannot process the corresponding data set. From these tables, it is found that in most cases, MVML-TDC has a better performance

[1] http://host.robots.ox.ac.uk/pascal/VOC/.
[2] http://press.liacs.nl/mirflickr/.
[3] http://mlg.ucd.ie/datasets/3sources.html.

Table 1. Average AUC (mean ± std.) of MVML-TDC with compared methods for test instances.

Data sets	MVML-TDC	LMSC	MVMLSS	MLDL	LF-LPLC
VOC	**0.729 ± 0.009**	0.687 ± 0.016 •	0.702 ± 0.035 •	0.686 ± 0.035 •	/
MIR	**0.522 ± 0.017**	0.488 ± 0.008 •	0.516 ± 0.036 •	0.520 ± 0.046	/
3Source	**0.731 ± 0.001**	0.711 ± 0.037 •	0.693 ± 0.045 •	0.725 ± 0.014	/
Arts	**0.890 ± 0.007**	/	/	/	0.820 ± 0.005 •
Business	0.958 ± 0.003	/	/	/	0.926 ± 0.003 •
NUS-WIDE	**0.850 ± 0.027**	/	/	/	/
win/tie/loss		3 / 0 / 0	3 / 0 / 0	1 / 2 / 0	2 / 0 / 0
Data sets	MLCHE	GLOCAL	MVMLP	SSDR-MML	LSA-MML
VOC	/	/	0.701 ± 0.003 •	0.626 ± 0.001 •	0.666 ± 0.049 •
MIR	/	/	0.512 ± 0.037 •	0.514 ± 0.025 •	0.475 ± 0.002 •
3Source	/	/	0.647 ± 0.003 •	0.704 ± 0.029 •	0.688 ± 0.023 •
Arts	0.782 ± 0.005 •	0.887 ± 0.005	0.788 ± 0.005 •	0.814 ± 0.005 •	0.771 ± 0.005 •
Business	0.961 ± 0.003	0.950 ± 0.003	0.808 ± 0.003 •	0.839 ± 0.003 •	**0.962 ± 0.003**
NUS-WIDE	/	/	0.822 ± 0.031 •	0.796 ± 0.056 •	0.801 ± 0.056 •
win/tie/loss	1 / 1 / 0	0 / 2 / 0	6 / 0 / 0	6 / 0 / 0	5 / 1 / 0

Table 2. Average precision (mean ± std.) of MVML-TDC with compared methods for test instances.

Data sets	MVML-TDC	LMSC	MVMLSS	MLDL	LF-LPLC
VOC	**0.701 ± 0.004**	0.617 ± 0.006 •	0.648 ± 0.047 •	0.683 ± 0.028 •	/
MIR	**0.521 ± 0.002**	0.471 ± 0.042 •	0.440 ± 0.004 •	0.496 ± 0.004 •	/
3Source	**0.703 ± 0.014**	0.630 ± 0.050 •	0.614 ± 0.034 •	0.655 ± 0.025 •	/
Arts	**0.655 ± 0.008**	/	/	/	0.634 ± 0.005 •
Business	**0.916 ± 0.004**	/	/	/	0.902 ± 0.004 •
NUS-WIDE	**0.871 ± 0.011**	/	/	/	/
win/tie/loss		3 / 0 / 0	3 / 0 / 0	3 / 0 / 0	2 / 0 / 0
Data sets	MLCHE	GLOCAL	MVMLP	SSDR-MML	LSA-MML
VOC	/	/	0.594 ± 0.011 •	0.682 ± 0.018 •	0.637 ± 0.034 •
MIR	/	/	0.462 ± 0.002 •	0.464 ± 0.045 •	0.486 ± 0.006 •
3Source	/	/	0.624 ± 0.005 •	0.692 ± 0.035 •	0.680 ± 0.002 •
Arts	0.580 ± 0.005 •	0.608 ± 0.005 •	0.603 ± 0.005 •	0.627 ± 0.004 •	0.639 ± 0.004 •
Business	0.839 ± 0.004 •	0.803 ± 0.004 •	0.909 ± 0.004	0.905 ± 0.004 •	0.895 ± 0.004 •
NUS-WIDE	/	/	0.782 ± 0.015 •	0.876 ± 0.011	0.853 ± 0.011 •
win/tie/loss	2 / 0 / 0	2 / 0 / 0	5 / 1 / 0	5 / 1 / 0	6 / 0 / 0

and according to the win/tie/loss counts, the proposed MVML-TDC is clearly superior to other compared learning methods, as it wins for most times and less loses.

Running Time. In Sect. 3.4, we said that the computational complexity of MVML-TDC can be reduced to $O(Sn^2)$ where S is a constant. This computational complexity is always smaller than $O(n^3)$ which is the computational complexity of many traditional methods, it still larger than some linear learning methods. Thus, we show the running time of these compared methods and observe the difference. Table 3 shows the related experimental results. From this

Table 3. Running time (in seconds) of MVML-TDC with compared methods.

Data sets	MVML-TDC	LMSC	MVMLSS	MLDL	LF-LPLC
VOC	80.37	73.03	81.61	78.39	/
MIR	377.49	376.59	379.69	377.98	/
3Source	0.23	0.23	0.21	0.23	/
Arts	58.31	/	/	/	53.47
Business	52.72	/	/	/	49.09
NUS-WIDE	42.09	/	/	/	/
Data sets	MLCHE	GLOCAL	MVMLP	SSDR-MML	LSA-MML
VOC	/	/	77.72	80.75	80.43
MIR	/	/	360.02	360.96	368.03
3Source	/	/	0.22	0.22	0.22
Arts	58.13	56.16	54.40	57.75	55.69
Business	49.72	52.49	53.43	53.76	49.30
NUS-WIDE	/	/	35.98	37.12	39.81

table, we find that our proposed method cost a little more running time which is also accepted by us.

5 Conclusions and Future Studies

In real-world applications, multi-view multi-label data sets are widely used and traditional learning methods always produce worse performances when these data sets exhibit complicate topologies. One main reason is that they cannot reveal the uncertain relationship between instances and the corresponding clusters. In this work, we develop a multi-view multi-label learning method with three-way decision-based clustering (MVML-TDC) to overcome such a problem. In MVML-TDC, it makes the belonging of instances to a cluster depend on the probability with three-way decision-based clustering strategy. Experimental results validate that (1) MVML-TDC achieves a better average AUC and precision in statistical; (2) the running time of MVML-TDC won't add too much.

Acknowledgment. This work is supported by 'Chenguang Program' supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission under grant number 18CG54. Furthermore, this work is also sponsored by Project funded by China Postdoctoral Science Foundation under grant number 2019M651576, National Natural Science Foundation of China (CN) under grant number 61602296, Natural Science Foundation of Shanghai (CN) under grant number 16ZR1414500. The authors would like to thank their supports.

References

1. Fu, Y.J., et al.: Service usage analysis in mobile messaging apps: a multi-label multi-view perspective. In: IEEE 16th International Conference on Data Mining (ICDM) (2016). https://doi.org/10.1109/ICDM.2016.0106
2. Zhang, C.Q., Yu, Z.W., Hu, Q.H., Zhu, P.F., Liu, X.W., Wang, X.B.: Latent semantic aware multi-view multi-label classification. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 4414–4421 (2018)
3. Zhang, C.Q., et al.: Generalized latent multi-view subspace clustering. IEEE Trans. Pattern Anal. Mach. Intell. (2018). https://doi.org/10.1109/TPAMI.2018.2877660
4. Yu, H., Wang, X.C., Wang, G.Y., Zeng, X.H.: An active three-way clustering method via low-rank matrices for multi-view data. Inf. Sci. (2018). https://doi.org/10.1016/j.ins.2018.03.009
5. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering. Technical report, Stanford (2002)
6. Zhu, C.M., Miao, D.Q., Wang, Z., Zhou, R.G., Wei, L., Zhang, X.F.: Global and local multi-view multi-label learning. Neurocomputing **371**, 67–77 (2020)
7. Zhu, Y., Kwok, J.T., Zhou, Z.H.: Multi-label learning with global and local label correlation. IEEE Trans. Knowl. Data Eng. **99**, 1–24 (2017)
8. Zhang, J., et al.: Multi-label learning with label-specific features by resolving label correlations. Knowl. Based Syst. **159**, 148–157 (2018)
9. Huang, J., et al.: Improving multi-label classification with missing labels by learning label-specific features. Inf. Sci. **492**, 124–146 (2019)
10. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: NUS-WIDE: a real-world web image database from national university of Singapore. In: Proceedings of the ACM International Conference on Image and Video Retrieval, p. 48 (2009)
11. He, Z.Y., Chen, C., Bu, J.J., Li, P., Cai, D.: Multi-view based multi-label propagation for image annotation. Neurocomputing **168**, 853–860 (2015)
12. Sun, S.L., Zhang, Q.J.: Multiple-view Multiple-learner Semi-supervised Learning. Neural Process. Lett. **34**, 229–240 (2011)
13. Wu, F., Jing, X.Y., You, X.G., Yue, D., Hu, R.M., Yang, J.Y.: Multi-view low-rank dictionary learning for image classification. Pattern Recogn. **50**, 143–154 (2016)
14. Weng, W., Lin, Y.J., Wu, S.X., Li, Y.W., Kang, Y.: Multi-label learning based on label-specific features and local pairwise label correlation. Neurocomputing **273**, 385–394 (2018)
15. Kumar, V., Pujari, A.K., Padmanabhan, V., Sahu, S.K., Kagita, V.R.: Multi-label classification using hierarchical embedding. Expert Syst. Appl. **91**, 263–269 (2018)
16. Qian, B.Y., Wang, X., Ye, J.P., Davidson, I.: A reconstruction error based framework for multi-label and multi-view learning. IEEE Trans. Knowl. Data Eng. **27**(3), 594–607 (2015)