Knowledge-Based Systems 54 (2013) 137–146

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Multi-view classification with cross-view must-link and cannot-link side information

Qiang Qian [a], Songcan Chen [a,*], Xudong Zhou [b]

[a] Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, P.R. China
[b] Information Engineering College, Yangzhou University, Yangzhou 225009, P.R. China

ARTICLE INFO

Article history:
Received 22 October 2012
Received in revised form 1 September 2013
Accepted 5 September 2013
Available online 9 October 2013

Keywords:
Classification
Multi-view learning
Without correspondence
Unpaired multi-view data
Cross-view side information

ABSTRACT

Side information, like must-link (ML) and cannot-link (CL), has been widely used in single-view classification tasks. However, so far such information has never been applied in multi-view classification tasks. In many real world situations, data with multiple representations or views are frequently encountered, and most proposed algorithms for such learning situations require that all the multi-view data should be paired. Yet this requirement is difficult to satisfy in some settings and the multi-view data could be totally unpaired. In this paper, we propose an learning framework to design the multi-view classifiers by only employing the weak side information of *cross-view* must-links (CvML) and *cross-view* cannot-links (CvCL). The CvML and the CvCL generalize the traditional single-view must-link (SvML) and single-view cannot-link (SvCL), and to the best of our knowledge, are first definitely introduced and applied into the multi-view classification situations. Finally, we demonstrate the effectiveness of our method in our experiments.

## 1. Introduction

Traditional learning only involves data with single view. However, in many real world circumstances, data with multiple natural feature representations or views are frequently encountered. For example, web pages can be represented by both its own content and hyperlinks pointing to it. To tackle this data type, multi-view learning has been developed since the pioneer works [7,31]. So far many approaches [7,8,24,20] have been proposed and achieved empirical and theoretical successes. All of those approaches rely on two common assumptions, compatibility and independence between views [7]. However, to make the two assumptions work, one requirement that should be fulfilled is such multi-view data should be paired. Specifically, for the representation of a sample in one view, there is a corresponding representation paired in the other view.

Sometimes this requirement is over-rigorous in some circumstances. For instance, in a wireless sensor network, collected data could be missed or polluted during data transmission due to device malfunction or malicious attacks. Thus only partial data are paired while the rest are unpaired [15]. In [19,15,18,25], some methods have been proposed to deal with this scenario. More extremely, in some circumstances where all data are even unpaired, for example, web pages from English Routers and French Routers are unpaired. We may not easily know which English web page corresponds to which French web page.

This paper focuses on the most difficult totally-unpaired extreme circumstance. Since no pairing information between multiple views exists, we introduce a new type of side information, called cross-view must-link and cross-view cannot-link, to help learning. Must-link and cannot-link side information is usually used in the classification [30,34] and clustering learning [33,28] on single-view data (called SvML and SvCL in this paper). Two samples belonging to the SvML set share the same label, while in the SvCL set possess different labels. Compared with commonly-used supervised labels, such SvMLs and SvCLs are weaker in characterizing supervision information. Virtually, we can infer both the SvML and SvCL between samples if knowing their label information, but cannot reversely. The SvML and the SvCL only provide the label relations between samples within view, thus cannot help the totally-unpaired multi-view learning. To achieve this goal, some cross-view relations are needed. Consequently, in this paper, we introduce the cross-view must-link (CvML) and cross-view cannot-link (CvCL). Two representations in different views in CvML set indicates that their labels are the same, while in CvCL set indicates that their labels are different. Unlike SvML and SvCL, CvML and CvCL build the implicit label relations across different views. As a result, we can transfer mutually the learning information between different views through these CvMLs and CvCLs. Compared with explicit label information, CvML and CvCL are likewise also weaker

* Corresponding author. Tel.: +86 25 84892956.
E-mail addresses: qian.qiang.yx@gmail.com (Q. Qian), s.chen@nuaa.edu.cn (S. Chen), xdzhou@nuaa.edu.cn (X. Zhou).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.09.010

in supervision like SvML and SvCL for the same reason. Moreover, the paired information belongs to the CvML because paired representations must own the same label, but the CvML does not mean paired representations because two representations linked by CvML could come from different samples. Intuitively, by forcing the outputs of the target classification functions in each view to obey the CvML and the CvCL constraints, the outputs learned in one view can be shifted to that in the other view through both of them and aid the classification learning in that view. To the best of our knowledge, such side information has never definitely been introduced and applied in multi-view classifier design.

The proposed framework is based on the classical regularization frameworks [27,2] and the new regularization terms which encode the CvMLs and the CvCLs side information, i.e. forcing the outputs of the representations in the CvMLs to be the same and the outputs of the representations in the CvCLs to be different. Since the true (strongly-supervised) labels are unknown, the classical regularization framework has to be modified by introducing probabilistic indicators which indicate how possible the sample belongs to a given class. Such a modified framework leads to a block-wise convex optimization problem which can be iteratively solved effectively by the classic block coordinate descent method with guarantee of iterative convergence to a stationary point [5]. Our experiments demonstrate its effectiveness as well.

We summarize our contribution as follows:

- We introduce and deepen the concepts of the CvML and the CvCL, which are extensions of the SvML and the SvCL, to aid the joint classification learning in different views.
- We develop a classification learning framework which utilizes the cross-view side information to learn classifiers in the tough unpaired multi-view settings.

The rest of the paper is organized as follows. In Section 2, we review some related work. Then we introduce our framework in Section 3. Next we illustrate our experiment results in Section 4. And finally, we conclude this paper and present the future work in Section 5.

## 2. Related work

Our work is related to both classification learning with the SvML and SvCL side information and multi-view classification. Thus we review the two parts respectively. Since the ML and CL side information has never been used in multi-view classification, we mainly review the related work on the single-view circumstance.

### 2.1. SvMLs and SvCLs for classification

The ML and CL side information in single view has demonstrated its value in classification tasks. Yan et al. [30] formulated both MLs and CLs into a convex pairwise loss function and integrated it into the traditional margin-based learning framework. Thus the proposed framework can handle both the label and MLs/CLs together. Nguyen and Caruana [22] incorporated both MLs and CLs into the margin-based learning framework and proposed PCSVM algorithm. Zhang and Yan [34] first transformed both the ML and the CL pairs of samples into a new space and learned an estimator there, then transformed the estimator back into the original sample space. They proved that the final estimator is sign-insensitively consistent with the optimal decision boundary and gave out its asymptotic variance.

Rather than those directly incorporating both the MLs and the CLs into classification models, metric learning goes along another line. It first learns a Mahalanobis metric which obeys MLs and CLs constraints, then uses the distance-based classifiers like the $k$ nearest neighbor to classify the test data. Typical works include [10,29,13,23]. The ML and CL side information is also used to learn proper kernel matrices for the later kernel machine algorithms. Li et al. [21] forced the entries of kernel matrix corresponding to MLs and CLs to be 1 and 0 respectively and developed a kernel learning algorithm PCP. PCP is computationally intensive because it is solved by Semidefinite Programming (SDP). Hu et al. [17] proposed kernel propagation method to avoid solving SDP on full kernel matrix. The main idea is first to learn a small kernel matrix then propagate it into full kernel matrix.

The SvMLs and the SvCLs are also applied in other tasks like clustering, image segmentation et al. For user's reference we name a few works in typical application domains like image segmentation [33], video surveillance [16], clustering [28,3,32]. Despite so many works on the SvMLs and the SvCLs, the CvMLs and the CvCLs are almost never touched to the best of our knowledge.

### 2.2. Multi-view learning

Multi-view learning is a very natural learning settings. It was first touched in Yarowsky's [31] and Blum et al.'s [7] works. Blum et al. proposed the renowned co-training algorithm. It alternatively trains the predictor in one view and uses the predicted labels to aid the training in another view. Dasgupta et al. [9] proved a theoretical PAC-style generalization bound of the co-training.

Sindhwani et al. [24] introduced the co-regularization algorithm. The co-regularization algorithm directly models the cross-view agreement and incorporates it into a regularization framework. They introduced a family of algorithms with different regularization frameworks (the classical regularization framework and manifold regularization framework). The formulation is a convex optimization problem rather than the style of alternatively learning on each view like the co-training. The formulation is related to our framework to some extent. We will compare with it in Section 3.4.

Since full paired multi-view data are over-rigorous in some applications, some methods were proposed for partially paired circumstance. Kimura et al. [18] considered the situation where additional unpaired data are provided and developed Semi Canonical Correlation Analysis (SemiCCA) algorithm. They used both the paired and unpaired data to regularize CCA through PCA-type penalty. Lampert and Kromer [19] proposed a modified Maximum Covariance Analysis algorithm for weakly-paired multimodal data. They guessed the pairing between data views and optimized it along with dimension reduction parameter matrix. Blaschko et al. [6] modified Kernel Canonical Correlation Analysis with Laplaician regularization by using unpaired data and propose SemiLRKCCA algorithm. However, they only embedded the local structure on constraints but not in its objective and had too many model parameters. The PPLCA algorithm propsed by Gu et al. [15] overcomes the shortcomings of SemiLRKCCA. PPLCA simultaneously embeds the local structure into both objectives and constraints and has less model parameters. Sun et al. [25] developed discriminative canonical correlation analysis in partially paired situations. They proposed DCCAM by estimating the within-class and between-class correlation on both the paired and unpaired data.

## 3. Multi-view learning under cross-view MLs and CLs

In this section, we introduce our framework. Later, we compare our framework with the co-regularization framework [24] at last.

Our training process can formally be conducted in two steps: in the 1st step, we employ the available CvMLs and CvCLs to design a (sign) classifier which is used to decide whether any given two

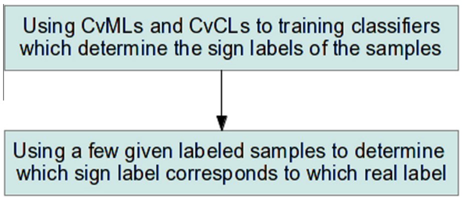

Using CvMLs and CvCLs to training classifiers which determine the sign labels of the samples
Using a few given labeled samples to determine which sign label corresponds to which real label

**Fig. 1.** Formal two-step training process.

samples are from the same class or not. In fact, this step does not involve any label, thus the classifier is not used to decide a real class of a sample but just returns the sign label of either +1 or −1 to indicate whether the sample shares the same real class or not with some training sample. In order to determine its real label finally, we need formally the 2nd step in which we adopt a few labeled samples provided to determine which real label the previous sign label corresponds to. Fig. 1 shows the two steps.

### 3.1. Learning classifiers determining sign labels

Our framework is based on the classical regularization framework [27,2] for supervised learning. It solves the following optimization problem:

$$\min_{f\in\mathcal{H}_k}\frac{1}{2}\sum_{i=1}^{N}V(x_i,c_i,f)+\lambda\|f\|^2_{\mathcal{H}_k} \tag{1}$$

where $\mathcal{H}_k$ is an Reproducing Kernel Hilbert Space (RKHS) induced by a kernel $k$, $c_i$ is the label of sample $x_i$, and $V$ is a loss function, such as the squared loss or the hinge loss.

In our settings, only the CvMLs and CvCLs side information are at hand and the sample labels are unavailable. Consequently, we are not certain of which classes the samples belong to. The brought uncertainty is handled by introducing the probabilistic indicators just like fuzzy c-means clustering algorithm [11], and Eq. (1) is modified as follows

$$\min_{f\in\mathcal{H}_k,u}\frac{1}{2}\sum_{i=1}^{N}\sum_{r\in\{+,-\}}u_{ir}^2V(x_i,c_r,f)+\lambda\|f\|^2_{\mathcal{H}_k} \tag{2}$$

where $u_i$ for each $x_i$ is the probabilistic indicator vector which subjects to non-negativity and unitary summation constraints. +, − are the positive and negative labels. We let $c_+ = 1$ and $c_- = -1$ respectively. Usually in order to avoid hard assignment of labels, the exponents of $u_{ir}$ are set to 2 rather than 1 [11].

Eq. (2) looks like the fuzzy c-means (FCM) formulation, however essentially there is a major difference: in FCM the clustering performs in sample space, thus the clustering centers are a set of vector prototypes to be optimized and have the same dimensionality as the sample space, while our algorithm performs in (label) output space with different dimensionality from the sample space, and the centers are fixed to −1 and 1.

The CvMLs and the CvCLs are used to regularize the learning in both views. The underlying principle is to force the classifiers' outputs in separate views to obey both the CvMLs and the CvCLs constraints. For CvMLs, we simply use the square of difference of the corresponding outputs. It is a convex formulation as follows:

$$\sum_{(i,j)\in\mathcal{M}}(f_x(x_i)-f_y(y_j))^2 \tag{3}$$

where $\mathcal{M}$ denotes the CvML set. It implies that large output differences would incur large penalties. But for the CvCLs, it is not so easy to formulate [14,26]. Here we employ Goldberg et al.'s [14] method and formulate the CvCLs into a convex penalty:

$$\sum_{(i,j)\in\mathcal{C}}(f_x(x_i)+f_y(y_j))^2 \tag{4}$$

where $\mathcal{C}$ denotes the CvCL set. Note that the minus in Eq. (3) is substituted for the plus in Eq. (4). The penalty is zero if $f_x(x_i)$ and $f_y(y_j)$ have the same absolute value but opposite signs, thus minimizing the penalty implies different output labels. The trivial case $f_x(x_i) = f_y(y_j) = 0$ is avoided because it will raise classification error.

This idea could also be applied in multi-view clustering and dimension reduction tasks if we can properly penalize the outputs which violate the CvMLs and the CvCLs constraints. For example, if the outputs of cluster algorithm are multinomial random variables, we can consider penalize large (small) Kullback–Leibler divergences of the representations in $\mathcal{M}(\mathcal{C})$.

Integrating them together suggests the following optimization problem:

$$\begin{aligned}\min_{f_x,f_y,u^x,u^y} J = {} & \frac{1}{2}\sum_{i=1}^{N_x}\sum_{r\in\{+,-\}}{u_{ir}^x}^2V(x_i,c_k,f_x)+\frac{\lambda_1}{2}\|f_x\|^2_{\mathcal{H}_{\|_\S}}\\ & +\frac{1}{2}\sum_{j=1}^{N_y}\sum_{r\in\{+,-\}}{u_{jr}^y}^2V(y_j,c_k,f_y)+\frac{\lambda_2}{2}\|f_y\|^2_{\mathcal{H}_{\|_\dagger}}\\ & +\frac{\lambda_3(N_x+N_y)}{2|\mathcal{M}|}\sum_{(i,j)\in\mathcal{M}}(f_x(x_i)-f_y(y_j))^2\\ & +\frac{\lambda_3(N_x+N_y)}{2|\mathcal{C}|}\sum_{(i,j)\in\mathcal{C}}(f_x(x_i)+f_y(y_j))^2\\ \text{s.t.}\quad & u_{ir}^x>=0,\quad u_{jr}^y>=0,\quad \text{for } r \text{ in } \{+,-\}\\ & u_{i+}^x+u_{i-}^x=1,\ u_{j+}^y+u_{j-}^y=1\end{aligned} \tag{5}$$

where $x_1,\ldots,x_{N_x}$ and $y_1,\ldots,y_{N_y}$ are training representations in two views respectively and $N_x$, $N_y$ are the number of them. $|\mathcal{M}|$ and $|\mathcal{C}|$ are the cardinality of $\mathcal{M}$ and $\mathcal{C}$ respectively. We divide the CvMLs and the CvCLs penalties by $|\mathcal{M}|$ and $|\mathcal{C}|$ to balance the different numbers of CvMLs and CvCLs. In this formulation, the first two lines are the probabilistic classification regularization framework in each views, and the last two lines are the CvMLs and the CvCLs penalties.

Note that besides CvMLs and CvCLs, no label information is used here. Thus the learned classifiers can only give out +1 and −1 sign labels. Later, a few labeled samples will be used to determine which real label the sign label corresponds to.

### 3.2. Optimization

Without loss of generality, we use the square loss and propose the regularized least square (RLS) under Cross-View MLs and CLs (RLSCVMC) algorithm. It is easy to see that the representer theorem holds (see appendix for the proof), thus the minimizer $f_x^\star, f_y^\star$ have the following forms:

$$f_x^\star=\sum_{i}^{N_x}\alpha_ik_x(x,x_i) \tag{6}$$

$$f_y^\star=\sum_{j}^{N_y}\beta_jk_y(y,y_j) \tag{7}$$

By substituting them for $f_x$, $f_y$ in Eq. (5), we get the following optimization problem:

$$\begin{aligned}\min_{\alpha,\beta,u^x,u^y} J = {} & \frac{1}{2}\sum_{i=1}^{N_x}\sum_{r\in\{+,-\}}{u_{ik}^x}^2(\alpha^Tk_{x_i}-c_k)^2+\frac{\lambda_1}{2}\alpha^TK_x\alpha\\ & +\frac{1}{2}\sum_{j=1}^{N_y}\sum_{r\in\{+,-\}}{u_{ik}^y}^2(\beta^Tk_{y_j}-c_k)^2+\frac{\lambda_2}{2}\beta^TK_y\beta\\ & +\frac{\lambda_3(N_x+N_y)}{2|\mathcal{M}|}\sum_{(i,j)\in\mathcal{M}}(\alpha^Tk_{x_i}-\beta^Tk_{y_j})^2\\ & +\frac{\lambda_3(N_x+N_y)}{2|\mathcal{C}|}\sum_{(i,j)\in\mathcal{C}}(\alpha^Tk_{x_i}+\beta^Tk_{y_j})^2\\ \text{s.t.}\ & u_{ir}^x>=0,\quad u_{jr}^y>=0,\quad \text{for } r \text{ in } \{+,-\}\\ & u_{i+}^x+u_{i-}^x=1,\ u_{j+}^y+u_{j-}^y=1\end{aligned}$$

where matrices $K_x$, $K_y$ are kernel matrix with their every entry be $k_x(x_i, x_j)$ and $k_y(y_i, y_j)$ respectively. For notation convenience, we let $k_{x_i} = [k(x_i, x_1), \ldots, k(x_i, x_{N_x})]^T$ and $k_{y_j} = [k(y_j, y_1), \ldots, k(y_, y_{N_y})]^T$. Note that the objective is convex with respect to each component though is nonconvex per se. Classic block coordinate descent method could be used to solve this problem [5]. The basic idea is optimizing one block variable while keeping other variable blocks fixed and repeating this step until meeting some stop criteria. Because the objective function value decreases constantly after each step, this procedure guarantees to converge to a stationary point [5].

We optimize ($u^x$, $u^y$), $\alpha$ and $\beta$ iteratively while keeping the rest variable blocks fixed. When $\alpha$ and $\beta$ are fixed, optimizing $u^x$ and $u^y$ can be decoupled into two independent but analogous problems similar to Fuzzy c-means. Eq. (8) calculates current optimal solution $u^x$, and $u^y$ has an analogous formula.

$$u^x_{ir} = \frac{1/(\alpha^T k_{x_i} - c_r)^2}{\sum_{r\in+,-} 1/(\alpha^T k_{x_i} - c_r)^2} \tag{8}$$

When $u^x$, $u^y$ and $\beta$ is fixed, the objective is convex and quadratic with respect to $\alpha$. The current optimal solution could be obtained by setting its derivative to zero. Eqs. (10) and (9) are $\alpha$'s coefficients of linear and quadratic terms. By solving the linear equations $-H_\alpha \alpha = g_\alpha$, we get the current optimal $\alpha$.

$$\begin{aligned} H_\alpha = & \sum_i^{N_x} \sum_{r\in\{+,-\}} {u^x_{ir}}^2 k_{x_i} k^T_{x_i} + \lambda_1 K_x + \frac{\lambda_3(N_x + N_y)}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} k_{x_i} k^T_{x_i} \\ & + \frac{\lambda_3(N_x + N_y)}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} k_{x_i} k^T_{x_i} \end{aligned} \tag{9}$$

$$\begin{aligned} g_\alpha = & -\sum_i^{N_x} \sum_{r\in\{+,-\}} {u^x_{ir}}^2 c_k k_{x_i} - \frac{\lambda_3(N_x + N_y)}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} \beta^T k_{y_j} k_{x_i} \\ & \times \frac{\lambda_3(N_x + N_y)}{|\mathcal{C}|} \sum_{(i,j)\in\mathcal{C}} \beta^T k_{y_j} k_{x_i} \end{aligned} \tag{10}$$

When $u^x$, $u^y$ and $\alpha$ is fixed, $\beta$ can be optimized by analogy with the updating formula of $\alpha$. We omit the related formula of $\beta$ here.

Note that we have closed-form solutions for updating each variable block. We summarize the whole algorithm into Algorithm 1.

**Algorithm 1.** RLS under the CvMLs and the CvCLs

```
Input: data matrix X,Y, number of class, maximum iteration
    number MaxIter
initialize $\alpha$, $\beta$,
while $iter < MaxIter$ do
    step 1: update $u^x$, $u^y$ with Eq. (8) and its analogy for $u^y$.
    step 2: update $\alpha = -H_\alpha \backslash g_\alpha$ by solving a linear equation.
    step 3: update $\beta$ by analogy with updating $\alpha$
end while
```

### 3.3. Determining real labels of the sign labels

In training data, besides the CvMLs and CvCLs available, a few labeled samples are also provided. Since the sign labels of those samples have already been known after the classifiers are learned, the sign labels and the real labels from the same samples can be connected. For example, if a sample has the +1 sign label as well as the first real class label, we may say the +1 sign label corresponds to the first real class label. Note that we can only need as few as two labeled samples, one for each class, to connect the sign labels to real classes.

**Table 1**
Used supervision in Co-RLS and RLSCVMC. Here the cross-view data correspondence in Co-RLS is categorized as ML. Y*: unlabeled data is only used in co-regularization term not in least square loss.

| | Labeled | Unlabeled | CvML | CvCL |
|---|---|---|---|---|
| Co-RLS | Y | Y* | Y | N |
| RLSCVMC | N | Y | Y | Y |

### 3.4. Comparison with co-regularization

RLSCVMC is related to Sindhwani et al.'s co-regularization framework [24]. They proposed a family of algorithms in the co-regularization framework: the Co-Regularized Least Squares (Co-RLS), the Co-Regularized Laplacian SVM and Least Squares (Co-LapSVM, Co-LapRLS). Co-LapSVM and Co-LapRLS place their root on Manifold Regularization framework [4] while our RLSCVMC is built on classical regularization framework. Thus we do not compare with them and only compare with Co-RLS algorithm listed as follows:

$$\begin{aligned} \min_{f_x,f_y} & \sum_i^l (f_x(x_i) - c_i)^2 + \mu \sum_i^l (f_y(y_i) - c_i)^2 + \gamma_1 \|f_x\|^2_{\mathcal{H}_{\|_\S}} \\ & + \gamma_2 \|f_y\|^2_{\mathcal{H}_{\|_\dagger}} + \frac{\gamma C}{(l+u)} \sum_i^{l+u} (f_x(x_i) - f_y(y_i))^2 \end{aligned} \tag{11}$$

where $l$, $u$ mean the numbers of labeled and unlabeled samples respectively. The formulation consists of two classical regularization framework for each data view and a co-regularization term.

We can see that both Co-RLS and our framework are based on the classical regularization framework. However our framework estimates the loss on unlabeled data by introducing probabilistic indicator vectors while Co-RLS only estimates the loss on the labeled data. From the perspective that the paired data can be treated as the CvML, the co-regularization term is exactly the same as the CvML penalty term in Eq. (3). Apparently Co-RLS does not employ the CvCL supervision explicitly while our framework explicitly penalize the violation of the CvCLs. We summarize the different kinds of information used in Co-RLS and our framework in Table 1.

## 4. Experiment

In this section, we show the empirical study on RLSCVMC algorithm. We first introduce the datasets used in our experiments, then show the performance of RLSCVMC under different numbers of the CvMLs and the CvCLs supervision. In the next, we illustrate the classification performance under different parameter settings. Finally, we compare RLSCVMC with Co-RLS algorithm. In all of our experiments, linear kernel is used as in Sindhwani et al.'s co-regularization paper [24].

### 4.1. Dataset description

In our experiments, four multi-view datasets listed below are used.

- The Multiple Feature (handwritten) digit data set (MFD).[1]
  This dataset comes from UCI machine learning repository [12]. It consists of features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. 200 patterns per class (for a total of 2000 patterns) have been digitized in binary images. These digits have six feature sets. Here we only choose two feature sets as two views used in our experiments. They

[1] http://archive.ics.uci.edu/ml/datasets/Multiple+Features.

**Table 2**
Experimental dataset. NumP: #positive, NumN: #negative, VName: View Name.

| Dataset | View 1 | | | | | View 2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | View | Dim | Num | NumP | NumN | View | Dim | Num | NumP | NumN |
| MFD | Pix | 240 | 400 | 200 | 200 | Mor | 6 | 400 | 200 | 200 |
| Course | Page | 3000 | 1051 | 821 | 230 | Link | 1840 | 1051 | 821 | 230 |
| ORL | Pix | 1024 | 40 | 20 | 20 | LBP | 1024 | 40 | 20 | 20 |
| MRC | EN | 21531 | 9433 | 4331 | 5102 | FR | 24893 | 9992 | 5000 | 4992 |

are pix (240 pixel averages in 2 × 3 windows) and mor (6 morphological features). We use '0' as positive class and '1' as negative class.

- Course dataset.
  This dataset consists of 1051 web pages collected from Computer Science department websites at four universities: Cornell, University of Washington, University of Wisconsin and University of Texas. These web pages are categorized into two classes: course and non-course. Two views are web page content and text on the links to the web page.
- ORL.
  This dataset is a face dataset. It contains two feature sets. One is the cropped face image (32 × 32) and the other is the LBP feature extracted from the image. The two feature sets are treated as two views in experiments. We use the first two persons as positive and negative classes.
- Multilingual Reuters Collection (MRC).[2]
  This dataset is defined and provided by [1]. It contains collections of five languages (EN,FR,GE,SP,IT) from six large Reuters categories (CCAT, C15, ECAT, E21, GCAT and M11) extracted from RCV1 and RCV2. This dataset is totally unpaired. In our experiments, we use English and French webpages as two views and the CCAT/C15 categories as the positive and negative class.

Details are listed in Tabel 2. Note that, the first three datasets have paired data while the last does not. In all our experiments, we reduce the dimensions of Course and ORL dataset to 100 and 20 by PCA, and the dimension of MRC to 100 by LSA.

### 4.2. Performance examination of RLSCVMC

In this experiment, we show the performance of RLSCVMC on different numbers of the CvMLs and the CvCLs. Since CvMLs and CvCLs are weak supervision, we want to know how many CvMLs and CvCLs are need to give out a satisfied performance. So we train our framework on different number of training set and see the prediction results. So we use the following experimental settings. The numbers of the CvMLs and the CvCLs grow from 0%$(N_x + N_y)/2$ to 100%$(N_x + N_y)/2$ at an interval of 10%$(N_x + N_y)/2$. For each CvMLs and CvCLs number combination, ten trials are run and the mean accuracies are reported. For each trial, we randomly and evenly split the dataset into training set and testing set. The CvML is constructed by randomly choosing two representations with the same label in the training set of two views, and the CvCL is constructed by randomly choosing two representations with different labels in the training set of two views. The $\alpha$, $\beta$ variables are randomly initialized, and in our experiments we find out that our framework is insensitive to the initial values. Traditional parameter selection method Cross-validation is not applicable here due to the absence of labeled data. So we tune the parameter heuristically. For all experiments we set $\lambda_3 = 1$ which approximately balances the loss term and cross-view side information regularization term. $\lambda_1$ is set to 100 for dataset MFD and 1 for the rest datasets and $\lambda_2$ is set as the same with $\lambda_1$ for convenience. We draw the 3D-bar graph of the mean accuracies in Fig. 2.

All subfigures in Fig. 2 demonstrate that the performance is increasing with the MLs and the CLs numbers. However the increase of performance is not monotonous with respect to the number of the CvMLs and the CvCLs especially for the ORL dataset. That may be caused by the unstable learned predictors on these datasets. Table 3 lists the average standard deviations on the four dataset. Among them, the ORL dataset gets a high average standard deviation and obtains a vibrating increase of performance, while the Course and the MRC datasets have very small average standard deviation and their increasing are almost monotonous.

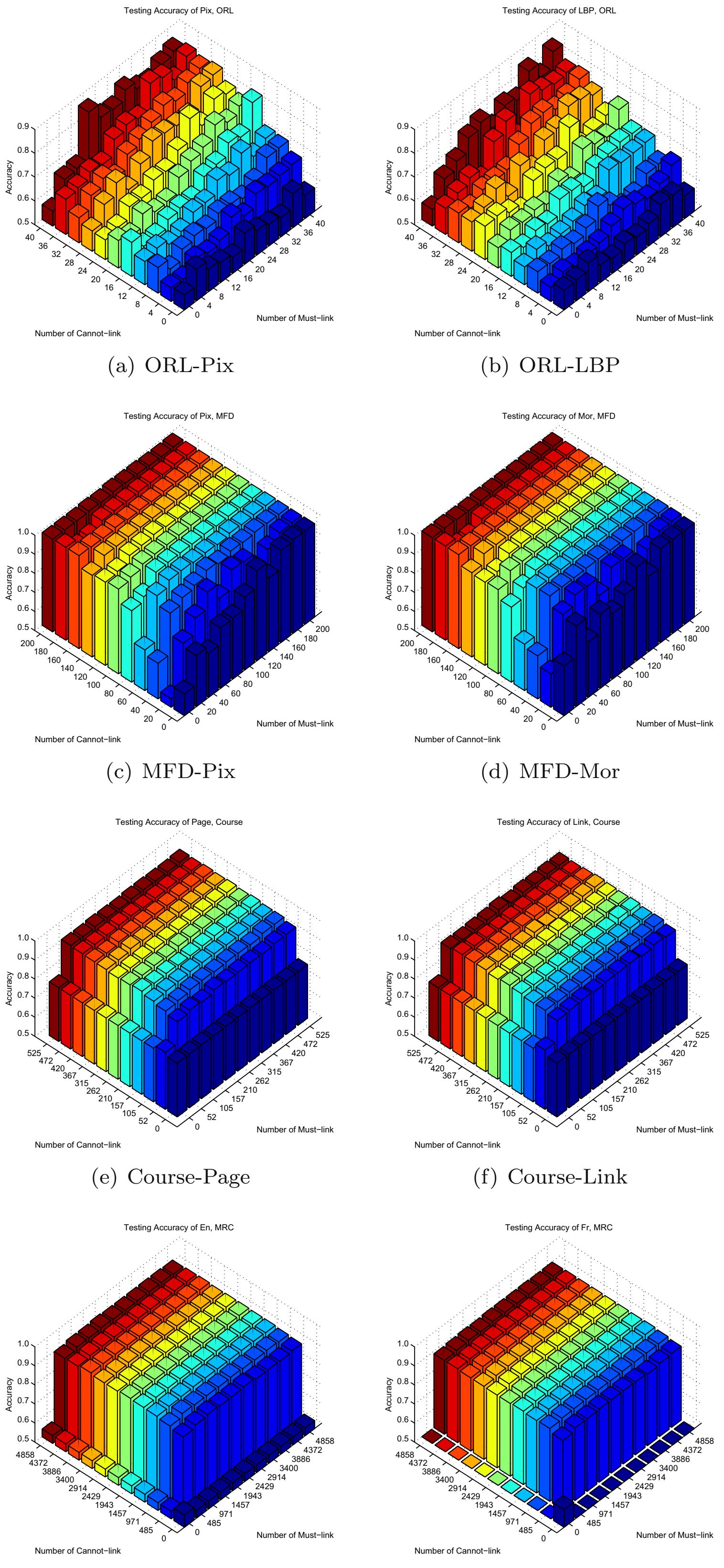
Fig. 4 plots the diagonal bars in Fig. 2 for each dataset. On the Course and the MRC datasets, their accuracies both rise quickly to about 0.9 at 20% supervision, then keep almost stable. While on the ORL dataset, the accuracy increases approximately linearly. In addition, on all but the MFD datasets, the accuracies on their two views assume similar trend which may be partly due to the co-regularization among different views.

An interesting phenomenon is that the accuracy does not increase when the number of CvMLs (CvCLs) increases and the number of CvCLs (CvMLs) is zero, especially on the Course, the MRC and the ORL datasets. Actually it is a degenerated solution of RLSCVMC. We demonstrate this phenomenon by a toy problem in Fig. 3. It is a two-view two-class dataset. Each class in each view is generated from a Gaussian distribution. We draw the classification hyperplane (the blue line), and label positive and negative area on the 2D plane on both views. Fig. 3(a)/(b) depicts the situation when CvCLs/CvMLs does not exist. When only CvMLs exist in Fig. 3(a), the classifiers in both views give all data the same label. Apparently this solution incurs little penalty on the CvMLs regularization because of the same label. Furthermore, due to adaption of the probabilistic indicator vector $u^x$, $u^y$ in Eq. (2), it also incurs small classification loss. When only the CvCLs exist, the situation is similar. The classifiers give data in different views opposite labels, thus incur little penalty on the CvCLs regularization. And the classification loss in Eq. (2) is also small. To avoid this kind of degenerated solution, both the CvMLs and the CvCLs supervision are needed. As we see in Fig. 2(c), (f), (g), and (h), the accuracies dramatically increase from 0%$(N_x + N_y)/2$ to 100%$(N_x + N_y)/2$ CvMLs (CvCLs).

### 4.3. Parameter study

In this experiment, we study the accuracies under different parameter settings. The experiment setting follows the previous experiment. The parameters $\lambda_1$, $\lambda_2$ are set to be the same and are chosen from [1$e$1, 1$e$2, 1$e$3, 1$e$4, 1$e$5]. The parameter $\lambda_3$ is chosen from [1$e$0, 1$e$1, 1$e$2, 1$e$3, 1$e$4, 1$e$5]. The generation of training data and testing data is the same with the previous experiment. For each parameter setting, ten trials are run and the mean accuracies are reported. Due to the space limitation, we only illustrate the results when the number of the CvMLs and the CvCLs are set to be 30%$(N_X + N_y)/2$. We also check the results of different

[2] http://multilingreuters.iit.nrc.ca/ReutersMultiLingualMultiView.htm.



(a) ORL-Pix

(b) ORL-LBP

(c) MFD-Pix

(d) MFD-Mor

(e) Course-Page

(f) Course-Link

(g) MRC-EN

(h) MRC-FR

**Fig. 2.** Performance of LSCVMC.

numbers of the CvMLs and the CvCLs and observe the similar results. Fig. 5 shows the heat map of the accuracies of different parameter settings. On general, the accuracies do not varies very much under different parameter settings. It implies that our framework is not quite sensitive to parameters. The ORL dataset has a relatively unstable accuracies, which could be caused by

**Table 3**
Average standard deviations

| | ORL | MFD | Course | MRC |
|---|---|---|---|---|
| View 1 | 0.099 | 0.039 | 0.010 | 0.005 |
| View 2 | 0.086 | 0.038 | 0.008 | 0.005 |

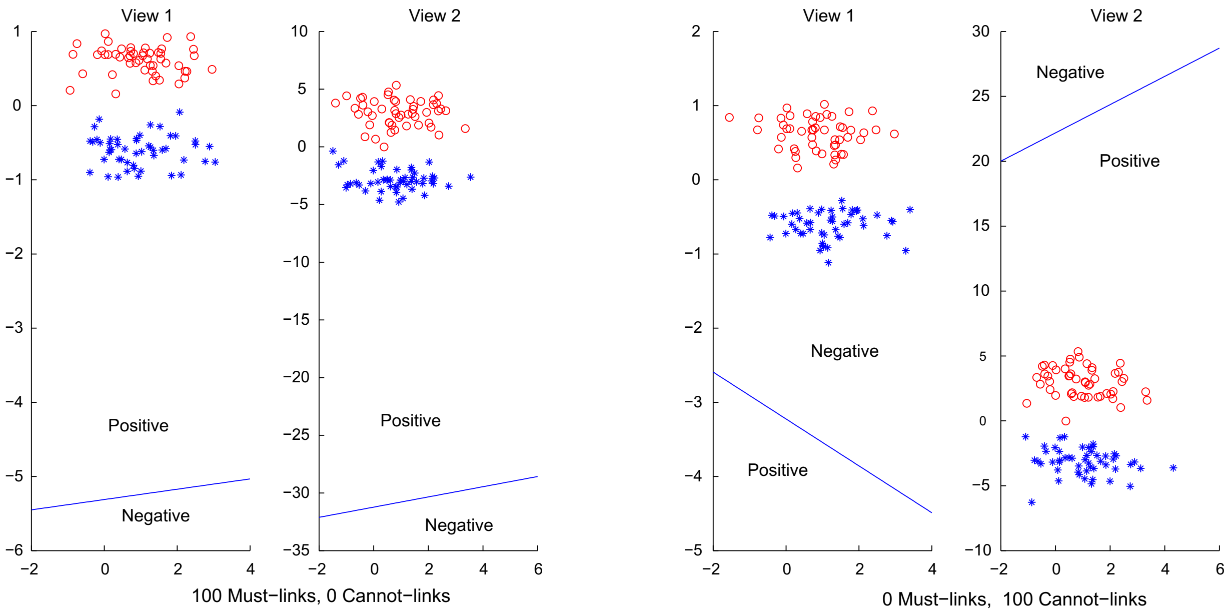
the unstable classifiers learned from too little training data rather than by different parameter settings. The accuracies on the MFD dataset keep above 90% for the most of the parameters and only drop on two group of parameters. On the Course dataset, the accuracies keep high when neither parameters are too large or too small.

### 4.4. Comparison with Co-RLS

In this experiments we compare RLSCVMC with Co-RLS to test the effectiveness of RLSCVMC.

We did not conduct more comparison experiments with other algorithms, mainly because the introduced CvMLs and CvCLs concepts are relatively new. So far as we know, at present there have not had related work based on such side information yet. However, loosely speaking, Co-RLS can be viewed as a related work. Since Co-RLS only works only on full paired data, only the ORL, the MFD and the Course dataset are used in this experiment.

In this experiment, the parameters in RLSCVMC are set to the same as in the above experiment. For Co-RLS, we set $\mu = 1$ and $\lambda_1 = \lambda_2$ in Eq. (11) to make the parameter setting in Co-RLS similar



(a) CvMLs=100,CvCLs=0

(b) CvMLs=0,CvCLs=100

**Fig. 3.** Degenerated solution of RLSCVMC when the CvMLs or the CvCLs does not exist.

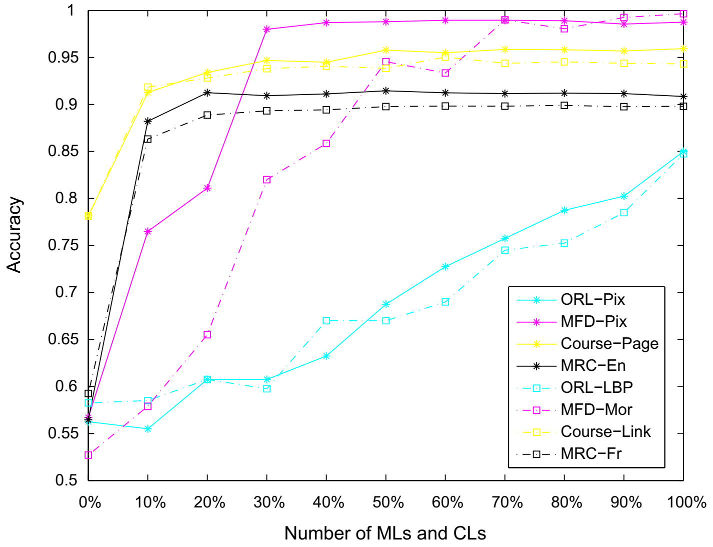
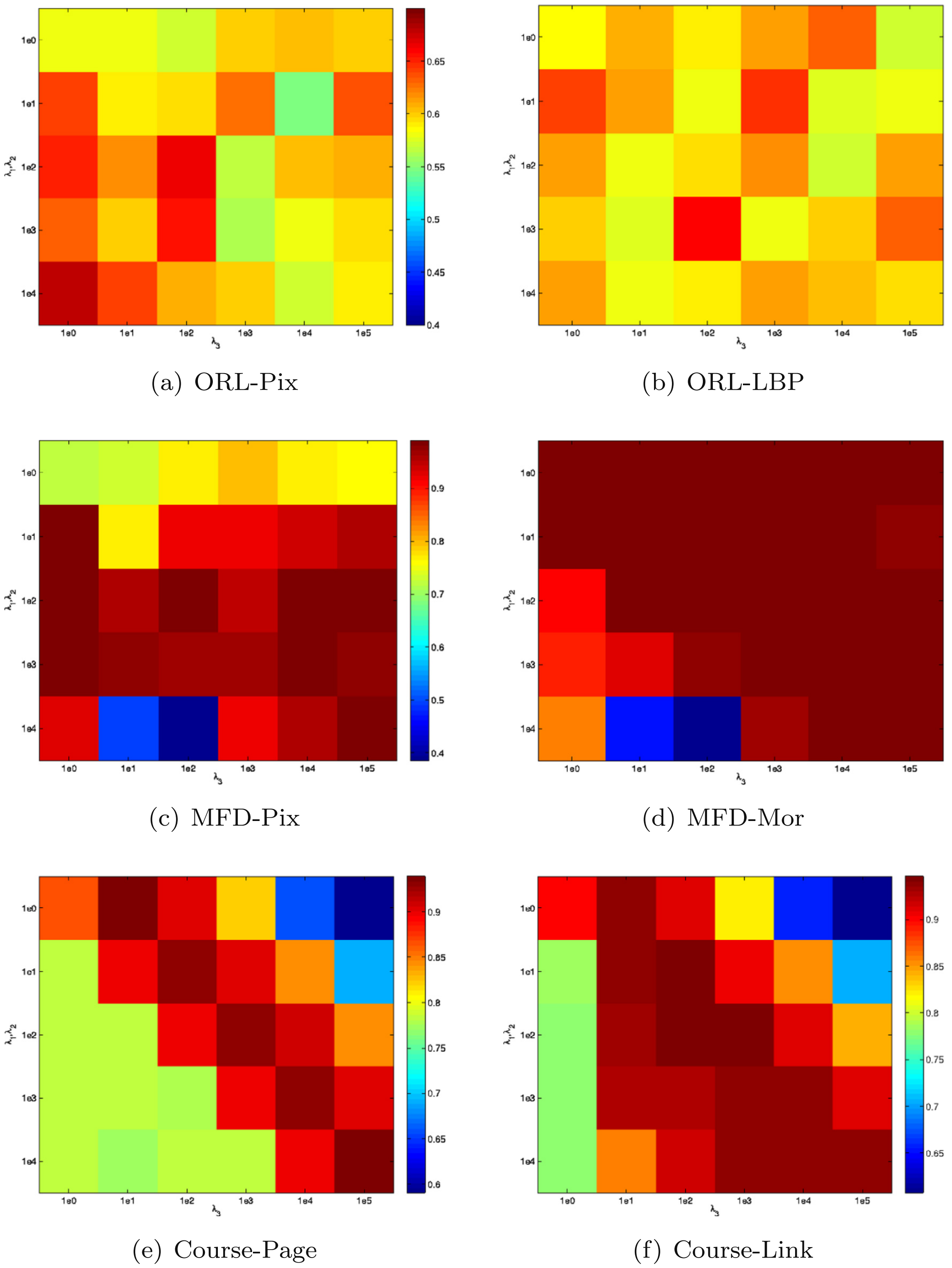


**Fig. 4.** Trend of increasing from small to large set of the CvMLs and the CvCLs.



(a) ORL-Pix (b) ORL-LBP

(c) MFD-Pix (d) MFD-Mor

(e) Course-Page (f) Course-Link

**Fig. 5.** Accuracies under different parameter settings. The *x* and *y* axes indicates the two parameters in our framework. The color indicates the accuracies. Red color means a high accuracy, while blue color means a low accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to that in RLSCVMC. We select the two parameters in Co-RLS by fivefold cross validation both in $\{0.01, 0.1, 1, 10, 100\}$. The means and standard deviations of accuracies are listed in Table 4.

We test the performance on different portions of labeled training data. We first randomly select half as training set and the rest as testing set. Then we choose a part of training set, which increases from 10%*N* to 100%*N* at an interval of 10%*N*, where *N* is the number of training samples, as the labeled set. The rest is used as unlabeled set. In the next, we create every possible the CvMLs and the CvCLs from the labeled training set for RLSCVMC algorithm. Thus the supervised information used in Co-RLS and RLSCVMC is the same. And the difference is that Co-RLS directly uses the labels while RLSCVMC first converts the labels into the CvMLs and the CvCLs and uses them instead. Note that, besides

**Table 4**
Accuracy comparison between RLSCVMC and Co-RLS. The boldface means *t*-test is passed.

| | RLSCVMC | | Co-RLS | |
|---|---|---|---|---|
| | ORL-Pix | ORL-LBP | ORL-Pix | ORL-LBP |
| 10% | .580 ± .043 | .565 ± .058 | .592 ± .141 | .575 ± .144 |
| 20% | .550 ± .031 | .600 ± .083 | **.732** ± **.128** | .670 ± .148 |
| 30% | .680 ± .110 | .662 ± .098 | .710 ± .148 | .690 ± .119 |
| 40% | .710 ± .100 | .657 ± .077 | .702 ± .122 | .677 ± .091 |
| 50% | .737 ± .071 | .727 ± .066 | .742 ± .078 | .730 ± .057 |
| 60% | .807 ± .111 | .722 ± .119 | .817 ± .094 | .695 ± .153 |
| 70% | .827 ± .123 | **.825** ± **.081** | .837 ± .079 | .780 ± .065 |
| 80% | .882 ± .077 | .800 ± .052 | .862 ± .089 | .830 ± .055 |
| 90% | .882 ± .044 | .785 ± .073 | .870 ± .081 | .800 ± .092 |
| 100% | .917 ± .047 | .832 ± .048 | .920 ± .038 | .835 ± .042 |
| | MFD-Pix | MFD-Mor | MFD-Pix | MFD-Mor |
| 10% | **.992** ± **.003** | **.996** ± **.004** | .942 ± .025 | .989 ± .008 |
| 20% | **.991** ± **.004** | **.993** ± **.006** | .952 ± .015 | .986 ± .007 |
| 30% | **.990** ± **.006** | **.994** ± **.005** | .951 ± .035 | .988 ± .007 |
| 40% | **.988** ± **.004** | .994 ± .004 | .963 ± .019 | .991 ± .007 |
| 50% | **.988** ± **.006** | .992 ± .002 | .972 ± .017 | .991 ± .008 |
| 60% | **.988** ± **.008** | .995 ± .004 | .963 ± .022 | .995 ± .005 |
| 70% | **.991** ± **.008** | .991 ± .004 | .971 ± .012 | .990 ± .006 |
| 80% | **.991** ± **.004** | .991 ± .005 | .976 ± .010 | .990 ± .006 |
| 90% | **.986** ± **.009** | .994 ± .004 | .975 ± .011 | .993 ± .007 |
| 100% | **.991** ± **.005** | .994 ± .005 | .977 ± .011 | .992 ± .006 |
| | Course-Page | Course-Link | Course-Page | Course-Link |
| 10% | .923 ± .036 | .894 ± .015 | .930 ± .013 | **.911** ± **.019** |
| 20% | **.943** ± **.011** | .911 ± .013 | .933 ± .013 | .920 ± .016 |
| 30% | **.953** ± **.014** | .922 ± .009 | .944 ± .010 | .924 ± .014 |
| 40% | .954 ± .007 | .935 ± .009 | .948 ± .012 | .939 ± .013 |
| 50% | **.960** ± **.011** | .934 ± .008 | .955 ± .011 | .931 ± .012 |
| 60% | **.962** ± **.005** | .943 ± .006 | .959 ± .006 | .945 ± .008 |
| 70% | **.964** ± **.006** | **.944** ± **.009** | .957 ± .007 | .941 ± .011 |
| 80% | .963 ± .011 | .946 ± .010 | .960 ± .011 | .944 ± .012 |
| 90% | .967 ± .007 | .945 ± .006 | .964 ± .007 | .944 ± .006 |
| 100% | .964 ± .012 | .944 ± .008 | .967 ± .008 | .947 ± .005 |

the label set, Co-RLS also makes full use of all the unlabeled samples and the pairing information on the unlabeled sample set while RLSCVMC does not. Thus in fact, Co-RLS uses much more information than RLSCVMC. In such a circumstance, the current comparisons will naturally be much more prone to Co-RLS than our model.

Through comparing the results of both Co-RLS and RLSCVMC in Table 4, it can be witnessed that our RLSCVMC exhibits comparable performance on two datasets, and significant performance on MFD dataset. Though so, we can find that current RLSCVMC can still leave a further room for its performance promotion since the present experimental setting is more favorable for Co-RLS than for our model. In addition, how to utilize CvMLs and CvCLs more effectively is still needed.

On the ORL dataset, the performance of RLSCVMC and Co-RLS is comparable. Most of the results are statistically not significant except two (20% ORL-Pix, and 70% ORL-LBP).

On the Pix view under 20% labeled samples, the accuracy of Co-RLS is higher than RLSCVMC by over 10 percent, however its variance is also high (0.128) thus is not so convincing.

RLSCVMC outperforms Co-RLS on the MFD dataset especially on the Pix view. On this view, all the results are significantly better than Co-RLS's. On the Mor view, three of ten results are better while the rest are comparable. Note that on the Pix view, RLSCVMC achieves high accuracies even under a small labeled set while Co-RLS does not. With less than or equal to 30% labeled set, RLSCVMC yields an average 4.2% higher accuracy. On the Course dataset, RLSCVMC still beats Co-RLS. On the Page view, half of the accuracies of RLSCVMC are significantly higher than Co-RLS, and the rest are comparable. On the Link view, each of both algorithms obtains a significantly higher accuracy.

So, RLSCVMC, which only uses CvMLs and CvCLs, demonstrates its learning ability, and it achieves better results on the MFD and the Course datasets, and comparative results on the ORL dataset by using less information than Co-RLS.

As discussed in Section 3.4, RLSCVMC estimates the classification loss on unlabeled data and employs the CvCLs explicitly, which makes it achieve better relatively performance than Co-RLS.

## 5. Conclusion and future work

In this paper, we develop a framework which utilizes the cross-view side information, specifically the CvMLs and the CvCLs, to learn classifiers in multi-view circumstance where view data are totally not paired. We show the effectiveness of our framework and demonstrate why the solutions are degenerated when only the CvMLs (CvCLs) are available. In our comparative experiments, we observe that our framework achieves better performance than Co-RLS algorithm under the same supervision. There are still some problems deserved to study in the future. So far, our framework only works for two-view dataset because of the limitation of modeling. How to extend it to the dataset with more than two views is a practical and important question. Furthermore, our CvMLs and CvCLs are general multi-view side information. They are not just limited to classification tasks. How to apply them into multi-view clustering and dimension reduction tasks under totally-unpaired dataset is also deserve examining. The performance of RLSCVML only gets one significant improvement comparing with Co-RLS, although the experiment is more favorable for Co-RLS. There is still a further room for the performance promotion, and this is our next work.

## Acknowledgments

This work was supported in part by the NSFC of China Grant Nos. 61170151, Natural Science Foundation of the Higher Education Institutions of Jiangsu under Grant No. 12KJB520018 and Sponsored by Jiangsu QingLan Project.

## Appendix A

**Theorem 5.1.** *The optimizers* $f_x^{\star}, f_y^{\star}$ *of* Eq. (5) *admit the representations of the form respectively*

$$f_x^{\star} = \sum_i^{N_x} \alpha_i k_x(x, x_i) \tag{12}$$

$$f_y^{\star} = \sum_j^{N_y} \beta_j k_x(y, y_j) \tag{13}$$

**Proof.** We decompose $f_x \in \mathcal{H}_{K_x}(f_y \in \mathcal{H}_{K_y})$ into two parts. The first part is in the subspace spanned by kernel functions $k_x(x_1, \cdot), \ldots, k_x(x_{N_x}, \cdot)$ $(k_y(y_1, \cdot), \ldots, k_y(y_{N_y}, \cdot))$, and the second part is in its orthogonal complement.

$$f_x = f_{x_\parallel} + f_{x_\perp} = \sum_i^{N_x} \alpha_i k(x_i, x) + f_{x_\perp} \tag{14}$$

$$(f_y = f_{y_\parallel} + f_{y_\perp} = \sum_i^{N_y} \beta_j k(y_j, y) + f_{y_\perp}) \tag{15}$$

Then we may write $f_x(x_k)$ and $f_y(y_k)$ as

$$\begin{aligned} f_x(x_k) &= \sum_i^{N_x} \alpha_i k(x_i, x_k) + f_{x_\perp}(x_k) = \sum_i^{N_x} \alpha_i k(x_i, x_k) + < f_{x_\perp}, k(x_k, x) > \\ &= \sum_i^{N_x} \alpha_i k(x_i, x_k) \end{aligned} \tag{16}$$

$$f_y(y_k) = \sum_i^{N_y} \beta_j k(y_j, y_k) + f_{y\perp}(y_k) = \sum_i^{N_y} \beta_j k(y_j, y_k) + < f_{y\perp}, k(y_k, y) > \\ = \sum_j^{N_y} \beta_j k(y_i, y_k) \tag{17}$$

And for all $f_{x\perp}$ and $f_{y\perp}$ we have

$$\|f_x\|^2_{\mathcal{H}_x} = \|f_{x\|} + f_{x\perp}\|^2_{\mathcal{H}_x} = \|f_{x\|}\|^2_{\mathcal{H}_x} + \|f_{x\perp}\|^2_{\mathcal{H}_x} \geqslant \|f_{x\|}\|^2_{\mathcal{H}_x} \tag{18}$$

$$\|f_y\|^2_{\mathcal{H}_y} = \|f_{y\|} + f_{y\perp}\|^2_{\mathcal{H}_y} = \|f_{y\|}\|^2_{\mathcal{H}_y} + \|f_{y\perp}\|^2_{\mathcal{H}_y} \geqslant \|f_{y\|}\|^2_{\mathcal{H}_y} \tag{19}$$

Thus for any fixed $\alpha_i$, $\beta_j$, the function value of objective is minimized for $f_{x\perp} = 0$ and $f_{y\perp} = 0$. Since these are also solutions, the theorem holds. □

## References

[1] Massih-Reza Amini, Nicolas Usunier, Cyril Goutte, Learning from multiple partially observed views – an application to multilingual text categorization, in: Advances in Neural Information Processing Systems (NIPS 2009), 2010.

[2] B. Schoelkopf, A.J. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.

[3] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in Proc. of the SIAM international conference on data mining (SDM 2004), 2004.

[4] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, The Journal of Machine Learning Research 7 (2006) 2399–2434.

[5] D.P. Bertsekas, W.W. Hager, O.L. Mangasarian, Nonlinear Programming, Athena Scientific Belmont, MA, 1999.

[6] M. Blaschko, C. Lampert, A. Gretton, Semi-supervised Laplacian regularization of kernel canonical correlation analysis, Machine Learning and Knowledge Discovery in Databases (2008) 133–145.

[7] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proc. of the 11th Annual Conference on Computational Learning Theory (COLT 1998), 1998.

[8] U. Brefeld, T. Scheffer, Co-em support vector learning, in: Proc. of the 21st International Conference on Machine Learning (ICML 2004), 2004.

[9] S. Dasgupta, M.L. Littman, D. McAllester, Pac generalization bounds for co-training, in: Advances in Neural Information Processing Systems (NIPS 2002), 2002.

[10] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of the 24th International Conference on Machine learning (ICML2007), 2007.

[11] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, New York, 2001.

[12] A. Frank, A. Asuncion, UCI Machine Learning Repository, 2010.

[13] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: Advances in Neural Information Processing Systems (NIPS2006), 2005.

[14] A.B. Goldberg, X. Zhu, S. Wright, Dissimilarity in graph-based semi-supervised classification, in: Proc. of the 11th Artificial Intelligence and Statistics (AISTATS 2007), 2007.

[15] Jingjing Gu, Songcan Chen, Tingkai Sun, Localization with incompletely paired data in complex wireless sensor network, IEEE Transactions on Wireless Communications (2011).

[16] T. Hertz, N. Shental, A. Bar-Hillel, D. Weinshall, Enhancing image and video retrieval: learning via equivalence constraints, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 2003.

[17] E. Hu, S. Chen, D. Zhang, X. Yin, Semisupervised kernel matrix learning by kernel propagation, IEEE Transactions on Neural Networks 21 (2010) 1831–1841.

[18] A. Kimura, H. Kameoka, M. Sugiyama, T. Nakano, E. Maeda, H. Sakano, K. Ishiguro, Semicca: Efficient semi-supervised learning of canonical correlations, in: Proc. of the 20th International Conference on Pattern Recognition (ICPR 2010), 2010.

[19] C. Lampert, O. Kromer, Weakly-paired maximum covariance analysis for multimodal dimensionality reduction and transfer learning, in: Proc. of the 11th European Conference on Computer Vision (ECCV 2010), 2010.

[20] G. Li, S.C.H. Hoi, K. Chang, Two-view transductive support vector machines, in: Proc. of the SIAM International Conference of Data Mining (SDM 2010), 2010.

[21] Z. Li, J. Liu, X. Tang, Pairwise constraint propagation by semidefinite programming for semi-supervised classification, in: Proc. of the 25th International Conference on Machine Learning (ICML 2008), 2008.

[22] N. Nguyen, R. Caruana, Improving classification with pairwise constraints: a margin-based approach, Machine Learning and Knowledge Discovery in Databases (2008) 113–124.

[23] S. Shalev-Shwartz, Y. Singer, A.Y. Ng, Online and batch learning of pseudo-metrics, in: Proceedings of the 21st International Conference on Machine Learning (ICML2004), 2004.

[24] V. Sindhwani, P. Niyogi, M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, in: Workshop on Learning with Multiple Views at ICML, 2005.

[25] T. Sun, S. Chen, J. Yang, X. Hu, P. Shi, Discriminative canonical correlation analysis with missing samples, in: Computer Science and Information Engineering (CSIE 2009), 2009.

[26] W. Tong, R. Jin, Semi-supervised learning by mixed label propagation, in: Proc. of the 22th National Conference on Artificial Intelligence (AAAI 2007), 2007.

[27] V.N. Vapnik, Statistical Learning Theory, Wiley-Interscience, 1998.

[28] K. Wagstaff, C. Cardie, S. Rogers, S. Schrodl, Constrained k-means clustering with background knowledge, in: Proc. of the 18th International Conference on Machine Learning (ICML 2001), 2001.

[29] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Advances in Neural Information Processing Systems (NIPS2002), 2002.

[30] R. Yan, J. Zhang, J. Yang, A.G. Hauptmann, A discriminative learning framework with pairwise constraints for video object classification, IEEE Transactions on Pattern Analysis and Machine Intelligence (2006) 578–593.

[31] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: Proc. of the 33rd Annual Meeting on Association for Computational Linguistics (ACL 1995), 1995.

[32] T. Yoshida, K. Okatani, A graph-based projection approach for semi-supervised clustering, Knowledge Management and Acquisition for Smart Systems and Services (2011) 1–13.

[33] S. Yu, J. Shi, Grouping with directed relationships, in: Energy Minimization Methods in Computer Vision and Pattern Recognition, 2001.

[34] J. Zhang, R. Yan, On the value of pairwise constraints in classification and consistency, in: Proc. of the 24th International Conference on Machine Learning (ICML 2007), 2007.