ABSTRACT

Complex media objects are often described by multi-view feature groups collected from diverse domains or information channels. Multi-view learning, which attempts to exploit the relationship among multiple views to improve learning performance, has drawn extensive attention. It is noteworthy that in some real-world applications, features of different views may come from different private data repositories, and thus, it is desired to exploit view relationship with data privacy preserved simultaneously. Existing multi-view learning approaches such as subspace methods and pre-fusion methods are not applicable in this scenario because they need to access the whole features, whereas late-fusion approaches could not exploit information from other views to improve the individual view-specific learners. In this paper, we propose a novel multi-view learning framework which works in a hybrid fusion manner. Specifically, we convert predicted values of each view into an Accumulated Prediction Matrix (APM) with low-rank constraint enforced jointly by the multiple views. The joint low-rank constraint enables the view-specific learner to exploit other views to help improve the performance, without accessing the features of other views. Thus, the proposed RANC framework provides a privacy-preserving way for multi-view learning. Furthermore, we consider variants of solutions to achieve rank consistency and present corresponding methods for the optimization. Empirical investigations on real datasets show that the proposed method achieves state-of-the-art performance on various tasks.

Categories and Subject Descriptors
H.2.8 [Database Management]: [Database Applications – Data Mining]; I.2.6 [Artificial Intelligence]: [Learning]

General Terms
Algorithms, Applications, Experimentation

Keywords
Multi-View Learning; Rank Consistency; Privacy-Preserving

1. INTRODUCTION

With the rapid development of the Internet, a huge number of rich media objects can be collected from various information channels, e.g., nowadays a piece of news can be naturally described using text, audio, video clip and hyperlink[17][35]. In this paper, we focus on problems where data are gathered from multiple private information channels, i.e., features cannot be shared during processing. This problem occurs frequently in many scenarios, e.g., knowledge management/analysis on database with different views/tables whose access are prohibited from different users; multi-view data analysis when cooperation-competition relationship exists among enterprises or researchers. In a more concrete case of image analysis, researchers can extract their own features of images as well as get the retrieval results from Google. Nevertheless they cannot obtain the image features used by Google, i.e., the features of researchers or Google are information from different private channels.

In order to analyze complex information from multiple channels, multi-view learning has attracted extensive attention [21][24][34]. Various multi-view learning approaches can be classified into four groups: multi-view subspace learning algorithms aim at obtaining a common subspace shared by multiple views and then learn models in that shared subspace [13][27]; pre-fusion methods like multiple kernel learning [11] mainly fuse feature information by weighted combination of kernels produced separately on each view; late fusion methods combine outputs of the models constructed from different view-specific features but leave the classifiers training phase unimproved [36]; disagreement-based methods focus on how to use unlabeled data to enhance the performance of learners via the compatibility between two views [5][31][39].

The aforementioned methods have achieved great success in various tasks, except scenarios with cooperation competition where access to multi-view features is restricted. Subspace style and pre-fusion approaches have to interact with view features, so they could not protect the privacy of different information channels. Late fusion methods only build a combined model based on outputs from each view, and therefore, the information from other information channels could not help improve learning ability of view-specific classifiers. Most existing disagreement-based multi-view learning approaches, such as co-training [5], rely on strong assumptions like redundant and independent views; though recent theoretical studies [32][33] disclosed that weaker assumption is sufficient, new effective algorithms are still in design.

In this paper, we propose a novel framework working in a hybrid fusion manner. This framework, RANC (RANk Consistency), can be easily applied to the cooperation-competition multi-view learning scenarios, because each view-specific learner is able to exploit the information from other views to improve the performance with-
out accessing features of these views; in other words, the RANC framework provides a way to privacy-preserving multi-view learning. The defined rank consistency is a criterion for seeking consistent predictions on multiple views. We formulate it by introducing an Accumulated Prediction Matrix (APM) which is stacked by predicted values/labels of each view. It is notable that in the ideal case, the predicted results of multi-view models should be consistent on each view so the rank of APM should be equal to \( C - 1 \), where \( C \) is the number of classes. However, the rank of APM is usually larger than \( C - 1 \) in practice for the potential inconsistency among multi-view predictions. Practically, lower rank of APM implies more view consistency. Fig. 1 is an illustration of APM on a multi-view dataset Reuters, which gives a spectrum plot on singular values of APM. The x-axis are singular values sorted in descending order. The spectrum is long-tail distributed started from the \((C - 1)\)th singular value, which implies low rank property of APM. There are two obvious knee points (KP) in Fig. 1. KP_1 may be caused by the ambiguities among \( C \) classes in Reuters. Note that KP_1 can be vanished for separable problems. KP_2 at \( C - 1 \) clearly reveals the rank consistency property in multi-view problems. Motivated by this observation from Fig. 1, RANC reduces the long tail components of APM to leverage predictions close to the ideal case, and apparently is naturally designed for applications with two or more views. Since those properties mentioned above can be estimated on the outputs from each view instead of directly using their features, the data privacy can be maintained to a greatest extent.

In our proposed framework, rank consistency criterion can be transformed into a rank regularizer term. Specifically, we use the truncated nuclear norm \([15]\) to model it which can be incorporated with many different losses. Solving this framework leads to enhance the classification ability of view-specific predictor. RANC constructs a hybrid fusion paradigm combining advantages of both pre-fusion and late-fusion methods. In this paper, we demonstrate this framework with square loss in detail and the implementation can be optimized effectively with both Proximal Gradient (PG) and Alternative Direction Method of Multipliers (ADMM) techniques. Furthermore, the paper also presents an accelerated version with rank-one update which can also get satisfying results. We empirically validate the effectiveness of our framework and our model achieves significantly better performance on various tasks. The main contributions of this paper can be summarized as follows:

- A novel rank consistency criterion based multi-view learning framework (RANC), which preserves data privacy of multiple channels, i.e., the learner on each view will access to the corresponding view features only and its interactions with other views are limited.

- RANC can naturally handle data with more than two views. Besides, it can help improve the learning ability of the individual predictor during the training phase.

- Our solutions to RANC are rendered effective. Meanwhile, an accelerated implementation is also presented.

Section 2 gives the related work. The main proposed framework together with detailed solutions is presented in section 3. In section 4, empirical investigations on real datasets are discussed. Finally, we conclude in section 5.

2. RELATED WORK

How to exploit relationship among multiple views is fundamental to multi-view learning approaches. As mentioned above, there are 4 categories of multi-view learning methods, and their strategies of exploiting view relationship are of diversity to improve performance. Subspace approaches seek correlations between views, and subsequent tasks like classification will be performed in the learned subspace. E.g., CCA \([13]\) finds a linear projection for each of two views and generates a shared subspace where the correlation between two views is maximized. Some others, such as MvDA \([18]\) and MvLPP \([27]\), are of similar intuitions. Pre-fusion methods consider fusing features or feature derivations (e.g., kernels \([11]\), distances \([37]\)) before the training phase. It is typical that they represent the features as multiple kernel matrices and then combine them in a kernel space. Multiple kernel learning, one of the widely used pre-fusion methods, learns a linear \([9]\) or nonlinear \([10]\) combination of kernels for classification. These two types of methods, however, both require access to features of all views.

Late fusion strategies like RLF \([36]\), aim at fusing the predicted values of each view while leaving multi-view features untouched. Nevertheless, the fusion process is not involved in classifier training phase, which makes late fusion not helpful in enhancing learners of each view. Moreover, it is required to retrain the whole late fusion model when the predictions on any of the views are changed, and this makes late fusion methods expensive and lack of flexibility.

Disagreement-based methods, such as Co-Training \([5]\), focus on exploiting unlabeled data in semi-supervised learning scenarios. Traditional Co-Training enhances the performance of learners via the compatibility between views, but can hardly deal with multiple views \([39]\). By comprehensively considering the advantages and drawbacks of existing multi-view learning approaches, the proposed framework RANC defines a new criterion named rank consistency to characterize the consistency of predictions among multiple views. Rank consistency is implemented with matrix rank minimization regularizer. Rank minimization for a matrix is usually used in multi-class \([12]\) or multi-label \([22]\) problem, where different labels (output) are related to uniform input space (single input). In multi-label learning, rank based regularization is often used for reflecting the correlation between labels. However, in multi-view settings, classifiers are functional mappings which stretches across different views (input) and a single output. Consequently, it is required to define the rank consistency regularizer with a different data structure, namely, the accumulated prediction matrix (APM). Formulated on prediction matrix APM, RANC can protect features of individual view from being accessed by learners built on other views. Besides, RANC is fully incorporated in training of individual classifier on each view to refining the classifiers with updated predictions.

Figure 1: A typical singular value spectrum plot of APM on Reuters (5 views and 6 classes). APM is obtained with linear SVM on each view separately.
3. RANK CONSISTENCY BASED MULTI-VIEW LEARNING

In multi-view learning, an instance is characterized by multiple groups of features while they are only with one unified label. Without loss of generality, we suppose there are \( K \) views and each view has \( n \) instances, where the first \( l \) of \( n \) instances are labeled, and the rest \( n - l \) are unlabeled. The \( r \)th instance \( x_i \) can be represented as a collection of view-specific vectors \( x_{i;k} \in \mathbb{R}^{d_k} \), where \( d_k \) is the dimension of the \( k \)th view. For labeled examples, instance \( x_i \) has label \( y_i \in \{0, 1\} \) in binary classification problem. In multi-class cases with \( C \) classes, the label \( y_i \) for instance \( x_i \) is expanded to a vector with \( C \) elements, where \( y_{i,j} = 1 \) indicates the \( j \)th instance is with label \( j \), otherwise \( y_{i,j} = 0 \). The whole data for \( k \)th view can be expressed as \( X_k = [x_{1;k} \ | \ x_{2;k} \ | \ \cdots \ | \ x_{n;k}] \in \mathbb{R}^{n \times d_k} \) and the corresponding labels can be expressed as \( Y = \{0, 1\}^{l \times C} \). Classifier on each view is denoted as \( f_k : x_{i;k} \rightarrow \hat{y}_i \), where \( x_{i;k} \in \mathbb{R}^{d_k} \), \( \hat{y}_i \in \mathbb{R}^C \), and loss of instance \( i \) on view \( k \) is defined as \( \ell(f_k(x_{i;k}), y_i) \). Then the prediction of \( X_k \) on the \( k \)th view is combined into \( F_k = [f_k(x_{1;k}) \ | \ f_k(x_{2;k}) \ | \ \cdots \ | \ f_k(x_{n;k})] \in \mathbb{R}^{n \times C} \), and predictions from all views can be stacked into an Accumulated Prediction Matrix (APM), which can be defined as \( F = \{F_1, F_2, \ldots, F_K\} \in \mathbb{R}^{n \times CK} \).

Ideal multi-view classifiers \( f_k \), \( k = 1, \ldots, K \), give identical outputs for a binary classification problem and consequently make the rank of APM \( F \) equal to one. Yet in practice, the rank of APM could not be exactly equal to one. As Fig. 1 shows, in a six-class problem, the singular values of APM in descending order reveal an “exponential like” decay with a long tail in the right part of Fig. 1. In particular, the second knee point appears at the 5th singular value. This is obviously consistent with our assumption that in practical cases, the rank of APM should be equal to the freedom degree of class number \( C - 1 \). This phenomenon implies the predictions from all views tend to be with low rank when multi-view classifiers are not exactly identical. To induce consistency among multiple classifiers in multi-view learning, the rank consistency therefore can be defined as:

**Definition 1.** Rank consistency for predictions on multiple views on a certain data collection is an operator \( RC(\cdot) : \mathbb{R}^{n \times CK} \rightarrow \mathbb{R} \), which defined on the APM \( F \), and we define \( RC(F) = \text{rank}(F) \).

We can summarize two fundamental properties of the defined rank consistency operator as follows:

**Property 1.** \( RC(F) \) reflects the prediction compatibility among views. A large value of \( RC(F) \) implies imperfection of prediction consistency, and a small value indicates predictions from all views are aligned well. (**Qualitative property**)

**Property 2.** The expected rank consistency is \( C - 1 \), which is the freedom degree of label assignments in all views for concerned dataset. In particular, for a binary problem it is equal to 1. (**Quantitative property**)

Rank consistency can be easily used as a regularizer in a learning framework, which is helpful to achieve compatible and consistent predictions upon all views and can generate better classifier on each view. In next subsections, we first propose the whole RANC framework based on the defined rank consistency. After implementing concrete classifiers for each view, we show the proposed framework with rank consistency as a regularizer can be effectively solved with different techniques.

### 3.1 Rank Consistency Framework

The key to the proposed method is the use of the rank consistency, which boosts performance of view-specific learner by seeking for prediction consistency. Benefitted from the rank consistency as a regularizer, we can bridge the maximization of label consistency among multiple views and the part of the classification task together. We define the RANC framework as:

\[
\min_{F_k \in \mathcal{D}} \sum_{k=1}^{K} L_k(F_k, Y) + \lambda RC(F). \tag{1}
\]

There are \( K \) views and \( F \) is the APM. The first term \( L_k \) depicts the objective functions according to the property of the \( k \)th view. Furthermore, \( L_0 \) and \( L_1 \) can be different while the predictor on each view is self-adaptive. The second term, \( RC(\cdot) \), is the rank consistency operator on APM, which leverages the prediction consistency to enhance learner on each view. \( \lambda > 0 \) is a balance parameter reflects the weights between view-specific objective and rank consistency regularizer.

Specifically, objective function \( L_k \) on the \( k \)th view in Eq. 1 is generally with the form of a regularized empirical loss:

\[
\min_{F_k} L_k(F_k, Y) = \ell(F_k, Y) + \gamma \tau(F_k),
\]

here \( \tau(\cdot) \) is the regularizer for view-specific classifier. \( \gamma > 0 \) is a scalar coefficient to balance the weights of the two terms. Here the loss term \( \ell(\cdot) \) can take several forms, e.g., square loss or hinge loss for both linear and nonlinear problems. Eq. 1 indicates the classifier in \( \ell(\cdot) \) and the prediction results \( F_k \) for instances are connected, which provides the possibilities of refining predictions with rank consistency by optimizing them simultaneously. To simplify the discussion, here we use the regularized square loss as the basic objective function for each view, with linear classifiers \( W_1, \ldots, W_K \):

\[
\begin{aligned}
\min_{W_k, b_k, F_k} & \ |X_kW_k + b_k|^2 + \gamma ||W_k||_F^2 \\
\text{s.t.} & \ F_k \in \mathcal{D} : F_k^{1 \cdots l} = Y, 0 \leq F_k \leq 1.
\end{aligned} \tag{2}
\]

The feasible domain of \( F_k \) is \( \mathcal{D} \). Constraint \( F_k^{1 \cdots l} = Y \) restricts the prediction on labeled data the same as the ground truth to avoid collapsing of predictions, where \( F_k^{1 \cdots l} \) is the first \( l \) rows of the prediction matrix \( F_k \). In addition, it constrains predicted values into the same range as true labels by \( 0 \leq F_k \leq 1 \) to avoid trivial solutions. In Eq. 2, \( b_k \in \mathbb{R}^C \) is the bias for current predictor. By centralizing both instances and predictions, the objective can be rewritten as follows:

\[
\ell(W_k, F_k) = ||HX_kW_k - HF_k||_F^2 + \gamma ||W_k||_F^2.
\]

\[
H = I - \frac{1}{n} 11^T \text{ is the centralization matrix, where } I \text{ is the identity matrix and } 1 \text{ is a vector with all the elements equal to one. Without loss of generality, we can assume the data matrix } X \text{ is centralized so that } GX_k = X_k. \text{ Furthermore, by taking derivative of Eq. 3 w.r.t. } W_k \text{ and setting it to zero, we have:}
\]

\[
W_k = (X_k^T X_k + \gamma I)^{-1} X_k^T F_k. \tag{4}
\]

Combining Eq. 3 and Eq. 4, we simplify the rank consistency framework in Eq. 1 into a form only relying on the predicted values \( F_k \) of each view, i.e., the APM \( F \), as follows:

\[
\min_{F_k \in \mathcal{D}} \sum_{k=1}^{K} \text{Tr}(F_k^T (H - X_kX_k^T + \gamma I)^{-1} X_k^T)F_k + \lambda RC(F),
\]

here \( Tr(\cdot) \) is the matrix trace operator.
3.2 Directional Rank Consistency Optimization with Truncated Nuclear Norm

Rank norm minimization is NP-hard and nuclear norm (or trace norm) [7] usually acts as a convex surrogate. For a matrix $X \in \mathbb{R}^{m \times n}$, we assume its singular values $\sigma_i, i = 1, \ldots, \min(m, n)$, are ordered from large to small. The nuclear norm is defined as $\|X\|_* = \sum_{i=1}^{\min(m, n)} \sigma_i$. Nuclear norm has been widely used in various scenarios where rank norm minimization is required [12].

However, the quantitative property of rank consistency indicates consistent predictions of learners constructed on each view respectively always have $C-1$ freedom degree on sufficient large dataset. Blindly minimizing the rank of APM will break the natural structure of multi-view predictions and may lead to degeneration of classification performance. Therefore, a directional optimization approach, which can conduct the RC($F$) until converging to $C-1$ during the minimization procedure, is desired in our task. Inspired by [15], we use truncated nuclear norm as a surrogate function of the RC($\cdot$) operator:

Definition 2. Given a matrix $X \in \mathbb{R}^{m \times n}$, the truncated nuclear norm $\|X\|_r$ is defined as the sum of $\min(m, n) - r$ minimum singular values, i.e., $\|X\|_r = \sum_{i=r+1}^{\min(m, n)} \sigma_i(X)$.

Different from traditional nuclear norm minimization with all singular values preserved, truncated nuclear norm minimizes singular values with first $r$ largest ones unchanged, which is more close to the true rank definition. If $\|X\|_r = 0$, there are only $r$ non-zero singular values for $X$, and this explicitly indicates rank of $X$ is less than or equals to $r$. Practically, in order to impel the RC($F$) directional to the freedom degree of the APM, it is clear to set $r = C-1$ in multi-view learning scenarios.

The truncated nuclear norm can be formulated as the equivalent form by the following theorem [15]:

**Theorem 1.** Given a matrix $X \in \mathbb{R}^{m \times n}$ and any non-negative integer $r (r \leq \min(m, n))$, for any matrix $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{r \times n}$ such that $AA^\top = I_r, BB^\top = I_r$, where $I_r \in \mathbb{R}^{r \times r}$ is identity matrix. Truncated nuclear norm can be reformulated as:

$$
\|X\|_r = \|X\|_* - \max \text{Tr}(AXB^\top).
$$

If the singular value decomposition of matrix $X$ is $X = U\Sigma V^\top$ where $\Sigma$ is the diagonal matrix of singular values sorted in descending order and $U \in \mathbb{R}^{m \times n}$, $V \in \mathbb{R}^{n \times n}$. The optimal solution for the trace term in the above equation has a closed form solution: $A = (u_{11}, u_{22}, \ldots, u_{rr})^\top$ and $B = (v_1, v_2, \ldots, v_r)^\top$, corresponds to the first $r$ columns of left and right singular vectors.

With Theorem 1, we can reformulate our objective function as:

$$
F = \min_{F_k \in \mathcal{D}} \sum_{k=1}^{K} L_k(F_k) + \|F\|_r
$$

$$
= \min_{F_k \in \mathcal{D}} \sum_{k=1}^{K} L_k(F_k) + \|F\|_* - \max \text{Tr}(AXB^\top),
$$

s.t. $AA^\top = I, BB^\top = I$.

Because of the non-convexity of truncated nuclear norm, alternative approaches can be utilized for the optimization. A simple solution to Eq. 6 is alternating descent method. We can fix $F$ and optimize $A, B$ via SVD on $F$ first, and then fix $A$ and $B$ to optimize $F$. When $A$ and $B$ are fixed, the subproblem is convex. The whole procedure is summarized in Algorithm 1.

In step 2, $A$ and $B$ can be obtained by SVD on $F$, which are the left and right singular vectors corresponding to the maximum $C-1$ singular values. As the number of actually required singular vectors is rather small, partial SVD can be used for more of efficient [3]. The most computational cost step, however, is the subproblem for solving $F$ in Eq. 7. We will give a detailed investigation on employing Accelerate Proximal Gradient Descent Method (APG) [2] and Alternative Direction Method of Multipliers (ADMM) [6] for solving this subproblem in the following subsections.

It is noteworthy that the whole training procedure enhances the view-specific classifier only based on alternative updates of the prediction matrix $F_k$ and its singular vectors. After each round of updating $F_k$, updated view-specific prediction will be passed back to corresponding learner of each view for model refinements. So RANC restricts interactions among multiple views within predictions without the access to original features of other views.

3.3 Solving RANC with APG

Note that when $A$ and $B$ are fixed, the problem is composed of two convex parts, i.e., a smooth loss term $P_1(F)$ and a non-smooth trace norm $P_2(F)$:

$$
P_1(F) = \sum_{k=1}^{K} L_k(F_k) - \text{Tr}(AFB^\top), \quad P_2(F) = \|F\|_*.
$$

APG is suitable for solving Eq. 8 [16], which optimizes on a linearized approximation version of the original problem. In the $t$th iteration, if we denote the current optimization variable as $F^t$, then we can linearize the smooth part $P_1(\cdot)$ at $F^t$ as:

$$
Q(F) \approx P_1(F^t) + \text{Tr}(\nabla P_1(F^t), F - F^t)
$$

$$
+ \frac{L}{2} \|F - F^t\|^2_F + P_2(F)
$$

$$
= \sum_{k=1}^{K} L_k(F^t_k) - \text{Tr}(AFB^\top) + \text{Tr}(\nabla P_1(F^t), F - F^t)
$$

$$
+ \frac{L}{2} \|F - F^t\|^2_F + \lambda \|F\|_* ,
$$

where $\nabla P_1(F^t) = 2[E_1F_{1}^t, E_2F_{2}^t, \ldots, E_kF_{k}^t] - \lambda A^\top B$, where $E_k = H - X_k (X_k^\top X_k + \gamma I)^{-1} X_k^\top, k = 1, \ldots, K$. Here $L$ is the Lipschitz coefficient, which can be estimated by line search strategy [2]. Minimizing $Q(F)$ w.r.t. $F$ is equivalent to solving:

$$
\hat{F} = \arg \min_{F} \lambda \|F\|_* + \frac{L}{2} \|F - (F^t - \frac{1}{L} \nabla P_1(F^t))\|^2_F.
$$

**Algorithm 1** The pseudo code of RANC

**Require:** training instances $X_k$ for each view, parameters $\lambda$ and $\gamma$, $\epsilon$, initialize $F$ using true label matrix;

1: while True do
2: Use $F$ to solve $A$ and $B$ as Theorem 1;
3: Solve $F$ with fixed $A$ and $B$, i.e.,

$$
F^{t+1} = \arg \min_{F_k \in \mathcal{D}} \sum_{k=1}^{K} L_k(F_k) + \|F\|_* - \text{Tr}(AFB^\top) (7)
$$

4: if $\|F^{t+1} - F^t\| \leq \epsilon$ then
5: Break;
6: end if
7: end while
8: Solve $W_k$ from $F$ using Eq. 4.
9: return $W_k$, classifier for each view.
Algorithm 2: The pseudo code for solving Eq. 7 with APG.

Require: $\gamma, \alpha > 1$, initialize $Z^*$ and $F^*$; while Stop criterion doesn’t meet do
1: Line search for best step-size $L$
2: $\hat{F}^{t+1} = \mathcal{D}_L(F^*) + \frac{1}{L} \nabla P_t(F^*)$
3: $\alpha_{t+1} = \frac{1}{\sqrt{1 + \epsilon^2}}$
4: $Z^{t+1} = \hat{F}^{t+1} + (\alpha_{t+1}/\gamma)(F^{t+1} - F^*)$
6: end while
7: return $F_{k}$.

The framework with APG in Algorithm 2.

Feature privacy is maintained. After we get $\Lambda$, we can solve Eq. 9 in a closed form:

$$L_{\beta} = \sum_{k=1}^{K} \text{Tr}(F_k^T E_k F_k) + \lambda\|U\|_* - \lambda \text{Tr}(A^T B^T)$$

$$- \langle A, F - (F - 1b^T) \rangle + \frac{\beta}{2} \|U - (F - 1b^T)\|_F^2,$$

$\beta > 0$ is a scalar for the augmented term. The problem in Eq. 11 is similar to those problems which can be solved with ADMM [6]. However, here we have three blocks of variables in Eq. 11, namely blocks with $U$, $b$ and $F$, which is with very different properties rather than the ordinary ADMM problems. Using ADMM directly can hardly get converged [29], consequently we employ a variant augmented Lagrange dual optimization technique [14] for our problem, which is a splitting variant of ADMM. After letting $Q = \mathbf{1} b^T$, we can solve an optimal candidate of $Q^{t+1}$ by taking derivative of Eq. 11 w.r.t. $Q$ and set it to zero, thus we have:

$$Q^{t+1} = \frac{1}{\beta} A^T - U^T + F^T.$$ (12)

Then the dual variable can be updated as:

$$\Lambda^{t+\frac{1}{2}} = \Lambda^t + \beta(Q^{t+1} + U^T - F^T).$$ (13)

With the updated $\Lambda$, the remaining variable $U$ can be updated with an added proximal term:

$$\min_U \lambda\|U\|_* - \lambda \text{Tr}(A^T B^T) - \langle \Lambda^{t+\frac{1}{2}}, U \rangle + \frac{\mu\beta}{2} ||U - U^t||_F^2.$$ (14)

According to Theorem 2, the optimal value has closed solution:

$$U^{t+1} = \mathcal{D}_U\left(U^t + \frac{1}{\mu\beta}(\Lambda^{t+\frac{1}{2}} + \frac{\lambda}{\mu\beta^2} A^T B)\right).$$ (15)

Similarly, the last block of variable $F$ updates iteratively with:

$$\min_F \sum_{k=1}^{K} \text{Tr}(F_{k}^T E_k F_{k}) + \langle \Lambda^{t+\frac{1}{2}}, F \rangle + \frac{\mu\beta}{2} ||F - F^t||_F^2,$$

where the scalar $\mu > 2$ [14] and $\hat{F}_{k}$ has a closed form solution:

$$\hat{F}_{k}^{t+1} = (2E_k + \mu\beta I)^{-1}(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})_k,$$ (16)

the subscript $k$ of $(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})_k$ means the $k$-th block corresponding to the $k$-th view. Similarly, each prediction $\hat{F}_{k}^{t+1}$ can be updated within its own view, i.e., each view receives the temporary result $(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})_k$ and combines its feature transformation $E_k$ to compute $\hat{F}_{k}^{t+1}$ in its own view respectively. The view independence of Eq. 15 ensures no interaction among views, which protects the data privacy. After obtained $\hat{F}_{k}^{t+1}$, we also need to project it into feasible domain as in last subsection by $F^{t+1} = \text{Proj}_F(\hat{F})$.

3.4 Solving RANC with ADMM

Considering the diversity between representation among different views, the prediction of classifier on each view may has its own bias. So it is more reasonable to learn an optimal bias for each view, combined with which the last prediction matrix among views can be more comparable. It is notable that the introduced biases here are different from the classifier bias on each view. Let $b \in \mathbb{R}^{K \times 1}$ be the bias vector, where each element is the individual bias for the corresponding view. Together with learned optimal biases, we therefore can assume low rank property for the biased APM. Then the subproblem of Eq. 7 can be further formulated with defined $E_k$:

$$\min_{F \in D} \sum_{k=1}^{K} \text{Tr}(F_{k}^T E_k F_{k}) + \lambda\|F - 1b^T\|_* - \lambda \text{Tr}(A^T B^T)$$ (10)

To solve the problem of Eq. 10, an inequality constraint $U = F - 1b^T$ is further introduced and the problem becomes:

$$\min_{F \in D} \sum_{k=1}^{K} \text{Tr}(F_{k}^T E_k F_{k}) + \lambda\|U\|_* - \lambda \text{Tr}(A^T B^T)$$

s.t. $U = F - 1b^T$.

which can be solved with augmented Lagrange dual form by maximizing the dual variable $\Lambda$:

$$L_{\beta} = \sum_{k=1}^{K} \text{Tr}(F_k^T E_k F_k) + \lambda\|U\|_* - \lambda \text{Tr}(A^T B^T)$$

$$- \langle A, U - (F - 1b^T) \rangle + \frac{\beta}{2} ||U - (F - 1b^T)||_F^2,$$

$\beta > 0$ is a scalar for the augmented term. The problem in Eq. 11 is similar to those problems which can be solved with ADMM [6]. However, here we have three blocks of variables in Eq. 11, namely blocks with $U$, $b$ and $F$, which is with very different properties rather than the ordinary ADMM problems. Using ADMM directly can hardly get converged [29], consequently we employ a variant augmented Lagrange dual optimization technique [14] for our problem, which is a splitting variant of ADMM. After letting $Q = \mathbf{1} b^T$, we can solve an optimal candidate of $Q^{t+1}$ by taking derivative of Eq. 11 w.r.t. $Q$ and set it to zero, thus we have:

$$Q^{t+1} = \frac{1}{\beta} A^T - U^T + F^T.$$ (12)

Then the dual variable can be updated as:

$$\Lambda^{t+\frac{1}{2}} = \Lambda^t + \beta(Q^{t+1} + U^T - F^T).$$ (13)

With the updated $\Lambda$, the remaining variable $U$ can be updated with an added proximal term:

$$\min_U \lambda\|U\|_* - \lambda \text{Tr}(A^T B^T) - \langle \Lambda^{t+\frac{1}{2}}, U \rangle + \frac{\mu\beta}{2} ||U - U^t||_F^2.$$ (14)

According to Theorem 2, the optimal value has closed solution:

$$U^{t+1} = \mathcal{D}_U\left(U^t + \frac{1}{\mu\beta}(\Lambda^{t+\frac{1}{2}} + \frac{\lambda}{\mu\beta^2} A^T B)\right).$$ (15)

Similarly, the last block of variable $F$ updates iteratively with:

$$\min_{F \in D} \sum_{k=1}^{K} \text{Tr}(F_{k}^T E_k F_{k}) + \langle \Lambda^{t+\frac{1}{2}}, F \rangle + \frac{\mu\beta}{2} ||F - F^t||_F^2,$$

where the scalar $\mu > 2$ [14] and $\hat{F}_{k}$ has a closed form solution:

$$\hat{F}_{k}^{t+1} = (2E_k + \mu\beta I)^{-1}(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})_k,$$ (16)

the subscript $k$ of $(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})_k$ means the $k$-th block corresponding to the $k$-th view. Similarly, each prediction $\hat{F}_{k}^{t+1}$ can be updated within its own view, i.e., each view receives the temporary result $(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})_k$ and combines its feature transformation $E_k$ to compute $\hat{F}_{k}^{t+1}$ in its own view respectively. The view independence of Eq. 15 ensures no interaction among views, which protects the data privacy. After obtained $\hat{F}_{k}^{t+1}$, we also need to project it into feasible domain as in last subsection by $F^{t+1} = \text{Proj}_F(\hat{F})$. Update iterations for $U$ and $F$ are separated and consequently can be implemented in a parallel paradigm. After that a renewal of the dual variable should be carried out:

$$\Lambda^{t+1} = \Lambda^{t+\frac{1}{2}} + \beta(U^t - U^{t+1}) - \beta(F^t - F^{t+1}).$$ (16)

Algorithm 3 gives the sketch of this procedure. Following [14][29], the whole procedure can be proved to be converged. 3.5 Rank-One Acceleration

RANC can be solved with APG or ADMM effectively together with truncated nuclear norm regularizer. In this section, an accelerated variation of RANC denoted as RANC$_1$ is proposed to restrict the rank of APM with rank-one update. Recall that in the ideal
Algorithm 3 The pseudo code for solving Eq. 7 with ADMM

Require: $\gamma, \mu$, initialize $F$ using true label matrix;

1: \textbf{while} Stop criterion doesn’t meet \textbf{do}
2: \hspace{0.5em} Solve $Q$ as Eq. 12
3: \hspace{0.5em} Update $A$ as Eq. 13
4: \hspace{0.5em} Solve $U$ as Eq. 14
5: \hspace{0.5em} Solve $\hat{F}$ by Eq. 15 and $F = \text{Proj}_{D}(\hat{F})$
6: \hspace{0.5em} Update $A$ as Eq. 16
7: \textbf{end while}
8: \textbf{return} $F_k$.

Algorithm 4 The pseudo code of RANC

Require: $\gamma, \lambda$, initialize $F$ using true label matrix;

1: \textbf{while} Stop criterion doesn’t meet \textbf{do}
2: \hspace{0.5em} Solve $u$ and $v$ as in Eq. 18 and Eq. 19
3: \hspace{0.5em} Update $F$ using Eq. 19 and $F = \text{Proj}_{D}(\hat{F})$
4: \textbf{end while}
5: \textbf{return} $F_k$.

Table 1: Brief dataset description. Datasets with two views and more than two views are separated with a horizontal line.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>$C$</th>
<th>$n$</th>
<th>$K$</th>
<th>$d_k (k = 1, \ldots, K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course</td>
<td>2</td>
<td>1051</td>
<td>2</td>
<td>66, 5</td>
</tr>
<tr>
<td>Citeseer</td>
<td>6</td>
<td>3264</td>
<td>2</td>
<td>3703, 3264</td>
</tr>
<tr>
<td>Cora</td>
<td>7</td>
<td>2708</td>
<td>2</td>
<td>1433, 2708</td>
</tr>
<tr>
<td>Cornell</td>
<td>5</td>
<td>195</td>
<td>2</td>
<td>1703, 195</td>
</tr>
<tr>
<td>Texas</td>
<td>5</td>
<td>185</td>
<td>2</td>
<td>1703, 185</td>
</tr>
<tr>
<td>Washington</td>
<td>5</td>
<td>217</td>
<td>2</td>
<td>1703, 217</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>5</td>
<td>262</td>
<td>2</td>
<td>1703, 262</td>
</tr>
<tr>
<td>Advertise</td>
<td>2</td>
<td>983</td>
<td>5</td>
<td>457, 495, 472, 111, 19</td>
</tr>
<tr>
<td>News-M5</td>
<td>5</td>
<td>500</td>
<td>3</td>
<td>2000, 2000, 2000</td>
</tr>
<tr>
<td>News-NG1</td>
<td>2</td>
<td>500</td>
<td>3</td>
<td>2000, 2000, 2000</td>
</tr>
<tr>
<td>News-NG2</td>
<td>5</td>
<td>400</td>
<td>3</td>
<td>2000, 2000, 2000</td>
</tr>
<tr>
<td>News-NG3</td>
<td>8</td>
<td>1000</td>
<td>5</td>
<td>2000, 2000, 2000</td>
</tr>
</tbody>
</table>

It is notable that in Eq. 18, only the eigenvector of the largest eigenvalue is needed, which can alleviate the computation burden greatly [3]. In the second step, with the fixed approximate rank-one matrix, we can update $F$ in a closed form. In particular, with the fixed rank-one approximation, we can update the view-specific predictors as:

$$
\tilde{F}_k = \lambda (E_k + \lambda I)^{-1} (uv^\top + 1b_1^\top)k,
$$

where $(uv^\top + 1b_1)k$ gives the $k$th column of matrix $uv^\top + 1b_1^\top$. Each $\tilde{F}_k$ can be updated on each view separately using a similar strategy as aforementioned. $F = \text{Proj}_{D}(\hat{F})$ is executed to project $\hat{F}$ into the feasible domain. The above procedure should be iterated until convergence, which can be summarized in Algorithm 4.

4. EXPERIMENT

We conduct extensive experiments on 15 real-world datasets with multiple views. We first give the general configurations. Then we comprehensively demonstrate effectiveness of our proposed framework (with three variants of solutions) in comparison with the state-of-the-art multi-view learning methods.

4.1 General Experiment Settings

Data used in experiments are consisted of two-view and multiple views (more than 2 views) datasets. Description sketches of datasets are summarized in Table 1. The Course dataset [5] describes web pages and the goal is to predict whether the given web page is a course page or not. The Citeseer dataset [26] is originally made of 4 views, i.e., content, inbound, outbound, cites, on the same documents. We follow [4] to choose the content and cites view in our experiment. In the content view, the documents are characterized by 3703 words. The Cora dataset [26] has the same structure as Citeseer. Following [4] the content view and the cites view are used in our experiment as well. The WebKB dataset [26] contains webpages collected from four universities: Cornell, Texas, Wisconsin and Washington which have 5 categories, i.e., student, project, course, staff and faculty. Data in WebKB are described with two views: content and citation. We treat WebKB in 4 separate datasets grouped by universities. The Advertise dataset [20] [40] has 5 views, i.e., caption and all features in html description together with base url, destination url and image url. Each example describes an image on the web, and the task of the dataset is to determine whether a given image may be an advertisement. The NewsGroup dataset [4] is of 6 groups extracted from the 20-Newsgroup dataset, i.e., M2, M5, M10, NG1, NG2, NG3. Every group contains 10 sample sets, and we choose the first set for all 6
groups in our experiment. There are 3 views in this dataset, which are made by different preprocessing methods for texts, namely using Partitioning Around Medoids, Supervised Mutual Information and Unsupervised Mutual Information [4]. In our experiments, we will denote these types of data as News-M2, News-M5, News-M10, News-NG1, News-NG2 and News-NG3. The Reuters dataset [4] is built from the Reuters RCV1/RCV2 Multilingual test collection, multi-view information is created from different languages, i.e., English, French, German, Italian and Spanish [4].

We run each method 30 times for 15 datasets. 70% of the data are randomly picked up for training and the remaining are for test. In the training set, we randomly choose 30% as the labeled data, and the left 70% as unlabeled ones. Parameters are selected by SCV from \{10^{-5}, 10^{-4}, \ldots, 10^{5}\} in the first split and fixed.

Since RANC leverages advantages of four different types of multi-view learning paradigms, we should compare it with approaches from these four approaches.

### 4.2 Comparing With Fusion Methods

RANC framework is first compared with fusion approaches since it is a hybrid fusion method with advantages of pre-fusion and late fusion. In detail, we compare with five multiple kernel learning (MKL) methods in pre-fusion and the state-of-the-art late fusion method RLF (Robust Late Fusion method) [36]. The MKL methods are Centered Alignment-Based MKL algorithms [9], original SOCP formulated MKL algorithm from [1], Simple MKL method proposed by [25], Group Lasso-based MKL method from [19] and Localized MKL algorithm [10], which are denoted as CABMKL, MKL, SimpleMKL, GLMKL, LMKL in the following contexts and tables respectively. In RLF, we use the best tuned classifier with least square loss as initialized predictor [36]. Furthermore, the ensemble of least square classifier (LS) is listed as a baseline. WNH method [30] which combines all views data together and then uses \(l_2,1\)-norm to perform view selection is also listed as a baseline.

It is notable that fusion methods can output only one classification result for multi-view data, so we compare the integrated result (mean accuracy and std.) of RANC with them. Win/tie/lose counts with t-test at significance level 95% are also recorded in Table 2, where the highest accuracy on each dataset is bolded. Probability voting is used for RANC to obtain the final fusion results. Three variants of RANC solutions in subsection 3.3, 3.4 and 3.5 are denoted as RANC\(_{PG}\), RANC\(_{ADM}\) and RANC\(_{C}\) respectively.

In Table 2, it can be clearly found that RANC gets better results on most datasets. The RANC\(_{ADM}\) returns more stable results. RANC\(_{C}\), the accelerated solver, can also outperform all compared methods from the win/tie/lose counts on most datasets. As to the statistical test results, the RANC framework outperforms fusion methods in most cases, which validates the superiority of RANC. In general, RANC\(_{PG}\) achieves better results on two-view datasets while RANC\(_{ADM}\) performs better on datasets with more than two views. This may be due to the bias term introduced in RANC\(_{ADM}\) when the number of views is large.

To investigate the efficiency of RANC\(_{C}\), we conduct more experiments on a linux cluster with 2.53GHz 12 cores and 48GB memory. The average training time costs (in seconds) of all RANC series methods, late fusion method RLF and a baseline method WNH are recorded in Table 3. Six datasets are picked up for this time costs test. Table 3 evidently verifies the efficiency of RANC\(_{C}\), i.e., 9.85, 65.95, 80.41, 161.52 times faster than RANC\(_{PG}\), RANC\(_{ADM}\), RLF and WNH respectively in average. It is noteworthy that for those compared methods, training time of base classifiers build on each view is not included in RLF, and the implementation of WNH is only with 10 trials. In other words, the superiority of RANC\(_{C}\) on speed can be further enlarged in a fair play.

### 4.3 Comparing with Subspace Approaches

Subspace multi-view learning approaches can provide classification results on individual view. In this section, we compare RANC with multi-view learning method in subspace learning paradigms. WNH [30] is also listed since it can provide predictions on each single view. In this part of experiments, multi-view Linear Discriminant Analysis, multi-view Canonical Correlation Analysis, multi-
Figure 2: Comparisons of classification performance with subspace and WNH methods on all datasets. Each plot shows the cumulative classification accuracy results of each view on certain datasets and different views are plotted in different shadow styles. RANC$_{PG}$ and RANC$_{ADM}$ represent for solving RANC with PGM or ADMM. RANC$_1$ corresponds to the acceleration method.

Table 3: Average training time on 6 datasets (in seconds). Late fusion and baseline methods are compared. Pre-fusion methods, however, highly depend on classifiers invoked, so are not compared.

<table>
<thead>
<tr>
<th>Time</th>
<th>RANC$_1$</th>
<th>RANC$_{PG}$</th>
<th>RANC$_{ADM}$</th>
<th>RLF</th>
<th>WNH</th>
</tr>
</thead>
<tbody>
<tr>
<td>News-M2</td>
<td>1.983</td>
<td>29.913</td>
<td>107.923</td>
<td>146.920</td>
<td>935.517</td>
</tr>
<tr>
<td>News-NG3</td>
<td>20.350</td>
<td>259.153</td>
<td>1398.123</td>
<td>1676.290</td>
<td>1372.167</td>
</tr>
<tr>
<td>Reuters</td>
<td>17.383</td>
<td>222.730</td>
<td>1206.193</td>
<td>803.263</td>
<td>3724.660</td>
</tr>
<tr>
<td>Citeseer</td>
<td>64.870</td>
<td>277.717</td>
<td>4700.440</td>
<td>6494.440</td>
<td>1489.923</td>
</tr>
<tr>
<td>Cora</td>
<td>34.590</td>
<td>201.727</td>
<td>3080.983</td>
<td>4817.303</td>
<td>537.353</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>0.580</td>
<td>2.677</td>
<td>24.157</td>
<td>23.450</td>
<td>102.797</td>
</tr>
</tbody>
</table>

The Cumulative Accuracies Plots (CAP) on each view are shown in Fig. 2 for each compared method. In a CAP, there are $K$ lines. The bottom line in each CAP gives the average accuracy on the first view. While the $i$th line provide the cumulative accuracy from the first view to the $i$th view, i.e., the cumulative accuracies equal to the summation of accuracies on all previous views. Therefore, the top line gives the overall accuracies of all views and the gap (marked with different style of shadows) between any adjacent lines describes the classification accuracy of different views. From the results in Fig. 2, RANC achieves best performance on most datasets, and RANC$_{ADM}$ appears more stable than RANC$_{PG}$ and RANC$_1$. Especially on some text datasets such as News-NG2 or Reuters, all of RANC method are significantly better than other compared methods.

4.4 Disagreement-based Methods Comparison

It is notable that during the training phase, multi-view learning approaches in fusion style and subspace learning style require to repeatedly access to the features from all views, thus those approaches are actually inappropriate for multi-view scenarios with private information channels. RANC and disagreement-based approaches, however, have the ability of boosting performance only with the predictions rather than directly reading the original features in other views.

We compare RANC with multi-view disagreement-based methods. Since most methods of this type are only applicable to two-view scenario, here comparisons on only two-view data are made in this section. Disagreement-based approaches can provide results on each view as well as an ensemble of two view’s results. In this experiment, we conduct comparisons with classical Co-Training [5], CoTrade (Confident Co-Training with data editing) [38] and Co-Label (Co-Regularized Laplacian SVM) [28]. The detailed results of KCCA (Kernel CCA) [13] are also reported. In KCCA, RBF kernel is used with default parameters.

The detailed results are listed in Table 4 where the classical Co-Training is denoted as CoTrain for short. From Table 4, it can be clearly found that RANC series methods achieve the best performance either on view-specific predictions or the final ensemble results. The $t$-test is also performed at 95% significance level, which shows the significant superiorities of RANC framework.

Although both RANC framework and disagreement-based methods can preserve the privacy of different information channels, the interactions of predictions are required in the training stage for both types of methods, which increases the chance of feature exposure from channels. Therefore this type of interactions should also be restricted during the training. To investigate the number of interactions during training, we conduct more experiments on the convergence of RANC and Co-Training with results shown in Fig. 3. Due to the page limits, Fig. 3 gives the convergence plot on two datasets of two-views, i.e., Course and Texas, in one run. From the Fig. 3, it is notable that RANC$_1$ converges rapidly on these datasets, i.e., 5 iterations on Course and Texas for the final result.
Table 4: Classification accuracies (average value ± std.) compared with disagreement-based methods and Kernel CCA on two-view datasets. RANC_PG and RANC_ADM represent for the PGM or ADMM solutions to RANC. RANC_1 denotes for the accelerated RANC version. There are three parts separated with horizontal lines, each of which shows classification results on different views or the final integrated classification results. The last three rows on each sub-table list the win/tie/lose counts on the compared datasets with t-test against other methods at significance level 95%. The best performance on each dataset and each view is bolded.

<table>
<thead>
<tr>
<th>View</th>
<th>dataset</th>
<th>RANC_PG</th>
<th>RANC_ADM</th>
<th>RANC_1</th>
<th>CoTrain</th>
<th>CoTrain</th>
<th>CoLap</th>
<th>KCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>View 1</td>
<td>Course</td>
<td>.864 ± .016</td>
<td>.870 ± .016</td>
<td>.878 ± .014</td>
<td>.754 ± .237</td>
<td>.784 ± .212</td>
<td>.781 ± .000</td>
<td>.882 ± .020</td>
</tr>
<tr>
<td></td>
<td>Citeseer</td>
<td>.687 ± .009</td>
<td>.695 ± .010</td>
<td>.663 ± .020</td>
<td>.208 ± .000</td>
<td>.208 ± .000</td>
<td>.210 ± .001</td>
<td>.236 ± .006</td>
</tr>
<tr>
<td></td>
<td>Cora</td>
<td>.702 ± .013</td>
<td>.703 ± .013</td>
<td>.609 ± .018</td>
<td>.302 ± .000</td>
<td>.302 ± .000</td>
<td>.302 ± .001</td>
<td>.312 ± .004</td>
</tr>
<tr>
<td></td>
<td>Cornell</td>
<td>.724 ± .057</td>
<td>.722 ± .058</td>
<td>.649 ± .065</td>
<td>.419 ± .018</td>
<td>.433 ± .012</td>
<td>.418 ± .000</td>
<td>.418 ± .009</td>
</tr>
<tr>
<td></td>
<td>Texas</td>
<td>.731 ± .052</td>
<td>.733 ± .049</td>
<td>.712 ± .056</td>
<td>.567 ± .012</td>
<td>.570 ± .011</td>
<td>.556 ± .000</td>
<td>.562 ± .009</td>
</tr>
<tr>
<td></td>
<td>Washington</td>
<td>.768 ± .049</td>
<td>.741 ± .051</td>
<td>.741 ± .039</td>
<td>.446 ± .013</td>
<td>.477 ± .012</td>
<td>.473 ± .000</td>
<td>.486 ± .019</td>
</tr>
<tr>
<td></td>
<td>W / T / L</td>
<td>.770 ± .042</td>
<td>.813 ± .031</td>
<td>.729 ± .030</td>
<td>.449 ± .011</td>
<td>.473 ± .017</td>
<td>.445 ± .003</td>
<td>.454 ± .012</td>
</tr>
</tbody>
</table>

| View 2 | Course | .889 ± .014 | .891 ± .014 | .890 ± .015 | .850 ± .028 | .832 ± .122 | .880 ± .015 | .901 ± .014 |
| | Citeseer | .497 ± .016 | .494 ± .015 | .447 ± .038 | .209 ± .000 | .209 ± .000 | .314 ± .022 | .222 ± .026 |
| | Cora | .698 ± .021 | .683 ± .021 | .645 ± .018 | .301 ± .001 | .300 ± .001 | .326 ± .005 | .322 ± .034 |
| | Cornell | .377 ± .066 | .378 ± .067 | .387 ± .068 | .429 ± .010 | .423 ± .007 | .418 ± .000 | .209 ± .057 |
| | Texas | .605 ± .041 | .603 ± .042 | .605 ± .042 | .566 ± .010 | .562 ± .008 | .556 ± .000 | .521 ± .061 |
| | Washington | .623 ± .038 | .639 ± .031 | .638 ± .032 | .480 ± .007 | .480 ± .007 | .478 ± .008 | .411 ± .085 |
| | W / T / L | .421 ± .084 | .492 ± .054 | .487 ± .036 | .449 ± .006 | .449 ± .006 | .448 ± .006 | .362 ± .065 |

| Final | Course | .893 ± .017 | .899 ± .014 | .901 ± .017 | .824 ± .116 | .813 ± .192 | .787 ± .008 | .901 ± .015 |
| | Citeseer | .694 ± .009 | .704 ± .010 | .671 ± .019 | .208 ± .000 | .208 ± .000 | .290 ± .019 | .228 ± .024 |
| | Cora | .784 ± .013 | .785 ± .013 | .653 ± .019 | .302 ± .000 | .302 ± .000 | .302 ± .002 | .334 ± .025 |
| | Cornell | .730 ± .055 | .729 ± .055 | .652 ± .045 | .418 ± .000 | .418 ± .000 | .418 ± .000 | .243 ± .059 |
| | Texas | .738 ± .045 | .737 ± .050 | .730 ± .046 | .588 ± .014 | .567 ± .010 | .556 ± .000 | .539 ± .069 |
| | Washington | .776 ± .033 | .745 ± .050 | .736 ± .041 | .473 ± .000 | .473 ± .000 | .473 ± .000 | .507 ± .068 |
| | W / T / L | .641 ± .061 | .819 ± .031 | .688 ± .044 | .444 ± .000 | .446 ± .006 | .444 ± .000 | .435 ± .039 |

Figure 3: Classification performance vs. number of interactions. RANC_1 V1 and V2 denotes for the changes of classification performance on the first and second view respectively. The same naming scheme is applied to CoTrain.

As to individual views, RANC_1 gets converged after 4 iterations on Course and 12 iterations on Texas. While for Co-Training, the results are unstable in iterations. Furthermore, the performance of RANC_1 is superior to Co-Training to a great extent. It is noteworthy that in each iteration only predictions rather than raw features are exchanged, which preserves the data privacy.

5. CONCLUSION

This paper presents a novel multi-view privacy-preserving framework RANC (RANk Consistency multi-view learning) to boost performance of the predictor constructed on each view by exploiting the relationship among features from multiple private channels. In this scenario, information of one view cannot be shared with others’. We put forward the rank consistency defined on the accumulated prediction matrix (APM) via stacking multi-view predictions, and integrate the rank consistency in a regularizer for improving the classification performance. Properties of RANC suggest employing truncated nuclear norm to control the APM rank into an appropriate range. In our framework, view-specific learner can be enhanced without access to features of other views, therefore the data privacy is well-preserved. Three effective solutions for RANC are provided together including an accelerated variant. Extensive experiments in comparison with the state-of-the-art multi-view approaches are conducted on real datasets, which demonstrates the superiority of RANC in handling multi-view data. Incorporating with different loss functions in RANC framework will be further investigated, and theoretical studies on effects of RANC in multi-view scenario where feature importance varies will also be carried out in future.

Acknowledgments

The authors want to thank reviewers for helpful comments. This work was supported by 973 Program (2014CB340501) and NSFC (61273301, 61333014).
6. REFERENCES


