Rank Consistency based Multi-View Learning: A Privacy-Preserving Approach

Han-Jia Ye, De-Chuan Zhan, Yuan Miao, Yuan Jiang, Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology, Nanjing University
Collaborative Innovation Center of Novel Software Technology and Industrialization
Nanjing, 210023, China
{yehj, zhandc, miaoy, jiangy, zhouzh}@lamda.nju.edu.cn

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

CIKM'15, October 19–23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806552.

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

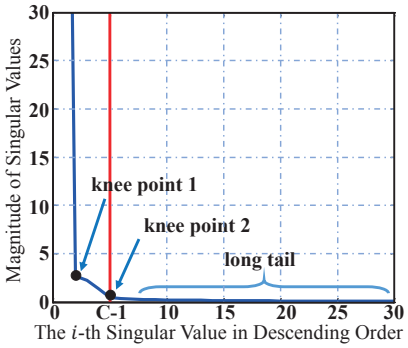


Figure 1: A typical singular value spectrum plot of APM on Reuters (5 views and 6 classes). APM is obtained with linear SVM on each view separately.

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.

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 ith instance $\mathbf{x}_i$ can be represented as a collection of view-specific vectors $\mathbf{x}_{i,k} \in \mathbb{R}^{d_k}$, where d_k is the dimension of the kth view. For labeled examples, instance $\mathbf{x}_i$ has label $y_i \in \{0,1\}$ in binary classification problem. In multi-class cases with C classes, the label y_i for instance $\mathbf{x}_i$ is expanded to a vector with C elements, where $y_{i,j}=1$ indicates the ith instance is with label j, otherwise, $y_{i,j}=0$. The whole data for kth view can be expressed as $X_k = [\mathbf{x}_{1,k}^\top; \mathbf{x}_{2,k}^\top; \cdots; \mathbf{x}_{n,k}^\top] \in \mathbb{R}^{n\times d_k}$, and the corresponding labels can be expressed as $Y \in \{0,1\}^{l\times C}$. Classifier on each view is denoted as $f_k : \mathbf{x}_{i,k} \to \hat{y}$, where $\mathbf{x}_{i,k} \in \mathbb{R}^{d_k}$, $\hat{y} \in \mathbb{R}^C$, and loss of instance i on view k is defined as $\ell(f_k(\mathbf{x}_{i,k}), y_i)$. Then the prediction of X_k on the kth view is combined into $F_k = [f_k(\mathbf{x}_{1,k})^\top; f_k(\mathbf{x}_{2,k})^\top; \cdots; f_k(\mathbf{x}_{n,k})^\top] \in \mathbb{R}^{n\times C}$, and predictions from all views can be stacked into an Accumulated Prediction Matrix (APM), which can be defined as $F = [F_1, F_2, \cdots, F_K] \in \mathbb{R}^{n\times CK}$.

Ideal multi-view classifiers f_k, $k=1,\cdots,K$, 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 $RC(\cdot) : \mathbb{R}^{n\times CK} \to \mathbb{R}$, which defined on the APM F, and we define $RC(F) = \mathrm{rank}(F)$.

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:

$$\min_F \sum_{k=1}^{K} L_k(F_k, Y) + \lambda \mathrm{RC}(F). \tag{1}$$

There are K views and F is the APM. The first term L_k depicts the objective functions according to the property of the kth view. Furthermore, L_i and L_j 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 kth view in Eq. 1 is generally with the form of a regularized empirical loss:

$$\min_{F_k} L_k(F_k, Y) = \ell(F_k, Y) + \gamma r(F_k),$$

here $r(\cdot)$ 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 $W_1, \cdots, W_K$:

$$\begin{aligned} \min_{W_k, b_k, F_k} \quad & \|X_k W_k + \mathbf{1}\boldsymbol{b}_k^\top - F_k\|_F^2 + \gamma \|W_k\|_F^2 \\ s.t. \quad & F_k \in \mathcal{D} : F_k^{1,\cdots,l} = Y,\ 0 \le F_k \le 1. \end{aligned} \tag{2}$$

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 $0 \le F_k \le 1$ to avoid trivial solutions. In Eq. 2, $\boldsymbol{b}_k \in \mathbb{R}^C$ is the bias for current predictor. By centralizing both instances and predictions, the objective can be rewritten as follows:

$$\ell(W_k, F_k) = \|HX_kW_k - HF_k\|_F^2 + \gamma\|W_k\|_F^2, \tag{3}$$

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

$$W_k = (X_k^\top X_k + \gamma I)^{-1} X_k^\top F_k. \tag{4}$$

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:

$$\min_{F_k \in \mathcal{D}} \sum_{k=1}^{K} \mathrm{Tr}(F_k^\top (H - X_k(X_k^\top X_k + \gamma I)^{-1} X_k^\top) F_k) + \lambda \mathrm{RC}(F), \tag{5}$$

here $\mathrm{Tr}(\cdot)$ 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 σ_i, $i = 1, \cdots, \min(m, n)$, 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 $\mathrm{RC}(F)$ 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 $\mathrm{RC}(\cdot)$ 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 $\mathrm{RC}(F)$ 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 $r(r \le min(m, n))$, for any matrix $A \in \mathbb{R}^{r\times m}$ 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:*

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

If the singular value decomposition of matrix X is $X = U\Sigma V^\top$ where Σ 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: $A = (\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_r)^\top$ and $B = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_r)^\top$, corresponds to the first r columns of left and right singular vectors.

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

$$\begin{aligned} F = \quad & \operatorname*{argmin}_{F_k\in\mathcal{D}} \sum_{k=1}^{K} L_k(F_k) + \|F\|_r \qquad (6)\\ = \quad & \operatorname*{argmin}_{F_k\in\mathcal{D}} \sum_{k=1}^{K} L_k(F_k) + \|F\|_* - \max \mathrm{Tr}(AFB^\top),\\ & s.t.\ AA^\top = I,\ BB^\top = I. \end{aligned}$$

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.

Algorithm 1 The pseudo code of RANC

Require: training instances X_k for each view, parameters λ and γ, ϵ, 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.,

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

4: **if** $\|F^{t+1} - F^t\|_F \le \epsilon$ **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.

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 $P_2(F)$:

$$P_1(F) = \sum_{k=1}^{K} L_k(F_k) - \mathrm{Tr}(AFB^\top),\ P_2(F) = \|F\|_* \qquad (8)$$

APG is suitable for solving Eq. 8 [16], which optimizes on a linearized approximation version of the original problem. In the tth 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:

$$\begin{aligned} Q(F) =& P_1(F^t) + \mathrm{Tr}(\langle \nabla P_1(F^t), F - F^t\rangle)\\ &+ \frac{L}{2}\|F - F^t\|_F^2 + P_2(F)\\ =& \sum_{k=1}^{K} L_k(F_k^t) - \mathrm{Tr}(AF^tB^\top) + \mathrm{Tr}(\langle \nabla P_1(F^t), F - F^t\rangle)\\ &+ \frac{L}{2}\|F - F^t\|_F^2 + \lambda\|F\|_*, \end{aligned}$$

where $\nabla P_1(F^t) = 2[E_1F_1^t, E_2F_2^t, \cdots, E_kF_k^t] - \lambda A^\top B$, where $E_k = H - X_k(X_k^\top X_k + \gamma I)^{-1}X_k^\top$, $k = 1, \cdots, K$. 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:

$$\hat{F} = \arg\min_F \lambda\|F\|_* + \frac{L}{2}\|F - (F^t - \frac{1}{L}\nabla P_1(F^t))\|_F^2. \quad (9)$$

Algorithm 2 The pseudo code for solving Eq. 7 with APG

Require: $\gamma, \alpha_1 = 1$, initialize Z^1 and F^1 using true label matrix;
1: **while** Stop criterion doesn't meet **do**
2: Line search for best step-size L
3: $\hat{F}^{t+1} = \mathfrak{D}_L(Z^t),\ F^{t+1} = \text{Proj}_{\mathcal{D}}(\hat{F}^t)$
4: $\alpha_{t+1} = \frac{1+\sqrt{1+4\alpha_t^2}}{2}$
5: $Z^{t+1} = F^{t+1} + (\frac{\alpha_t - 1}{\alpha_{t+1}})(F^{t+1} - F^t)$
6: **end while**
7: **return** F_k.

APG updates using the optimal solution in Eq. 9 at each iteration. Given the following theorem [7] about the proximal operator for nuclear norm:

THEOREM 2. *For each $\tau \geq 0$ and $Y \in \mathbb{R}^{m \times n}$, we have*

$$D_\tau(Y) = \arg\min_X \frac{1}{2}\|X - Y\|_F^2 + \tau\|X\|_*$$

Here, $D_\tau(Y)$ is a matrix shrinkage operator for matrix Y, which can be calculated by SVD of Y. If SVD of Y is $Y = U\Sigma V^T$, then

$$D_\tau(Y) = UD_\tau(\Sigma)V^T, D_\tau(\Sigma) = diag(max(\sigma_i - \tau, 0)).$$

we can solve Eq. 9 in a closed form:

$$\hat{F} = \mathfrak{D}_L(F^t) \stackrel{\text{def}}{=} D_{\frac{\lambda}{L}}(F^t - \frac{1}{L}\nabla P_1(F^t)).$$

Note that the SVD approach in proximal projection is also time efficient, since it will only be applied to a thin matrix. The computation of F acquires the gradient of P_1, blocks of which are constructed using view feature information. Note that the k blocks are view-independent, we can distribute updated view-specific prediction in updated F_k back to each view and compute the $E_k F_k^t$ within the corresponding view. Then only K computation results with the same size of the prediction matrix F_k are returned, through which feature privacy is maintained. After we get $\hat{F}$ from Eq. 9, the feasible F can be obtained by projecting $\hat{F}$ into the $\mathcal{D}$ as in [8], which can be denoted as $\text{Proj}_{\mathcal{D}}(\hat{F})$. As a consequence, we have the RANC framework with APG in Algorithm 2.

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 $\mathbf{b} \in \mathbb{R}^{KC \times 1}$ be the biases 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:

$$\min_{F \in \mathcal{D}} \sum_{k=1}^{K} \text{Tr}(F_k^\top E_k F_k) + \lambda\|F - \mathbf{1}\mathbf{b}^\top\|_* - \lambda\text{Tr}(A[F - \mathbf{1}\mathbf{b}^\top]B^\top). \tag{10}$$

To solve the problem of Eq. 10, an equality constraint $U = F - \mathbf{1}\mathbf{b}^\top$ is further introduced and the problem becomes:

$$\min_F \sum_{k=1}^{K} \text{Tr}(F_k^\top E_k F_k) + \lambda\|U\|_* - \lambda\text{Tr}(AUB^\top)$$
$$s.t.\ U = F - \mathbf{1}\mathbf{b}^\top,$$

which can be solved with augmented Lagrange dual form by maximizing the dual variable Λ:

$$L_\beta = \sum_{k=1}^{K} \text{Tr}(F_k^\top E_k F_k) + \lambda\|U\|_* - \lambda\text{Tr}(AUB^\top) \tag{11}$$
$$- \langle \Lambda, U - (F - \mathbf{1}\mathbf{b}^\top)\rangle + \frac{\beta}{2}\|U - (F - \mathbf{1}\mathbf{b}^\top)\|_F^2,$$

$\beta > 0$ 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, $\mathbf{b}$ 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 $Q = \mathbf{1}\mathbf{b}^\top$, 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:

$$Q^{t+1} = \frac{1}{\beta}\Lambda^t - U^t + F^t. \tag{12}$$

Then the dual variable can be updated as:

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

With the updated Λ, the remaining variable U can be updated with an added proximal term:

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

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

$$U^{t+1} = D_{\frac{\lambda}{\mu\beta}}(U^t + \frac{1}{\mu\beta}\Lambda^{t+\frac{1}{2}} + \frac{\lambda}{\mu\beta}A^\top B). \tag{14}$$

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

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

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

$$\hat{F}_k^{t+1} = (2E_k + \mu\beta I)^{-1}(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})_k, \tag{15}$$

the subscript k of $(\mu\beta F^t - \Lambda^{t+\frac{1}{2}})$ means the kth block corresponding to the kth 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 their 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 $F^{t+1} = \text{Proj}_{\mathcal{D}}(\hat{F})$. 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:

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

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: γ, μ, initialize F using true label matrix;
1: **while** Stop criterion doesn't meet **do**
2: Solve Q as Eq. 12
3: Update Λ as Eq. 13
4: Solve U as Eq. 14
5: Solve $\hat{F}$ by Eq. 15 and $F = \mathrm{Proj}_{\mathcal{D}}(\hat{F})$
6: Update Λ as Eq. 16
7: **end while**
8: **return** F_k.

case, APM in C-class problem should be with rank $C-1$, i.e., for binary problems, rank of APM should be restrained to one, which provides facilitation for designing efficient rank-one approximation approach to RANC.

To simplify the discussion, we demonstrate the implementation in binary case where each view only gets a vector prediction value output, i.e., $F_k \in \mathbb{R}^{n\times 1}$ and the APM $F \in \mathbb{R}^{n\times K}$. For multi-class problem, the classification can be carried out with one-vs-rest strategies. The rank-one acceleration is brought forward based on the following property: $\mathrm{rank}(X) = 1$, where $X \in \mathbb{R}^{m\times n}$, if and only if there are two vectors $\mathbf{u} \in \mathbb{R}^{m\times 1}$ and $\mathbf{v} \in \mathbb{R}^{n\times 1}$ that X can be decomposed into the outer product of $\mathbf{u}$ and $\mathbf{v}$, i.e., $X = \mathbf{u}\mathbf{v}^\top$.

Consequently, we can reduce the rank consistency of APM to one for binary case simply by using two vectors to approximate APM, i.e., we define $RC(F) = \|F - \mathbf{u}\mathbf{v}^\top\|_F^2$, where the error is estimated using Frobenius norm. Note that there are diversities between different views, we add biases $\mathbf{b} \in \mathbb{R}^K$ for view predictors to further facilitate the rank reduction of APM, i.e., we can redefine $RC(F) = \|F - \mathbf{1}\mathbf{b}^\top - \mathbf{u}\mathbf{v}^\top\|_F^2$. Unique solution of $\mathbf{u}$ and $\mathbf{v}$ can be obtained by restricting $\mathbf{u}$ and $\mathbf{v}$ orthogonal. We therefore can reformulate the original problem in Eq. 5 as follows:

$$\min_{F\in\mathcal{D},\mathbf{u}^\top\mathbf{u}=1,\mathbf{v}^\top\mathbf{v}=1} \sum_{k=1}^{K} L_k(F_k) + \lambda\|F - \mathbf{1}\mathbf{b}^\top - \mathbf{u}\mathbf{v}^\top\|_F^2. \quad (17)$$

The rank-one approximation formulation in Eq. 17 can be solved by alternative optimization. We first fix F for solving $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{b}$, i.e., solving the following problem:

$$\min_{\mathbf{u}^\top\mathbf{u}=1,\mathbf{v}^\top\mathbf{v}=1} \|F - \mathbf{1}\mathbf{b}^\top - \mathbf{u}\mathbf{v}^\top\|_F^2.$$

Bias term $\mathbf{b}$, which aims at finding an optimal mean for individual prediction, can be solved as [23], i.e., $\mathbf{b}$ can be solved by $\mathbf{b} = F^\top\mathbf{1}/\mathbf{n}$, and then we can solve $\mathbf{u}$ and $\mathbf{v}$ with eigenvalue decomposition:

$$\begin{aligned} \mathbf{u} &= \arg\max_{\mathbf{u}^\top\mathbf{u}=1} \mathrm{Tr}(\mathbf{u}^\top HFF^\top H\mathbf{u}), \quad (18)\\ \mathbf{v} &= (F^\top - \mathbf{b}\mathbf{1}^\top)\mathbf{u}. \end{aligned}$$

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:

$$\hat{F}_k = \lambda(E_k + \lambda I)^{-1}(\mathbf{u}\mathbf{v}^\top + \mathbf{1}\mathbf{b}^\top)_k, \quad (19)$$

where $(\mathbf{u}\mathbf{v}^\top + \mathbf{1}\mathbf{b}^\top)_k$ gives the kth column of matrix $\mathbf{u}\mathbf{v}^\top + \mathbf{1}\mathbf{b}^\top$. Each $\hat{F}_k$ can be updated on each view separately using a similar strategy as aforementioned. $F = \mathrm{Proj}_{\mathcal{D}}(\hat{F})$ 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.

Algorithm 4 The pseudo code of RANC_1

Require: γ, λ, initialize F using true label matrix;
1: **while** Stop criterion doesn't meet **do**
2: Solve $\mathbf{u}$ and $\mathbf{v}$ as in Eq. 18 and Eq. 19
3: Update F using Eq. 19 and $F = \mathrm{Proj}_{\mathcal{D}}(\hat{F})$
4: **end while**
5: **return** F_k.

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

Datasets	C	n	K	d_k $(k = 1, \cdots, K)$
Course	2	1051	2	66, 5
Citeseer	6	3264	2	3703, 3264
Cora	7	2708	2	1433, 2708
Cornell	5	195	2	1703, 195
Texas	5	185	2	1703, 185
Washington	5	217	2	1703, 217
Wisconsin	5	262	2	1703, 262
Advertise	2	983	5	457, 495, 472, 111, 19
News-M2	2	1200	3	2000, 2000, 2000
News-M5	5	500	3	2000, 2000, 2000
News-M10	10	500	3	2000, 2000, 2000
News-NG1	2	500	3	2000, 2000, 2000
News-NG2	5	400	3	2000, 2000, 2000
News-NG3	8	1000	3	2000, 2000, 2000
Reuters	6	1600	5	2000, 2000, 2000, 2000, 2000

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, stuff 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 alt 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

Table 2: Classification accuracies (average value ± std.) compared with fusion and baseline methods. RANC_{PG} and RANC_{ADM} represent for the APG or ADMM solutions to RANC. RANC_1 denotes for the accelerated RANC version. Last three rows list the win/tie/lose counts on all datasets with t-test against other methods at significance level 95%. The best performance on each dataset is bolded.

	Our Methods			Pre-Fusion Methods					Late Fusion	Baseline	
Dataset	RANC_{PG}	RANC_{ADM}	RANC_1	CABMKL	MKL	SimpleMKL	GLMKL	LMKL	RLF	WNH	LS
Course	.893±.017	.899±.014	**.901±.017**	**.901±.018**	.893±.015	.893±.015	.898±.018	.834±.030	.866±.015	.875±.035	.887±.016
Citeseer	.694±.009	.704±.010	.671±.019	.692±.011	.689±.011	.685±.012	**.707±.008**	.681±.011	.693±.010	.650±.013	.694±.010
Cora	.784±.013	**.785±.013**	.653±.019	.748±.015	.733±.022	.731±.020	.762±.014	.675±.018	.782±.014	.648±.017	.716±.015
Cornell	**.730±.055**	.729±.055	.652±.045	.629±.068	.621±.069	.620±.067	.600±.055	.641±.055	.624±.060	.529±.070	.689±.056
Texas	**.738±.045**	.737±.050	.730±.046	.557±.006	.556±.003	.556±.003	.558±.007	.556±.000	.658±.038	.591±.082	.701±.047
Washington	**.776±.033**	.745±.050	.736±.041	.684±.031	.675±.034	.674±.033	.686±.027	.719±.029	.655±.032	.691±.058	.731±.047
Wisconsin	.641±.061	**.819±.031**	.688±.044	.727±.030	.727±.036	.725±.038	.702±.029	.740±.027	.637±.052	.719±.050	.545±.057
Advertise	.898±.020	.887±.024	.880±.096	.861±.003	.860±.000	.860±.000	.862±.004	.860±.000	.805±.028	**.918±.028**	.786±.080
News-M2	.963±.015	**.974±.013**	.963±.016	.849±.069	.779±.050	.779±.050	.930±.025	.818±.064	.964±.012	.700±.208	.963±.015
News-M5	.918±.033	.921±.019	.903±.024	.864±.033	.881±.030	.877±.028	.912±.025	.802±.036	**.924±.021**	.820±.145	.869±.023
News-M10	.788±.023	.787±.023	**.791±.026**	.656±.051	.667±.034	.657±.033	.756±.026	.555±.030	.741±.026	.700±.072	.735±.026
News-NG1	**.936±.031**	**.936±.032**	.906±.033	.904±.041	.865±.090	.865±.090	.901±.042	.879±.046	.928±.025	.894±.112	.885±.026
News-NG2	**.928±.012**	**.928±.012**	.922±.015	.896±.027	.908±.019	.907±.021	**.928±.015**	.862±.013	.922±.014	.461±.178	.875±.016
News-NG3	.923±.012	.923±.012	**.928±.010**	.910±.017	.887±.018	.887±.020	.926±.013	.852±.016	.911±.010	.385±.046	.884±.012
Reuters	.696±.022	.706±.020	**.715±.021**	.682±.023	.680±.024	.679±.022	.682±.018	.659±.022	.706±.019	.559±.028	.633±.025
W / T / L	RANC_{PG} vs. others			13 / 1 / 1	14 / 0 / 1	14 / 0 / 1	10 / 3 / 2	14 / 0 / 1	8 / 6 / 1	13 / 0 / 2	14 / 1 / 0
W / T / L	RANC_{ADM} vs. others			15 / 0 / 0	15 / 0 / 0	15 / 0 / 0	11 / 3 / 1	15 / 0 / 0	11 / 3 / 1	14 / 0 / 1	14 / 1 / 0
W / T / L	RANC_1 vs. others			9 / 3 / 3	11 / 1 / 3	11 / 1 / 3	7 / 6 / 2	9 / 3 / 3	9 / 2 / 4	11 / 3 / 1	11 / 1 / 3

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 5CV from $\{10^{-5}, 10^{-4}, \cdots, 10^{5}\}$ in the first split and fixed.

Since RANC leverages advantages of above 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 5 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_1 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_1, 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} works when the number of views is large.

To investigate the efficiency of RANC_1, 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_1, 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_1 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-

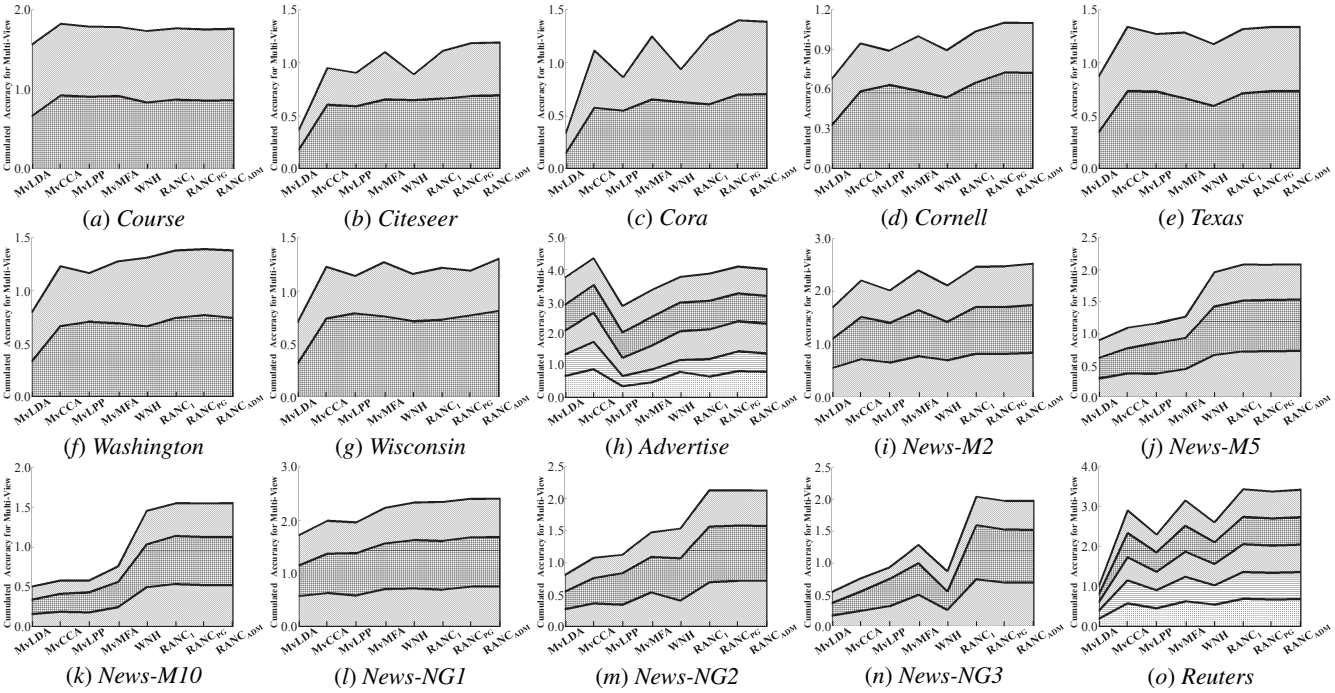


Figure 2: Comparisons of classification performance with subspace and WNH methods on all datasets. Each plot shows the cumulative classification accuracy results of each view on certain datasets and different views are plotted in different shadow styles. RANC_{PG} and RANC_{ADM} represent for solving 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.

Time	RANC_1	RANC_{PG}	RANC_{ADM}	RLF	WNH
News-M2	1.983	29.913	107.923	146.920	935.517
News-NG3	20.350	259.153	1398.123	1676.290	1372.167
Reuters	17.383	222.730	1206.193	803.263	3724.660
Citeseer	64.870	277.717	4700.440	6494.440	1489.923
Cora	34.590	201.727	3080.983	4817.303	537.353
Wisconsin	0.580	2.677	24.157	23.450	102.797

view Locality Preserving Projections, multi-view Marginal Fisher Analysis (which are denoted as MV-LDA, MV-CCA, MV-LPP and MV-MFA) [27] are compared.

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 kth line provide the cumulative accuracy from the first view to the kth 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.

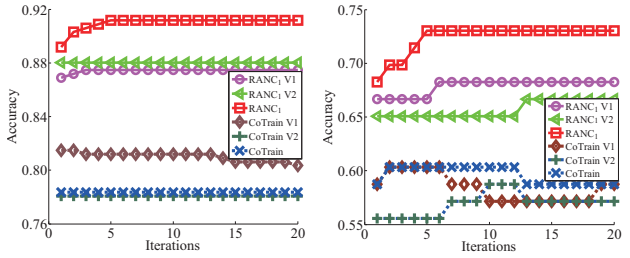
We compare RANC with multi-view disagreement-based methods. Since most methods of this type are only applicable to two-view scenario, here comparisons on only two-view data are made in this section. Disagreement-based approaches can provide results on each view as well as an ensemble of two view's results. In this experiment, we conduct comparisons with classical Co-Training [5], CoTrade (Confident Co-Training with data editing) [38] and Co-Lap (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_1 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.

View	dataset	RANC_{PG}	RANC_{ADM}	RANC_1	CoTrain	CoTrade	CoLap	KCCA
View 1	Course	.864±.016	.870±.016	**.878±.014**	.754±.237	.784±.212	.781±.000	.882±.020
	Citeseer	.687±.009	**.695±.010**	.663±.020	.208±.000	.208±.000	.210±.001	.236±.006
	Cora	.702±.013	**.703±.013**	.609±.018	.302±.000	.302±.000	.302±.001	.312±.004
	Cornell	**.724±.057**	.722±.058	.649±.065	.419±.018	.433±.012	.418±.000	.418±.009
	Texas	.731±.052	**.733±.049**	.712±.056	.567±.012	.570±.011	.556±.000	.562±.009
	Washington	**.768±.049**	.741±.051	.741±.039	.446±.013	.477±.012	.473±.000	.486±.019
	Wisonsin	.770±.042	**.813±.031**	.729±.030	.449±.011	.473±.017	.445±.003	.454±.012
	W / T / L	RANC_{PG} vs. others			7 / 0 / 0	7 / 0 / 0	7 / 0 / 0	6 / 0 / 1
	W / T / L	RANC_{ADM} vs. others			7 / 0 / 0	7 / 0 / 0	7 / 0 / 0	6 / 0 / 1
	W / T / L	RANC_1 vs. others			7 / 0 / 0	7 / 0 / 0	7 / 0 / 0	6 / 1 / 0
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
	Wisonsin	.421±.084	**.492±.054**	.487±.036	.449±.006	.449±.006	.448±.006	.362±.065
	W / T / L	RANC_{PG} vs. others			5 / 1 / 1	5 / 1 / 1	5 / 1 / 1	6 / 0 / 1
	W / T / L	RANC_{ADM} vs. others			6 / 0 / 1	6 / 0 / 1	6 / 0 / 1	6 / 0 / 1
	W / T / L	RANC_1 vs. others			6 / 0 / 1	6 / 0 / 1	6 / 0 / 1	6 / 0 / 1
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	.305±.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
	Wisonsin	.641±.061	**.819±.031**	.688±.044	.444±.000	.446±.006	.444±.000	.435±.039
	W / T / L	RANC_{PG} vs. others			7 / 0 / 0	7 / 0 / 0	7 / 0 / 0	6 / 0 / 1
	W / T / L	RANC_{ADM} vs. others			7 / 0 / 0	7 / 0 / 0	7 / 0 / 0	6 / 1 / 0
	W / T / L	RANC_1 vs. others			7 / 0 / 0	7 / 0 / 0	7 / 0 / 0	6 / 1 / 0



(a) *Course* (b) *Texas*

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

[1] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, pages 6–13, Banff, Canada, 2004.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] P. Berkhin. A survey on pagerank computing. *Internet Mathematics*, 2(1):73–120, 2005.

[4] G. Bisson and C. Grimal. Co-clustering of multi-view datasets: A parallelizable approach. In *Proceedings of the IEEE 12th International Conference on Data Mining*, pages 828–833, Brussels, Belgium, 2012.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, Madison, WI., 1998.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[7] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[8] X. Chang, F. Nie, Y. Yang, and H. Huang. A convex formulation for semi-supervised multi-label feature selection. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1171–1177, Quebec, Canada, 2014.

[9] C. Cortes, M. Mohri, and A. Rostamizadeh. Two-stage learning kernel algorithms. In *Proceedings of the 27th International Conference on Machine Learning*, pages 239–246, Haifa, Israel, 2010.

[10] M. Gönen and E. Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 352–359, Helsinki, Finland, 2008.

[11] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[12] Z. Harchaoui, M. Douze, M. Paulin, M. Dudik, and J. Malick. Large-scale image classification with trace-norm regularization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3386–3393, Providence, RI., 2012.

[13] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[14] B. He, M. Tao, and X. Yuan. A splitting method for separable convex programming. *IMA Journal of Numerical Analysis*, 35(1):394–426, 2015.

[15] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, 2013.

[16] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th International Conference on Machine Learning*, pages 457–464, Montreal, Canada, 2009.

[17] X. Jin, F. Zhuang, H. Xiong, C. Du, P. Luo, and Q. He. Multi-task multi-view learning for heterogeneous tasks. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 441–450, Shanghai, China, 2014.

[18] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen. Multi-view discriminant analysis. In *Proceedings of the 12th European Conference on Computer Vision*, pages 808–821, Florence, Italy, 2012.

[19] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Non-sparse regularization and efficient training with multiple kernels. *arxiv preprint arXiv:1003.0079*, 2010.

[20] N. Kushmerick. Learning to remove internet advertisements. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 175–181, Seattle, WA., 1999.

[21] X. Li, J. Gao, H. Li, L. Yang, and R. K. Srihari. A multimodal framework for unsupervised feature fusion. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 897–902, Burlingame, CA., 2013.

[22] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l2,1-norm minimization. In *Proceedings of the 25th conference on Uncertainty in Artificial Intelligence*, pages 339–348, Montreal, Canada, 2009.

[23] F. Nie, J. Yuan, and H. Huang. Optimal mean robust principal component analysis. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1062–1070, Beijing, China, 2014.

[24] M. Qian and C. Zhai. Unsupervised feature selection for multi-view clustering on text-image web news data. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 1963–1966, Shanghai, China, 2014.

[25] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[26] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[27] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2160–2167, Providence, RI., 2012.

[28] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML workshop on Learning With Multiple Views*, pages 74–79, Bonn, Germany, 2005.

[29] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

[30] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. In *Proceedings of the 30th International Conference on Machine Learning*, pages 352–360, Atlanta, GA., 2013.

[31] W. Wang and Z.-H. Zhou. Multi-view active learning in the non-realizable case. In *Advances in Neural Information Processing Systems 23*, pages 2388–2396. Cambridge, MA.: MIT Press, 2010.

[32] W. Wang and Z.-H. Zhou. A new analysis of co-training. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1135–1142, Haifa, Israel, 2010.

[33] W. Wang and Z.-H. Zhou. Co-training with insufficient views. In *Proceedings of the 5th Asian Conference on Machine Learning*, pages 467–482, Canberra, Australia, 2013.

[34] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[35] Y. Yang, C. Lan, X. Li, B. Luo, and J. Huan. Automatic social circle detection using multi-view clustering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 1019–1028, Shanghai, China, 2014.

[36] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3021–3028, Providence, RI., 2012.

[37] D. Zhai, H. Chang, S. Shan, X. Chen, and W. Gao. Multiview metric learning with global consistency and local smoothness. *ACM Transactions on Intelligent Systems and Technology*, 3(3):Article 53, 2012.

[38] M.-L. Zhang and Z.-H. Zhou. Cotrade: Confident co-training with data editing. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(6):1612–1626, 2011.

[39] Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3):415–439, 2010.

[40] Z.-H. Zhou, D.-C. Zhan, and Q. Yang. Semi-supervised learning with very few labeled training examples. In *Proceedings of the 22nd AAAI conference on Artificial Intelligence*, pages 675–680, Vancouver, Canada, 2007.