Cross-Domain Feature Learning in Multimedia

Xiaoshan Yang, Tianzhu Zhang, *Member, IEEE*, and Changsheng Xu, *Fellow, IEEE*

***Abstract*—In the Web 2.0 era, a huge number of media data, such as text, image/video, and social interaction information, have been generated on the social media sites (e.g., Facebook, Google, Flickr, and YouTube). These media data can be effectively adopted for many applications (e.g., image/video annotation, image/video retrieval, and event classification) in multimedia. However, it is difficult to design an effective feature representation to describe these data because they have multi-modal property (e.g., text, image, video, and audio) and multi-domain property (e.g., Flickr, Google, and YouTube). To deal with these issues, we propose a novel cross-domain feature learning (CDFL) algorithm based on stacked denoising auto-encoders. By introducing the modal correlation constraint and the cross-domain constraint in conventional auto-encoder, our CDFL can maximize the correlations among different modalities and extract domain invariant semantic features simultaneously. To evaluate our CDFL algorithm, we apply it to three important applications: sentiment classification, spam filtering, and event classification. Comprehensive evaluations demonstrate the encouraging performance of the proposed approach.**

***Index Terms*—Cross-domain, deep learning, feature learning, multi-modal.**

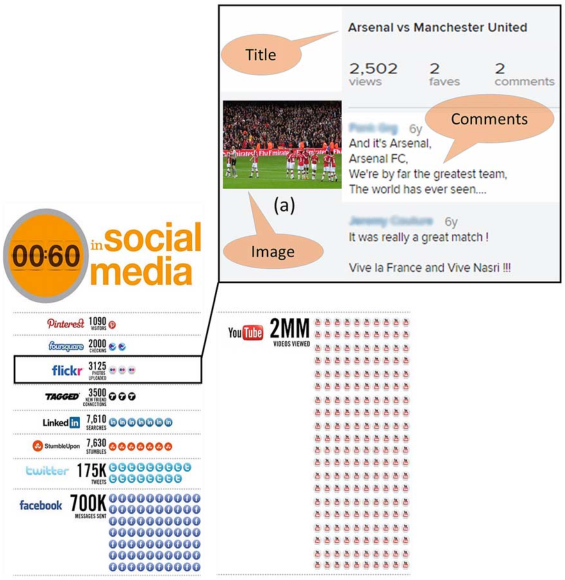
Title
Arsenal vs Manchester United
2,502 views
2 faves
2 comments
Comments
6y
And it's Arsenal,
Arsenal FC,
We're by far the greatest team,
The world has ever seen....
(a)
Image
6y
It was really a great match !
Vive la France and Vive Nasri !!!
00:60 in social media
Pinterest 1090 VISITORS
foursquare 2000 CHECKINS
flickr 3125 PHOTOS UPLOADED
TAGGED 3500 NEW FRIEND CONNECTIONS
LinkedIn 7,610 SEARCHES
StumbleUpon 7,630 STUMBLES
twitter 175K TWEETS
facebook 700K MESSAGES SENT
YouTube 2MM VIDEOS VIEWED

Fig. 1. Details about the speed of media generation on several social websites. The data have both multi-domain property (e.g., Flickr, Pinterest, and Facebook) and multi-modal property (e.g., images, videos, and texts) (Image (a) via Flickr under Creative Commons License). (https://www.flickr.com/photos/wonker/3015441676/)

I. Introduction

With the rapid development of Web 2.0, there are more and more social media sites (e.g., Flickr, YouTube, Facebook, and Google) for people to capture and share media data online. As a result, what is happening around us and around the world can spread very fast, and there are substantial amounts of media data with multi-modal features (e.g., images, videos, and texts) in the Internet. As in Fig. 1, we show the growing amounts of media data generated on several social media sites in 60 seconds.[1] Based on this figure, we can see that all these media data are not only large in scale, but also various in both modality (e.g., photos, videos, and texts) and domain (e.g., Flickr, Pinterest, and Facebook). For example, as shown in Fig. 1, the uploaded photos on Flickr always have the corresponding titles, users' preference and comments. Besides, photos about the football players also appear on other social media sites (domains), such as Pinterest and Facebook.

The huge volume of existing multimedia data contains useful information and has been adopted for many applications. For example, real-time media data have been utilized in semantic video indexing, event prediction, image/video context annotation [1]. Streaming data coming from platforms such as YouTube and Twitter have been utilized in multimedia applications, clustering-based video retrieval [2] and socialized video recommendation [3]. Data from Flickr have been used in personalized search and recommendation [4], predicting the winner of the 2008 United Sates president election and monitoring the product distribution in the world [5]. Facial expressions in photos are explored to measure the public opinion during the election period [6]. In [7], social relationship information is adopted to improve user-level sentiment analysis. In [8]–[15], social media data are adopted for event detection and classification.

In the above different applications, to make use of the multimedia data, one of the most important problems is how to learn effective features. Most of the existing applications use the metadata, such as time, location and descriptions, as features. For example, in [5], the number of photos uploaded in a

Manuscript received February 11, 2014; revised June 18, 2014 and September 04, 2014; accepted November 08, 2014. Date of publication December 02, 2014; date of current version December 23, 2014. This work was supported in part by the National Program on Key Basic Research Project under the 973 Program, Project No. 2012CB316304, the National Natural Science Foundation of China under Grants 61225009, 61303173, 61373122, and 61432019, and the Beijing Natural Science Foundation under Grant 4131004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. K. Selcuk Candan.

The authors are with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the China-Singapore Institute of Digital Media, Singapore 119613, Singapore (e-mail: xiaoshang.yang@nlpr.ia.ac.cn; tzzhang@nlpr.ia.ac.cn; csxu@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2014.2375793

[1] http://imeco.nl/onze-diensten/social-media/social-media-infographics

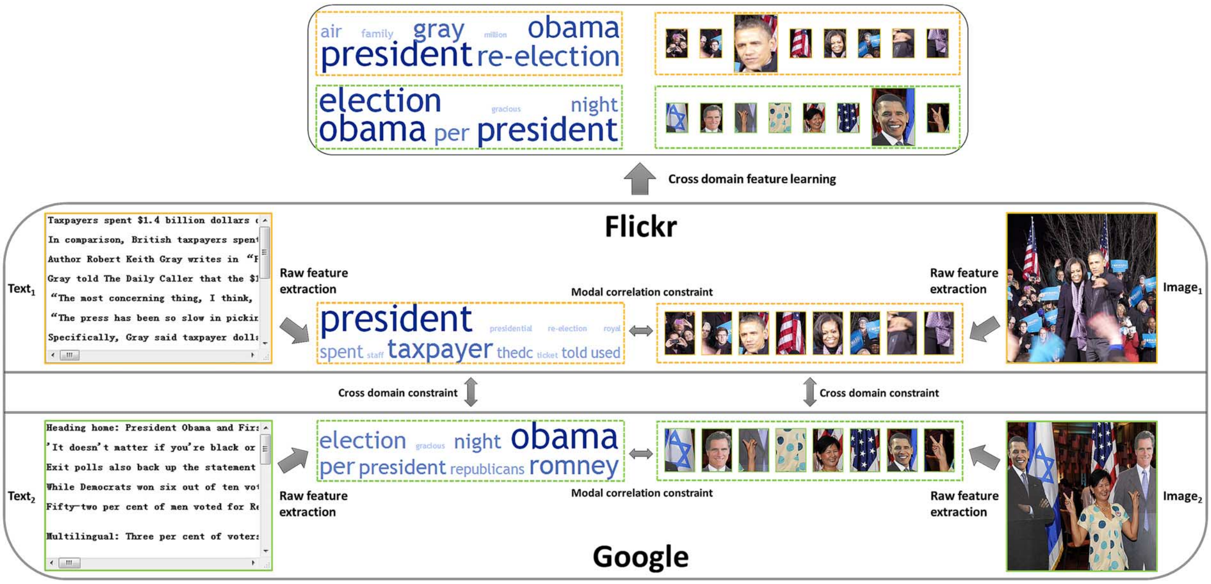
air family gray million obama
president re-election
election gracious night
obama per president
Cross domain feature learning
Flickr
Text$_1$
Taxpayers spent $1.4 billion dollars
In comparison, British taxpayers spent
Author Robert Keith Gray writes in "F
Gray told The Daily Caller that the $1
"The most concerning thing, I think,
"The press has been so slow in pickir
Specifically, Gray said taxpayer doll
Raw feature extraction
Modal correlation constraint
president presidential re-election royal
spent staff taxpayer thedc ticket told used
Raw feature extraction
Image$_1$
Cross domain constraint
Cross domain constraint
Text$_2$
Heading home: President Obama and Fir
'It doesn't matter if you're black or
Exit polls also back up the statement
While Democrats won six out of ten vot
Fifty-two per cent of men voted for Re
Multilingual: Three per cent of voter
election gracious night obama
per president republicans romney
Raw feature extraction
Modal correlation constraint
Raw feature extraction
Image$_2$
Google

Fig. 2. Illustration of the key idea of our CDFL. For simplicity, we only show two event instances in two domains (Flickr and Google) with two modalities (text and image). Text_1 and Image_1 are from the Flickr domain. Text_2 and Image_2 are from the Google domain (Image_1 and Image_2 under Creative Commons License). (https://www.flickr.com/photos/jneider/8459372838 and https://www.flickr.com/photos/usembassyta/8168917239)

fixed time duration is used to predict the election winner. In [3], only text descriptions of the video are used for recommendation. These metadata are easy to be extracted. However, they may not be available in some data samples and cannot be obtained as features. To deal with this issue, we can make use of the content of the multimedia data to learn effective features. For social media analysis, it is a fundamental problem to learn such features to represent the semantic information in the data. However, there are three main challenges: 1) The media data have multi-modal property. As shown in Fig. 1, a data sample in social media sites is typically described with images, texts and other metadata simultaneously. 2) The media data have multi-domain property. As shown in Fig. 1, the data may come from different domains (e.g., Flickr, YouTube, and Facebook), and they can compensate each other, but also have domain discrepancy. For example, if Google News is regarded as one domain and Flickr as the other domain, there are more noises on the images from Flickr, such as the stylized face images. Besides, images on Google are mostly captured by journalists while images on Flickr are typically captured by users, who may upload images which do not focus on the targets of a specific event perfectly. This kind of domain discrepancy is similar to the Office dataset and the multi-view action dataset used for domain adaptation [16]–[18]. 3) The traditional manually designed low level features for some modalities, such as images, texts, and videos, still cannot effectively represent their semantic information.

In this paper, to address the above challenges for feature learning in multimedia, we propose a novel Cross-domain Feature Learning (CDFL) approach based on marginalized denoising auto-encoders. In our CDFL: (1) To make use of the multi-modal property, we propose a modal correlation constraint to maximize the correlations among different modalities of media data. (2) To deal with the domain discrepancy, we adopt a cross-domain constraint to learn the domain invariant features and make different domains share a common feature space. (3) To avoid using manually designed low level features, we adopt a data driven method to consider the data distribution and learn more abstract and semantic features by deep learning. It has been proven that stacked layer-wise features are much more effective than the conventional shallow models [19]–[21]. The key idea of our CDFL algorithm is shown in Fig. 2. For simplicity, we only show two domains (Flickr and Google) with two modalities (Text and Image) related to event "2012 United States Election". For the Flickr domain, text and image features are extracted from user uploaded photos and their descriptions. For the Google domain, text and image features are extracted from webpages returned from Google search engine. Most of these webpages are news articles accompanied with pictures. (Text_1, Image_1) are text and image data for an event instance on the Flickr domain, and (Text_2, Image_2) are text and image data for the same event on the Google domain. Though these two data pairs are both about the US president election, they contain some specific domain features. For example, the words "taxpayer" and "spent" are highlighted in the extracted raw features for Text_1, and the words "obama" and "romney" are highlighted for Text_2. From the results, we can see that the highlighted features for the same event are different in the two domains. After cross-domain feature learning, the domain invariant features like "election"/"re-election" and "president" are highlighted as features. In this way, data in different domains can have similar features. For the other modality (image), the same cross-domain feature learning scheme can be used to obtain domain invariant features, such as the Barack Obama's faces, in the images. As a result, due to the modal correlation constraint and the cross-domain constraint, our CDFL can learn domain invariant features for cross-domain multi-modal data analysis.

To evaluate the proposed CDFL algorithm, we test it in three different applications in multimedia: sentiment classification, spam filtering, and event classification. The first two applications are special cases of our CDFL, and they are used to verify the performance for single modal features. The third application considers event classification using media data from different domains with multiple modalities. Because there are no available event datasets for this task, we construct one with 11 events by crawling images and the corresponding texts from Flickr and Google. The experimental results on the three different applications demonstrate the effectiveness of our CDFL. Compared with the existing methods, our contributions in this work are threefold:

- To consider the multi-modal property, we introduce a modal correlation constraint in conventional denoising auto-encoders by maximizing the correlations among different modalities of media data.
- To reduce the domain discrepancy among multiple domains, we introduce a cross-domain constraint in single denoising autoencoders by use of maximum mean discrepancy (MMD).
- To verify our CDFL, we evaluate it on three different applications in multimedia and demonstrate that it achieves much better performance than existing methods. Besides, we collect a dataset for research on event classification with multi-modal information in multiple domains, and will release it for academic use.

The remainder of this paper is organized as follows. In Section II, we review the related work. Section III introduces the formulation of our CDFL algorithm. In Section IV, we show how to solve the optimization problem. In Section V, we show and analyze the experimental results for all three applications. The conclusion is presented in Section VI.

II. Related Work

In this section, we briefly introduce existing methods which are most related to our CDFL algorithm, including multi-modal feature learning, multi-domain feature learning, and deep feature representation.

A. Multi-Modal Feature Learning

To deal with the multiple modalities of features, there are mainly three different kinds of learning methods.

1) Feature Subspace: The most widely used feature subspace method for multiple modalities is the canonical correlation analysis (CCA) [22]–[24]. The CCA can be seen as the problem of finding basis vectors for variables with different modalities. Thus, the correlation between the projected vectors of the variables along the basis vectors is mutually maximized. The basis vectors are decided by a set of linear transformations, one for each modality of the variables.

2) Semantic Integration: In [25], the query-by-example paradigm is extended to the semantic domain. A semantic feature space where each image is represented by the vector of posterior concept probabilities is defined. In [24], the semantic representation for each image is constructed based on the correlation space where the original features are mapped using CCA. The correlation between two modalities based on CCA and the semantic representation based on multi-class logistic regression are combined in this method.

3) Kernel Method: In [26], a semi-supervised learning approach is proposed to leverage the information contained in the tags associated with unlabeled images. The multiple kernel learning (MKL) framework is used to combine a kernel based on the image content with a second kernel which encodes the tags associated with each image.

Though these methods perform well on some problems like image retrieval, most of them can deal with multi-modal data coming from only a single domain and only few of them can deal with multi-modal data coming from different domains.

B. Multi-Domain Feature Learning

In multi-domain feature learning, most of the existing methods are designed for improving the classification accuracies for unlabeled instances in the target domain by leveraging on the labeled instances in the source domain. In [27], a structural correspondence learning method is proposed to induce correspondence among features from two domains by modeling their relations with pivot features that appear frequently in both domains. The techniques in [28] reduce the distance across two domains by learning a latent feature space where domain similarity is measured through maximum mean discrepancy. In [29], [30], the source and target domains are linked by sampling finite or infinite number of intermediate subspaces on the Grassmannian manifold. In [31], the boosting scheme is applied by optimizing the source classification error and margin constraints over the unlabeled target instances.

In multimedia community, there are also some algorithms proposed for improving the learning task in the target domain by leveraging on the source domain. A knowledge adaptation method for Ad Hoc multimedia event detection is proposed in [32]. In [33], cross-domain correlation knowledge is used for web multimedia object classification. In [34], a feature transformation method is proposed to indirectly transfer semantic knowledge between text and images. In [3], the authors use a graph based framework to model the distribution discrepancy problem between the social and the video domains. However, most of these multi-domain feature learning methods do not consider the multi-modality of data.

C. Deep Feature Representation

Deep learning can be considered as a particular kind of representation learning procedure that discovers multiple levels of representation, with higher-level features representing more abstract aspects of the data [35]. The earliest approaches that have been used to reduce the initialization sensitivity of the deep compositions of non-linearities are based on greedy layer-wise pre-training [19]–[21]. With unsupervised pre-training, each layer is trained by RBM [36], [19] or denoising auto-encoders [37] to model the distribution of values which are output of the previous layer.

As one of the building blocks of deep learning, the denoising auto-encoder [37] has been used in cross-domain feature learning [38]. The key idea behind this method is that the union of the samples from both source and target domains can be used to train a common feature representation. In [39], the denoising auto-encoder is simplified as a single linear denoiser for cross-domain feature learning where hidden nodes are discarded. This change greatly improves the efficiency for training of the original auto-encoder. Recently, there are also some deep learning methods proposed for multi-modal data. In [40], the audio and video features are considered simultaneously through autoencoders. For the speech classification on noise data source, the videos of the mouth, as a complement for audios, can obviously improve the recognition performance. However, these methods do not consider the domain discrepancy problem.

Though the above methods perform well in various fields, such as computer vision and natural language processing, they are not easy to be adopted in multimedia where the domain discrepancy is large and the media data typically have different kinds of modalities.

III. Cross-Domain Feature Learning

In this section, we introduce the details of our CDFL algorithm for multimedia data. Here, for simplicity, only media data with two modalities, image and text, are adopted. Note that, other types of media data can also be applied in our CDFL.

A. Problem Definition

Our goal is to learn a shared feature representation for instances contained in two different domains in an unsupervised way. These instances considered in this paper are either single modal or multi-modal, such as product reviews, spam e-mail messages or events described by images and texts. The feature representation is determined by stacked linear denoising auto-encoders (denoisers) in our CDFL algorithm. Denoising auto-encoders are adopted to reconstruct image features and text features from their corrupted features. Moreover, the modal correlation and cross-domain constraints are considered in the reconstruction process simultaneously. Practically, we need to compute the mapping matrices of auto-encoders for image and text features, respectively.

Let $X_s = [x_1, \ldots, x_{n_1}]$ and $Y_s = [y_1, \ldots, y_{n_1}]$ denote the image and text features of n_1 instances in the source domain. Similarly, let $X_t = [x_{n_1+1}, \ldots, x_n]$ and $Y_t = [y_{n_1+1}, \ldots, y_n]$ be image and text features of n_2 instances in the target domain, where $n_2 = n - n_1$. Furthermore, we denote image and text features of n instances in two domains as $X = [X_s, X_t]$, $Y = [Y_s, Y_t]$. Let $\tilde{x}$ and $\tilde{y}$ denote the corrupted version of x and y by adding noises. Besides, $\tilde{X}$ and $\tilde{Y}$ are the corrupted version of X and Y, respectively.

B. Formulation

To learn a common feature space for instances in the source domain and target domain, we adopt the single-layer denoiser. Linear mapping matrices need to be computed in conventional single-layer denoiser. Here, we want to construct two mapping matrices W_x and W_y for image and text in each layer, respectively. The errors for the two kinds of feature reconstruction determined by W_x and W_y are denoted with $\mathcal{L}_x$ and $\mathcal{L}_y$. Our final objective function is shown in Eq. (1), where modal correlation constraint $\mathcal{L}_{mc}$ and cross-domain constraint $\mathcal{L}_{cd}$ are adopted. λ_x, λ_y, λ_{mc}, and λ_{cd} are four regularization parameters which control the weights of $\mathcal{L}_x$, $\mathcal{L}_y$, $\mathcal{L}_{mc}$, and $\mathcal{L}_{cd}$, respectively.

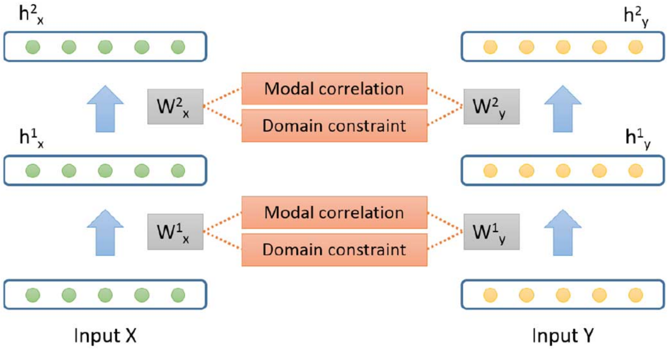

Fig. 3. Illustration of the layer-wise structure of our CDFL. Multiple denoisers with the modal correlation and cross-domain constraints are stacked together. X denotes image features, and Y denotes text features.

$$\arg \min_{W_x, W_y} \lambda_x \mathcal{L}_x + \lambda_y \mathcal{L}_y + \lambda_{mc} \mathcal{L}_{mc} + \lambda_{cd} \mathcal{L}_{cd} \tag{1}$$

The modal correlation constraint $\mathcal{L}_{mc}$ and cross-domain constraint $\mathcal{L}_{cd}$ in Eq. (1) assure that the learned W_x and W_y will map the original feature into a feature space, where the modal correlation is maximized and domain discrepancy is minimized. In the remainder of this paper, we denote $\mathcal{L} = \lambda_x \mathcal{L}_x + \lambda_y \mathcal{L}_y + \lambda_{mc} \mathcal{L}_{mc} + \lambda_{cd} \mathcal{L}_{cd}$. The details of $\mathcal{L}_x$ and $\mathcal{L}_y$ are illustrated in Section III-B1. $\mathcal{L}_{mc}$ and $\mathcal{L}_{cd}$ are explained in Section III-B2 and III-B3, respectively.

After learning the single layer features by solving the optimization problem (1), we stack several single layer denoisers by feeding the output of the $(t-1)$ th denoiser as the input into the t th denoiser. The training processes for W_x and W_y are performed greedily layer by layer, and the output of the t th denoiser is denoted as $h_x^t = tanh(W_x^t h_x^{t-1})$, $h_y^t = tanh(W_y^t h_y^{t-1})$. Here, the nonlinear function $tanh(.)$ is not contained in the optimization, thus solving W_x and W_y is extremely fast for a single layer denoiser. An example of 2 layer stacked denoisers is shown in Fig. 3.

1) Corrupted Feature Reconstruction: The corrupted feature reconstruction formulations for image and text are shown in Eq. (2) and Eq. (3), respectively.

$$\mathcal{L}_x = \sum_{i=1}^{n} \|x_i - W_x \tilde{x}_i\|_2^2 = Tr\left[\left(X - W_x \tilde{X}\right)^\top \left(X - W_x \tilde{X}\right)\right] \tag{2}$$

$$\mathcal{L}_y = \sum_{i=1}^{n} \|y_i - W_y \tilde{y}_i\|_2^2 = Tr\left[\left(Y - W_y \tilde{Y}\right)^\top \left(Y - W_y \tilde{Y}\right)\right]. \tag{3}$$

Here, the image or text features from both the source domain and the target domain are combined together. This scheme is beneficial to learn the domain invariant features as discussed in [39].

2) Modal Correlation Constraint: To maximize the correlations among different feature modalities, inspired by the canonical correlation analysis (CCA) [22]–[24], we propose to minimize the following objective function:

$$\begin{aligned}\mathcal{L}_{mc} = & Tr\left(W_x C_{xx} W_x^\top\right) + Tr\left(W_y C_{yy} W_y^\top\right) \\ & - 2Tr\left(W_x C_{xy} W_y^\top\right)\end{aligned} \tag{4}$$

where $C_{xx} = \tilde{X}\tilde{X}^\top$ and $C_{yy} = \tilde{Y}\tilde{Y}^\top$ are covariance matrices of image and text, respectively. The $C_{xy} = \tilde{X}\tilde{Y}^\top$ is the cross-covariance matrix between image and text. Compared with the formulation of CCA as in [22]–[24], the advantages of our objective function (4) are discussed in Section III-C2.

3) Cross-Domain Constraint: In conventional deep learning based domain adaptation methods [38], [39], the samples in source and target domains are combined together to train the auto-encoders. Here, to further reduce the distribution discrepancy between the source domain and the target domain, we introduce the maximum mean discrepancy (MMD) [41] in each single layer of auto-encoder. The definition is shown in Eq. (5).

$$\begin{aligned}\mathcal{L}_{cd} = & \left\| \frac{1}{n_1}\sum_{i=1}^{n_1} \phi\left(W_x \tilde{x}_i\right) - \frac{1}{n_2}\sum_{i=n_1+1}^{n} \phi\left(W_x \tilde{x}_i\right) \right\|_2^2 \\ & + \left\| \frac{1}{n_1}\sum_{i=1}^{n_1} \phi\left(W_y \tilde{y}_i\right) - \frac{1}{n_2}\sum_{i=n_1+1}^{n} \phi\left(W_y \tilde{y}_i\right) \right\|_2^2 \\ = & \frac{1}{n_1^2}\sum_{i,j=1}^{n_1} \left(K(W_x\tilde{x}_i, W_x\tilde{x}_j) + K(W_y\tilde{y}_i, W_y\tilde{y}_j)\right) \\ & - \frac{2}{n_1 n_2}\sum_{\substack{i=1,\\ j=n_1+1}}^{n_1,n} \left(K(W_x\tilde{x}_i, W_x\tilde{x}_j) + K(W_y\tilde{y}_i, W_y\tilde{y}_j)\right) \\ & + \frac{1}{n_2^2}\sum_{i,j=n_1+1}^{n} \left(K(W_x\tilde{x}_i, W_x\tilde{x}_j) + K(W_y\tilde{y}_i, W_y\tilde{y}_j)\right)\end{aligned} \tag{5}$$

where $\phi(\cdot)$ is a mapping function which can be expressed in terms of a kernel function $K(\cdot,\cdot) = (\phi(\cdot), \phi(\cdot))$. Strictly speaking, the kernel function $K(\cdot,\cdot)$ should be universal (e.g., Gaussian) so that the domain discrepancy can be measured by the expectations of the samples in the source domain and the target domain in a reproducing kernel Hilbert spaces (RKHS) [41], [42]. However, practically the kernel in MMD can be relaxed to a more general case, such as the polynomial kernel, which also presents comparable results to the Gaussian kernel [43]. Because the linear function is a special case of the polynomial function, it's also reasonable to use the linear kernel as a replacement of the Gaussian kernel. Though there has been no formal justification for utilizing a MMD with linear kernel, the effectiveness has been demonstrated in [44]. Besides, in our experiments, the linear kernel is even more effective when marginalization is considered. Details of the cross-domain constraints using Gaussian and linear kernels are illustrated as follows.

Gaussian kernel: Here, we adopt the Gaussian kernel function $K_g(x_1, x_2) = exp(-\|x_1 - x_2\|^2/\sigma)$, which is known to be universal [45]. We denote $\mathcal{L}_{cd}$ with a Gaussian kernel as $\mathcal{L}_{cd}^g$. Then, Eq. (5) could be rewritten as

$$\begin{aligned}\mathcal{L}_{cd}^g = & \frac{1}{n_1^2}\sum_{i,j=1}^{n_1} \left(K_g(W_x\tilde{x}_i, W_x\tilde{x}_j) + K_g(W_y\tilde{y}_i, W_y\tilde{y}_j)\right) \\ & - \frac{2}{n_1 n_2}\sum_{\substack{i=1,\\ j=n_1+1}}^{n_1,n} \left(K_g(W_x\tilde{x}_i, W_x\tilde{x}_j) + K_g(W_y\tilde{y}_i, W_y\tilde{y}_j)\right) \\ & + \frac{1}{n_2^2}\sum_{i,j=n_1+1}^{n} \left(K_g(W_x\tilde{x}_i, W_x\tilde{x}_j) + K_g(W_y\tilde{y}_i, W_y\tilde{y}_j)\right)\end{aligned} \tag{6}$$

where $K_g(W_x\tilde{x}_i, W_x\tilde{x}_j) = \exp(-\frac{(\tilde{x}_i-\tilde{x}_j)^\top W_x^\top W_x(\tilde{x}_i-\tilde{x}_j)}{\delta})$ and $K_g(W_y\tilde{y}_i, W_y\tilde{y}_j) = \exp(\frac{-(\tilde{y}_i-\tilde{y}_j)^\top W_y^\top W_y(\tilde{y}_i-\tilde{y}_j)}{\delta})$.

Linear kernel: Replacing the kernel function $K(\cdot,\cdot)$ in Eq. (5) with a linear kernel, we obtain $K_l(W_x x_1, W_x x_2) = x_1^\top W_x^\top W_x x_2$, $K_l(W_y y_1, W_y y_2) = y_1^\top W_y^\top W_y^\top y_2$. We denote $\mathcal{L}_{cd}$ with a linear kernel as $\mathcal{L}_{cd}^l$. Then, Eq. (5) could be rewritten as

$$\mathcal{L}_{cd}^l = G_x^\top W_x^\top W_x G_x + G_y^\top W_y^\top W_y G_y \tag{7}$$

where $G_x = (\frac{1}{n_1}\sum_{i=1}^{n_1}\tilde{x}_i - \frac{1}{n_2}\sum_{i=n_1+1}^{n}\tilde{x}_i)$ and $G_y = (\frac{1}{n_1}\sum_{i=1}^{n_1}\tilde{y}_i - \frac{1}{n_2}\sum_{i=n_1+1}^{n}\tilde{y}_i)$.

C. Discussion

Our CDFL is a novel denoising auto-encoder with both multi-modal and multi-domain properties as shown in Eq. (1). Here, $\mathcal{L}_x$ and $\mathcal{L}_y$ denote the feature reconstruction errors for image and text, respectively. $\mathcal{L}_{mc}$ and $\mathcal{L}_{cd}$ model the modal correlation and the cross-domain constraints, respectively. With different settings of the four parameters λ_x, λ_y, λ_{mc} and λ_{cd}, our CDFL algorithm has different properties. (a) If we set λ_x (or λ_y) and λ_{mc} to zeros, our CDFL is similar to the conventional cross-domain classification algorithm for single modal features like mSDA. In this case, to evaluate our CDFL, we test it on two applications (sentiment classification and spam filtering) with single-modal features as introduced in Sections V-A and V-B. (b) If λ_x, λ_y, λ_{mc}, and λ_{cd} are non-zeros, our CDFL is appropriate for cross-domain multi-modal data analysis, and we evaluate it for event classification as discussed in Section V-C.

1) Comparison Between Our CDFL and mSDA: Based on Eq. (2) and Eq. (3), we can see that both our CDFL algorithm and mSDA [39] adopt the denoising auto-encoders as the building block for unsupervised cross-domain feature learning. Their differences are as follows. (a) In mSDA, only single modal features are adopted. However, our CDFL can be used for multi-modal features. To achieve this goal, we introduce the modal correlation constraint($\mathcal{L}_{mc}$) in the denoising auto-encoders to consider the correlations among multi-modal features. (b) In mSDA, the instances from source domain and target domain are simply merged together to train the denoising auto-encoders. In our CDFL, we adopt Maximum Mean Discrepancy (MMD) to reduce the domain discrepancy. Thus, for single modal features, our CDFL is an improved mSDA by considering the cross-domain constraint ($\mathcal{L}_{cd}$).

2) Comparison Between Our CDFL and Conventional Multi-Modal Methods: Traditional methods for multi-modal classification or cross-modal retrieval mostly adopt shallow model, such as CCA, to map features in different modalities into a common feature space [25]–[26]. Different from these methods, our CDFL makes use of an transformation of the CCA, the modal correlation constraint shown in Eq. (4), to model the correlations among different modalities. Furthermore, in our CDFL, the correlations among different modalities are considered in each layer of the auto-encoder, and the stacked auto-encoders in our CDFL can be viewed as a deep correlation maximization model. Thus, our CDFL can learn more abstract and semantic representations for multi-modal features compared with the existing methods.

IV. Optimization

In this section, we introduce how to solve the optimization problem in Eq. (1). Since two different kinds of kernel functions [Eqs. (6) and (7)] are considered for the cross-domain constraint $\mathcal{L}_{cd}$ in Eq. (1), we need to solve these two cases of Eq. (1) separately. For the Gaussian kernel adopted in $\mathcal{L}^g_{cd}$, we need to solve

$$\arg\min_{W_x,W_y} \mathcal{L}_g \tag{8}$$

where $\mathcal{L}_g = \lambda_x\mathcal{L}_x + \lambda_y\mathcal{L}_y + \lambda_{mc}\mathcal{L}_{mc} + \lambda_{cd}\mathcal{L}^g_{cd}$. For the linear kernel adopted in $\mathcal{L}^l_{cd}$, we need to solve

$$\arg\min_{W_x,W_y} \mathcal{L}_l \tag{9}$$

where $\mathcal{L}_l = \lambda_x\mathcal{L}_x + \lambda_y\mathcal{L}_y + \lambda_{mc}\mathcal{L}_{mc} + \lambda_{cd}\mathcal{L}^l_{cd}$. Next, we will introduce how to solve those two optimization problems.

A. Solving Eq. (8)

For the Gaussian kernel in the cross-domain constraint, we make use of conjugate gradients which is a popular nonlinear optimization method with fast convergence rate. Mapping matrix Wx or Wy is iteratively computed with the other one fixed. To solve Wx and Wy, we need to compute the gradient of $\mathcal{L}_g$ with respect to Wx and Wy.

$$\frac{\partial\mathcal{L}_g}{\partial Wx} = \lambda_x\frac{\partial\mathcal{L}_x}{\partial Wx} + \lambda_{mc}\frac{\partial\mathcal{L}_{mc}}{\partial Wx} + \lambda_{cd}\frac{\partial\mathcal{L}^g_{cd}}{\partial Wx} \tag{10}$$

$$\frac{\partial\mathcal{L}_g}{\partial Wy} = \lambda_y\frac{\partial\mathcal{L}_y}{\partial Wy} + \lambda_{mc}\frac{\partial\mathcal{L}_{mc}}{\partial Wy} + \lambda_{cd}\frac{\partial\mathcal{L}^g_{cd}}{\partial Wy} \tag{11}$$

The partial gradients of $\mathcal{L}_x$, $\mathcal{L}_y$, $\mathcal{L}_{mc}$ are shown in Appendix B. Partial Gradients of $\mathcal{L}^g_{cd}$ are computed as follows.

$$\begin{aligned}\frac{\partial\mathcal{L}^g_{cd}}{\partial W_x} = &-\frac{2}{\sigma n_1^2}\sum_{i=1,j=1}^{n1} GK_g(\tilde{x}_i,\tilde{x}_j)\\ &+\frac{4}{\sigma n_1 n_2}\sum_{\substack{i=1,\\ j=n_1+1}}^{n_1,n} GK_g(\tilde{x}_i,\tilde{x}_j)\\ &-\frac{2}{\sigma n_2^2}\sum_{i,j=n_1+1}^{n} GK_g(\tilde{x}_i,\tilde{x}_j)\end{aligned} \tag{12}$$

where $GK_g(\tilde{x}_i,\tilde{x}_j) = \frac{\partial K_g(\tilde{x}_i,\tilde{x}_j)}{\partial W_x} = K_g(\tilde{x}_i,\tilde{x}_j)W_x(\tilde{x}_i - \tilde{x}_j)(\tilde{x}_i - \tilde{x}_j)^\top$,

$$\begin{aligned}\frac{\partial\mathcal{L}^g_{cd}}{\partial W_y} = &-\frac{2}{\sigma n_1^2}\sum_{i=1,j=1}^{n1} GK_g(\tilde{y}_i,\tilde{y}_j)\\ &+\frac{4}{\sigma n_1 n_2}\sum_{\substack{i=1,\\ j=n_1+1}}^{n_1,n} GK_g(\tilde{y}_i,\tilde{y}_j)\\ &-\frac{2}{\sigma n_2^2}\sum_{i,j=n_1+1}^{n} GK_g(\tilde{y}_i,\tilde{y}_j)\end{aligned} \tag{13}$$

where $GK_g(\tilde{y}_i,\tilde{y}_j) = \frac{\partial K_g(\tilde{y}_i,\tilde{y}_j)}{\partial W_y} = K_g(\tilde{y}_i,\tilde{y}_j)W_y(\tilde{y}_i - \tilde{y}_j)(\tilde{y}_i - \tilde{y}_j)^\top$.

As discussed in [43], the resulting mapping functions used within nonlinear kernel of the MMD probably only capture the noise of both domains if we directly adopt the preceding gradient descent method. As a result, it will decrease the performance dramatically. Similar to [43], we also add orthogonal constraints for Wx and Wy to prevent our model wrongly matching the two distributions. Then, we adopt the Manopt [46], a Matlab toolbox for optimization on manifolds, to solve the problem shown in Eq. (8). Here, Wx and Wy are optimized alternatively with one of them fixed. We observe that learning Wx and Wy typically converges quite fast (less than 10 iterations).

B. Solving Eq. (9)

1) Closed Form Solution: If we use the linear kernel in the cross-domain constraint, the four items $\mathcal{L}_x$, $\mathcal{L}_y$, $\mathcal{L}_{mc}$ and $\mathcal{L}^l_{cd}$ are all convex (the convexities of $\mathcal{L}_x$, $\mathcal{L}_y$ and $\mathcal{L}^l_{cd}$ are obvious, and the convexity of $\mathcal{L}_{mc}$ will be explained in the Appendix A). In this case, Eq. (9) represents a convex quadratic programming problem. The global optimal solution can be obtained by finding the points where the partial gradients of $\mathcal{L}^l = \mathcal{L}_x + \mathcal{L}_y + \mathcal{L}_{mc} + \mathcal{L}^l_{cd}$ are equal to zeroes. The partial gradients for all these four items are shown in Appendix B.

The partial derivative of $\mathcal{L}^l$ according to W_x is computed as

$$\frac{\partial\mathcal{L}^l}{\partial W_x} = W_xQ_x - 2\lambda_cW_yC_{xy}^\top - 2\lambda_x\bar{C}_{xx}, \tag{14}$$

where $C_{xx} = \tilde{X}\tilde{X}^\top$, $\bar{C}_{xx} = X(\tilde{X})^\top$ and $Q_x = 2(\lambda_x + \lambda_c)C_{xx} + \lambda_mG_xG_x^\top)$. Similarly, we get the partial derivative of $\mathcal{L}^l$ according to W_y

$$\frac{\partial\mathcal{L}^l}{\partial W_y} = W_yQ_y - 2\lambda_cW_xC_{xy} - 2\lambda_y\bar{C}_{yy}, \tag{15}$$

where $C_{yy} = \tilde{Y}\tilde{Y}^\top$, $\bar{C}_{yy} = Y(\tilde{Y})^\top$ and $Q_y = 2((\lambda_y + \lambda_c)C_{yy} + \lambda_mG_yG_y^\top)$.

Let Eq. (14) and Eq. (15) be zeroes, we get a linear equation array for matrices W_x and W_y. If Q_x and

$(Q_y - 4\lambda_c^2 C_{xy}^\top Q_x^{-1} C_{xy})$ are invertible matrices (this condition is always satisfied in practice), we obtain the solution of W_y as follows:

$$W_y = 2\left(2\lambda_c\lambda_x\bar{C}_{xx}Q_x^{-1}C_{xy} + \lambda_y\bar{C}_{yy}\right) \times \left(Q_y - 4\lambda_c^2 C_{xy}^\top Q_x^{-1} C_{xy}\right)^{-1}. \tag{16}$$

Once we get W_y, we can compute W_x according to the following equation:

$$W_x = 2\left(\lambda_c W_y C_{xy}^\top + \lambda_x \bar{C}_{xx}\right) Q_x^{-1}. \tag{17}$$

If the inverse matrices of Q_x and $(Q_y - 4\lambda_c^2 C_{xy}^\top Q_x^{-1} C_{xy})$ do not exist, we can use the corresponding pseudo-inverse matrix to compute the approximate solutions of W_x and W_y.

2) Marginalization: One serious issue with auto-encoder is that only using the reconstruction error constraint may potentially lead to learn an identity function [47]. To prevent this issue, the strategy adopted by denoising auto-encoder is to reconstruct the inputs from their stochastically corrupted features [37]. The stochastic corruption process consists of randomly setting some of the inputs to zero. Hence, the denosing auto-encoder is trying to predict the missing values from the non-missing values. The corruptions are useful for capturing the statistical dependencies between the inputs [47]. As in [39], the solutions in Section IV-B1 are based on corrupted input features. The more corruptions we average over, the more effective and robust the learned mapping W_x and W_y will be. Here, we introduce how to marginalize infinitely many times random noise through the weak law of large numbers [39].

Let $q = [1-p, \ldots, 1-p, 1]^\top$ be a probability vector, where p is the corruption probability and q_α represents the probability of a feature value with subscript α "surviving" the corruption. Following, we will explain how to compute the marginalized version of C_{xx}, $\bar{C}_{xx}$, C_{yy}, $\bar{C}_{yy}$, C_{xy}, G_x and G_y which are used for the closed form solution in Section IV-B1.

Let $S = XX^\top$, α and β are used to denote the matrix row and column subscripts, respectively. Then we get the following marginalized expressions of C_{xx} and $\bar{C}_{xx}$:

$$E(C_{xx}) = \sum_{i=1}^{n} E[\tilde{x}_i\tilde{x}_i^\top] \tag{18}$$

$$E(C_{xx})_{\alpha,\beta} = \begin{cases} S_{\alpha\beta}q_\alpha q_\beta & \textbf{if} \quad \alpha \neq \beta \\ S_{\alpha\beta}q_\alpha & \textbf{if} \quad \alpha = \beta \end{cases} \tag{19}$$

$$E(\bar{C}_{xx}) = \sum_{i=1}^{n} E[x_i\tilde{x}_i^\top], E(\bar{C}_{xx})_{\alpha,\beta} = S_{\alpha\beta}q_\beta \tag{20}$$

Let $R = YY^\top$, then we obtain the following marginalized expressions of C_{yy} and $\bar{C}_{yy}$:

$$E(C_{yy}) = \sum_{i=1}^{n} E[\tilde{y}_i\tilde{y}_i^\top] \tag{21}$$

$$E(C_{yy})_{\alpha,\beta} = \begin{cases} R_{\alpha\beta}q_\alpha q_\beta & \textbf{if} \quad \alpha \neq \beta \\ R_{\alpha\beta}q_\alpha & \textbf{if} \quad \alpha = \beta \end{cases} \tag{22}$$

$$E(\bar{C}_{yy}) = \sum_{i=1}^{n} E[y_i\tilde{y}_i^\top], E(\bar{C}_{yy})_{\alpha,\beta} = R_{\alpha\beta}q_\beta \tag{23}$$

Let $U = XY^\top$, then we get the following marginalized expression of C_{xy}:

$$E(C_{xy}) = \sum_{i=1}^{n} E[\tilde{x}_i\tilde{y}_i^\top] \tag{24}$$

$$E(C_{xy})_{\alpha,\beta} = \begin{cases} U_{\alpha\beta}q_\alpha q_\beta & \textbf{if} \quad \alpha \neq \beta \\ U_{\alpha\beta}q_\alpha & \textbf{if} \quad \alpha = \beta \end{cases} \tag{25}$$

Let

$$V = \left(\frac{1}{n_1}\sum_{i=1}^{n_1} x_i - \frac{1}{n_2}\sum_{i=n_1+1}^{n} x_i\right)$$
$$Z = \left(\frac{1}{n_1}\sum_{i=1}^{n_1} y_i - \frac{1}{n_2}\sum_{i=n_1+1}^{n} y_i\right)$$

we obtain the following marginalized vectors of G_x and G_y:

$$E(G_x) = \left(\frac{1}{n_1}\sum_{i=1}^{n_1} E(\tilde{x}_i) - \frac{1}{n_2}\sum_{i=n_1+1}^{n} E(\tilde{x}_i)\right)$$
$$E(G_y) = \left(\frac{1}{n_1}\sum_{i=1}^{n_1} E(\tilde{y}_i) - \frac{1}{n_2}\sum_{i=n_1+1}^{n} E(\tilde{y}_i)\right) \tag{26}$$
$$E(G_x)_\alpha = V_\alpha q_\alpha, E(G_y)_\alpha = Z_\alpha q_\alpha \tag{27}$$

Thus, we get the final marginalized solutions as shown in Eq. (28) and Eq. (29) by replacing all matrices in Eq. (16) and Eq. (17) with their marginalized versions.

$$W_y = 2(2\lambda_c\lambda_x E(\bar{C}_{xx})E(Q_x)^{-1}E(C_{xy}) + \lambda_y E(\bar{C}_{yy})) \times (E(Q_y) - 4\lambda_c^2 E(C_{xy})^\top E(Q_x)^{-1}E(C_{xy}))^{-1} \tag{28}$$

$$W_x = 2\left(\lambda_c W_y E(C_{xy})^\top + \lambda_x E(\bar{C}_{xx})\right) E(Q_x)^{-1} \tag{29}$$

where

$$E(Q_x) = 2\left((\lambda_x + \lambda_c)E(C_{xx}) + \lambda_m E(G_x)E(G_x)^\top\right),$$
$$E(Q_y) = 2\left((\lambda_y + \lambda_c)E(C_{yy}) + \lambda_m E(G_y)E(G_y)^\top\right).$$

V. Experiments

In this section, we show the experimental results of our CDFL algorithm on three different applications: sentiment classification, spam filtering and event classification. The first two applications are for single modal features while the third application is for multi-modal features. The three applications can demonstrate the usefulness of our CDFL algorithm for feature extraction from multimedia data.

To better evaluate the domain transfer power of our method, we compare the CDFL algorithm with recent related methods, such as *GFK* [29], *Landmark* [30] and *mSDA* [39]. For our cross-domain feature learning method, we implemented three versions. We use *CDFL-g* to denote the algorithm with Gaussian kernel as shown in Eq. (8). We set the δ of the Gaussian kernel to be the median squared distance between all source examples as in [43]. The algorithm with linear kernel as shown in Eq. (9) is denoted as *CDFL* which is solved in closed form

TABLE I
ACCURACY ON EACH TASK AND THEIR AVERAGE ACCURACIES FOR SENTIMENT CLASSIFICATION

Target domain	B	D	E	K	Avg
In-domain	81.6	80.9	84.2	86.1	83.2
In-domain-FL	85.6	82.0	87.2	89.2	86.0

Transfer Task	D-B	E-B	K-B	B-D	E-D	K-D	B-E	D-E	K-E	B-K	D-K	E-K	Avg
SVM	73.5	65.4	66.7	74.6	68.2	71.6	71.4	70.8	79.7	72.9	73.8	81.1	72.4
GFK [29]	70.1	66.8	65.5	70.9	66.5	67.7	67.7	66.4	73.8	70.3	70.1	76.6	69.4
Landmark [30]	79.0	74.3	75.6	78.3	76.3	75.1	78.5	80.7	82.3	81.6	79.3	83.4	78.7
mSDA-1 [39]	77.7	70.0	70.4	77.4	68.1	75.1	71.5	75.1	82.6	74.5	74.9	85.2	74.8
mSDA [39]	80.9	71.4	**76.2**	78.8	72.8	76.7	79.3	80	84.8	75.1	77.7	86.3	78.3
CDFL-g	79.1	73.5	72.3	81.8	74.8	76.9	77.4	78.9	**87.5**	80.2	80.0	**88.2**	79.2
CDFL-l	77.5	70.6	71.1	78.4	71.2	74.1	75.8	76.0	82.3	78.6	78.4	84.1	76.5
CDFL	**80.9**	**75.0**	74.8	**82.6**	**76.5**	**78.6**	**80.2**	**80.9**	86.1	**82.8**	**82.8**	87.9	**80.8**

with marginalization. As shown in Section IV, our cross-domain feature learning algorithm with Gaussian kernel could not be solved with marginalization directly. To fairly compare with the *CDFL-g*, we test our algorithm with linear kernel but without marginalization denoted as *CDFL-1*. We also show the results of mSDA method without marginalization as *mSDA-1*. As a simple baseline, we test the classification method *SVM* without considering domain discrepancy. Specifically, the classifier model is trained on the source domain and tested directly on the target domain. Moreover, we also show the classification accuracies within the target domain as *In-domain* in all experiments. Half of the instances in the target domain are selected for training SVM classifiers using the raw features, and the other half of the instances are used for testing.

A. Sentiment Classification

Analyzing sentiment in text has emerged as a very interesting and challenging research subject in the past decade. One of the major challenges is the domain discrepancy among reviews of different kinds of products. In practice, a sentiment analysis model that is learned on book reviews does not perform well on kitchen appliance reviews if applied directly [48]. One reason of the poor performance is that the kinds of features that indicate positive or negative sentiment in book reviews(source domain) are not the same as the features in kitchen appliances reviews(target domain).

For this application, our CDFL algorithm can reduce the domain discrepancy for the reviews of different products. Because the reviews are represented by text, we only need to consider single modal features. As discussed in III-C, the single modal cross-domain feature learning is the special case of our CDFL by discarding $\mathcal{L}_x$ (or $\mathcal{L}_y$) and $\mathcal{L}_{mc}$ in Eq. (1). As illustrated in Section III, we can obtain multiple layers of feature representations for each instance in source domain and target domain using our CDFL algorithm. These features of multiple layers of auto-encoders are concatenated together as a single feature representation for each instance. Then we train a supervised classifier, such as SVM, in source domain and test it in target domain.

1) Dataset and Features: For the cross-domain sentiment classification, we use the same dataset as in [27]. This dataset contains more than 340,000 reviews from 25 different types of products from Amazon. As in [39], we only consider the binary classification problem whether a review is positive or negative. Besides, a smaller dataset is created to evaluate the existing domain adaption methods. Each domain consists of 6,000 instances. In this work, we evaluate our method on the same small dataset which contains four types of products: books (B), DVDs (D), electronics (E) and kitchen (K) appliances. With these four domain instances, there are 12 tasks in total when we take every pair as a task. We use the same features as in [39], where the raw bag-of-words features are extracted.

2) Results: In the experiment, for the GFK method, we test it using different dimensions of the learned subspace. In Table I, we show the best results for each task. In Table I, we also show the experimental results of the Landmark method reported in [30]. The authors of [30] only show results on 4 tasks of the sentiment classification dataset. We show the results of remaining 8 tasks using the code provided by the authors. On the task K-B, mSDA performs better than our CDFL. In this experiment, the only difference between our CDFL and mSDA [39] is the cross-domain constraint. In mSDA, the scheme to learn domain invariant features is to reconstruct the corrupted features in the source domain and the target domain simultaneously. In our CDFL, besides the scheme used in mSDA, we add the cross-domain constraint $\mathcal{L}_{cd}$. Compared with mSDA, the worse performance of our CDFL on task K-B is probably due to the instance noises, which disturb the cross-domain constraint in the source domain and the target domain. Though CDFL-g achieves the best results on tasks K-E and E-K, and mSDA shows the best result on task K-B, our CDFL algorithm performs best on average. The average results are shown in the last column of Table I. Besides, CDFL-g performs better than mSDA-1 and CDFL-1 which are solved without marginalization. Thus we believe that the better performance of the CDFL, compared with CDFL-g, results from the marginalization, which also contributes to the performance advantage of mSDA [39]. It's worth noting that, also appeared in [39], on target domains D, K and E, our CDFL method trained on the source domain performs even better than the In-domain classifiers. This is because the mSDA or our CDFL can not only reduce the domain discrepancy but also improve the representability of the feature. To demonstrate this point, in the third row "in-domain-FL" of Table I, we add the accuracy result of the classifier for each domain which is trained on half of the data using the learned feature by our CDFL (do not consider the domain discrepancy) and tested on the other half. We can see that the results on all four domains are all increased when compared with "in-domain" results. However, on the DVDs domain and the electronics domain, the transfer accuracies are still higher than the "in-domain-FL" results using the learned features. This may be because more instances in the source domain can be used to train the domain transfer classifier.

B. Spam Filtering

To construct a spam filtering system, we need to train a classifier based on emails from a group of users with corresponding spam or not spam labels. For a new email, we want to classify it as spam or not spam by using our trained classifier. The challenge is that the distributions among various users are different. Besides, the spam emails always change their appearances overtime. We assume that the source domain contains previously collected spam e-mail messages, and the target domain contains spam e-mail messages generated from different users or in different time. One of the main problems of spam filtering model is the discrepancy between source domain and target domain.

A domain invariant feature representation can be learned to solve the above mentioned problem in spam filtering. Here, the single-modal version of our CDFL in Section III-C is adopted. The learned multiple layers of features for each instance through our CDFL algorithm are concatenated together for training a classifier on the source domain. The learned classifier on the source domain is tested on the target domain. To verify the performance of our algorithm, we compare it with recent cross-domain feature representation methods with single modal features.

1) Dataset and Features: For the spam filtering application, we use the UCI Spam dataset.[2] This dataset contains 4,601 emails with 2,788non-spam instances and 1,813 spam instances, which are represented by 57-dimensional features. For fair comparison with state-of-the-art methods, we use the same scheme as in [31]. The original UCI Spam dataset is randomly split into three different sets of equal size. The first sample set is used to represent the source domain. The other two sets are used as unlabeled training samples from the target domain and test samples in the target domain. To simulate different distributions, the last two sample sets are created by adding Gaussian noise as in [31]. Specifically, Gaussian noise is generated for the n-th element of the original features according to $\mathcal{G}(\mu_n, \delta_n)$. The mean μ_n and the standard deviation δ_n are sampled from a uniform distribution among $[-0.15, 0.15]$ and $[0, 0.5]$, respectively. This process is repeated for 5 times for 5 different cross-domain tasks in the experiments.

2) Results: For spam filtering, we also compare our CDFL with the state-of-the-art unsupervised cross-domain learning methods. The results in Table II show that the proposed CDFL method performs the best against all the other methods. The first 6 results in Table II are reported in [31]. To fairly compare with the SLDAB [31], we denote the SVM in [31] as SVM-1, and our implemented version as SVM-2. For the GFK [29] and Landmark [30] methods, we give the results by running the source code provided by the authors. For the mSDA method [39] and the proposed CDFL algorithm, we use two layer auto-encoders. Furthermore, the regularization parameters used for linear SVM in SVM-2, mSDA [39] and our CDFL methods are all set to be the same value 10 for fair comparison. Other values of the regularization parameters can not affect the final conclusion for the comparison of our method and the baselines. Although the randomly generated 5 tasks are different, the average performance demonstrates the effectiveness of our proposed CDFL method. In addition, the accuracy on each task is shown in Table III.

TABLE II
ACCURACIES FOR SPAM FILTERING (SVM-2 IN THE BOTTOM PART ARE OUR IMPLEMENTED VERSION USING LIBSVM)

Method	Avg
SVM-1	62.0
Adaboost [49]	40.6
DASVM [50]	62.5
SVM-W [51]	62.1
SLDAB-H [31]	62.9
SLDAB-gn [31]	64.2
In-domain	79.8
SVM-2	64.3
GFK [29]	65.8
Landmark [30]	66.2
mSDA-1 [39]	65.8
mSDA [39]	68.0
CDFL-g	68.2
CDFL-l	66.5
CDFL	**70.1**

TABLE III
ACCURACIES FOR EACH TASK OF THE FIVE SPAM FILTERING TASKS

Task	1	2	3	4	5	Avg
In-domain	80.6	79.9	78.2	79.0	81.5	79.8
SVM-2	64.4	63.8	64.4	64.6	64.2	64.3
GFK [29]	65.4	76.2	75.0	56.0	56.7	65.8
Landmark [30]	67.4	66.7	63.9	67.1	65.9	66.2
mSDA-1 [39]	66.1	64.8	65.7	66.2	66.3	65.8
mSDA [39]	72.6	66.8	71.6	63.5	65.4	68.0
CDFL-g	69.5	67.2	67.2	**68.3**	**68.9**	68.2
CDFL-l	67.1	66.1	66.1	66.2	66.8	66.5
CDFL	**74.5**	**68.6**	**72.6**	67.6	67.2	**70.1**

C. Event Classification

The above two applications are adopted to evaluate the performance of our CDFL for single modal features. To test our CDFL for multi-modal features, we apply it to event classification. Here, we introduce a cross-domain experiment based on the event dataset crawled from Flickr and Google. Each instance in the event dataset is represented by the corresponding image and text descriptions. The cross-domain experiment consists of two sub-experiments. In the first sub-experiment, we classify the event samples on Google domain with the help of the samples on Flickr domain. In the second sub-experiment, Google is adopted as the source domain to classify the samples with only images on Flikcr domain. In the following, we will firstly introduce the event dataset in Section V-C1. Then details of the two cross-domain sub-experiments for event classification will be illustrated in Section V-C2 and V-C3 respectively.

Dataset and Features: There are already some event datasets, such as the TRECVID [52]. However, the TRECVID dataset mainly focuses on some simple and general events. In this paper, to analyze events captured by media data with the multi-modal and multi-domain properties, we mainly focus on 11$ complex

[2] http://archive.ics.uci.edu/ml/datasets/Spambase

TABLE IV
ILLUSTRATION OF THE EVENT NAME, DURATION TIME, AND NUMBER OF CRAWLED SAMPLES FOR EACH EVENT IN OUR EVENT DATASET

Event ID	Event Name	Start Time	End Time	# Google$_{image}$	# Google$_{text}$	# Flickr$_{image}$	# Flickr$_{text}$
1	Occupy Wall Street	09.2011	09.2012	435	108	181	181
2	2008 Chinese Milk Scandal	07.2008	11.2009	252	115	108	108
3	2012 United States Election	10.2009	01.2013	329	299	394	394
4	Protests against the Iraq War	09.2002	05.2012	465	204	311	311
5	Mass Killings in America	04.1999	12.2012	289	154	159	159
6	Greek Protests	05.2010	04.2012	312	100	140	140
7	2000s European Sovereign Debt Crisis	10.2009	05.2012	424	335	365	365
8	Mars Reconnaissance Orbiter	04.2005	08.2012	404	123	109	109
9	War in Afghanistan	10.2001	08.2012	493	402	441	441
10	North Korea Nuclear Program	01.1989	04.2012	416	402	630	630
11	2008 Great Recession	07.2008	11.2012	416	401	345	345

TABLE V
AVERAGE ACCURACIES OF EVENT CLASSIFICATION ON GOOGLE WITH THE HELP OF FLICKR

Feature	In-domain	SVM	GFK [29]	Landmark [30]	mSDA-1 [39]	mSDA [39]	CDFL-g	CDFL-l	CDFL
Text	88.6	82.9	74.0	83.5	82.7	83.0	84.2	83.1	**84.6**
Image	30.9	21.5	18.0	20.6	22.6	23.6	22.9	22.1	**24.9**
Combined	89.2	79.7	68.1	80.7	75.8	80.5	83.6	80.0	**86.8**

and public events. These 11 events are well-known all over the world and have abundant media data captured by users. In order to cover the whole evolutionary process of each event, we manually create the introduction page of each event or download it from the Wikipedia.[3] We then search and download related web pages from both Google and Flickr based on the keywords for each event. For each query, the corresponding texts (On Flickr, they are descriptions, tags and title. On Google, they are common web pages like news or blogs.) and corresponding images are downloaded. The details of our collected dataset are shown in Table IV. The collected 11 events cover a wide range of topics including politics, economics, techniques, military and society. Totally, there are 2463 documents and 4235 images from Google, and 3183 documents and 3183 images from Flickr. Some events are quite similar, such as "Occupy Wall Street" and "Greek protests" with similar topics, which brings great challenges to classification.

The textual and visual features are extracted as follows. *For text features*, two preprocessing tasks are conducted. The first task is segmenting the document to get the separate feature items in words. The second task is to remove the stop words. After the preprocessing, we extract the remaining words as the feature items of the event texts. This paper takes the commonly used vector space model to represent the text features [53]. The corresponding weights of the feature items are calculated by TF-IDF [54]. *For visual features*, we adopt the popular sparse coding based method by considering its performance and efficiency. It includes the extraction of local descriptors, codebook design, local descriptors coding, pooling, and a spatial layout sensitive concatenating. The local descriptors can be obtained by densely extracting SIFT [55] from images. The codebook (or dictionary) is built to represent the local descriptors using K-means. For local descriptors coding, the localized soft-assignment coding (LSC) [56] is adopted. Max-pooling [57] is adopted to obtain a compact description from the obtained codes. Finally, Spatial Pyramid Matching (SPM), proposed in [58], is exploited to concatenate all the code vectors. For the same query with multiple returned images, we adopt the max-pooling strategy [57] to obtain a single feature vector. The final vector of each image is viewed as a semantic description of the image.

1) Event Classification With the Help of Flickr: The feature representation is important for event recognition in practical applications [59]. As mentioned in Section I, a huge number of media data are generated in the Internet with multiple modalities. These media data can be easily collected from the social media sites like Flickr. And the uploaded images on Flickr are always accompanied with textual information, such as, tags and descriptions. Based on these media data on Flickr, we can learn an event classifier. However, the domain discrepancy will influence the performance if the classifier is applied on the target domain (Google).

In this sub-experiment, we learn domain invariant event feature representation for instances on both Flickr and Goolge domains in an unsupervised manner. The learned multiple layers of features for each event instance through our CDFL algorithm are concatenated together for training a classifier on the Flickr domain. Then, we test it for event instances on Google domain. Here, the image and text features of each event instance are adopted simultaneously.

To evaluate our algorithm, we compare it with two unsupervised cross-domain methods, GFK [29] and mSDA [39]. To show the best results of GFK method, we select the subspace dimension to be 55, 105 and 350 for text, image and combined features, respectively. The experimental results are shown in Table V. We also report the results without considering the domain discrepancy in the first column of Table V. These results are achieved by training an SVM classifier on source domain and directly testing it on the target domain. To utilize the combined features, all algorithms except our method simply concatenate the image and text features together. Based on the re-

[3] http://www.wikipedia.org/

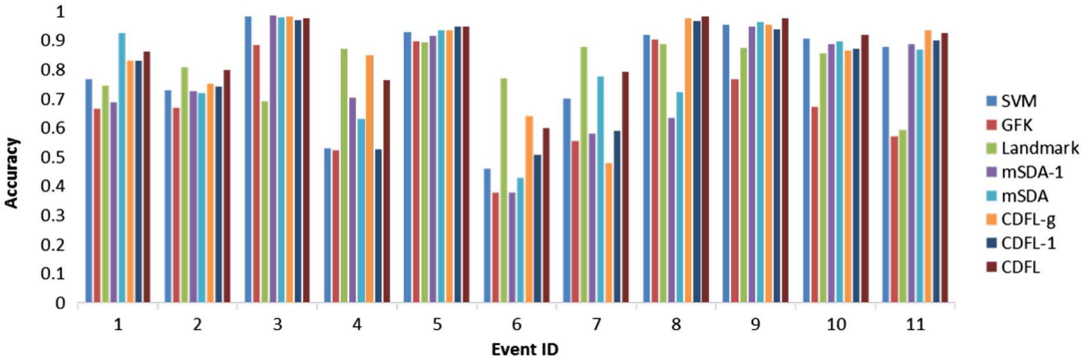

Fig. 4. Classification accuracy comparison for each event on Google.

sults in Table V, we can see that our CDFL algorithm performs the best compared with the three baselines. GFK method is effective for object recognition as illustrated in [29]. However, the accuracies of GFK method are much lower than the SVM baseline in our event classification experiment. This is probably due to the large discrepancy between visual features in different social media sites, which contain many noisy images. Moreover, both our CDFL and mSDA adopt marginalized denoising auto-encoder for feature learning. However, our CDFL performs better than mSDA due to the modal correlation constraint and the cross-domain constraint, especially when using both image and text features jointly. We also show the accuracy comparison of each event class by the combined features in Fig. 4. We can see that, even though the CDFL-g performs better on some event categories, our CDFL performs best on average. This is consistent with the results of the previous experiments shown in Section V-A and Section V-B.

Finally, we show an extra comparison with the multimodal deep learning method [40] which also deals with the multimodal problem. The authors do not provide source code. So we implement their algorithm by ourselves and adopt the similar scheme as mSDA for cross-domain feature learning. With the similar architecture of a 4-layer deep autoencoder as in [40], we only obtain 73% accuracy performance on the event dataset. The multi-modal deep learning method [40] does not consider the domain discrepancy, to make it suitable for the cross-domain event classification task mentioned in Section V-C-II, we adopt the similar scheme as [39] by training the deep autoencoder with the instances in the Flickr domain and the Google domain. After finishing the training of the deep autoencoder, we train an SVM classifier with the shared feature representations on the Flickr domain and test it on the Google domain. Since the authors of [40] do not share their code, we implement the bimodal deep autoencoder based on a deep learning toolbox written in Matlab.[4] More details about the multi-modal deep model are introduced in following. *Architecture of the deep autoencoder:* In the bimodal deep autoencoder, there are two separate hidden layers for image and text respectively which are followed with a shared feature layer. The node number of the hidden layer for image is 2 times the dimension of the image features while the node number for text is 1.5 times. The node number of the second layer (shared feature layer) is 1.5 times the total number of the nodes contained in two hidden layers for image and text. Note that this deep structure is slightly different from the deep autoencoder proposed in [40] for video and audio inputs. Here, we change the architecture and the parameters of the deep autoencoder through coarse tuning to improve the performance. *Training scheme*: The bimodal deep autoencoder is initialized using sparse RBMs. With the same scheme adopted in [40], we also add examples which have zero values for one of the input modalities (e.g., image) and original values for the other input modality (e.g, text). For these additional examples, the deep autoencoder is still required to reconstruct both modalities (image and text).

2) Event Classification on Flickr: In this sub-experiment, we consider the case where only images are available for event recognition on the Flickr domain. In this case, only visual features can be used to decide the class labels of event samples. This is a difficult but common problem in practical applications, because there are a huge number of photos on the social media sites without any metadata like tags or descriptions. Here, we attempt to use both image and text features on Google domain to classify events on Flickr domain. The samples on Google domain contain both text and image features while samples on Flikcr domain only contain image features.

For this sub-experiment, the objective functions $\mathcal{L}_g$ and $\mathcal{L}_l$ are redefined as $\widetilde{\mathcal{L}_g}$ and $\widetilde{\mathcal{L}_l}$, respectively. Here, $\widetilde{\mathcal{L}_g} = \lambda_x \mathcal{L}_x + \lambda_y \widetilde{\mathcal{L}_y} + \lambda_{mc} \widetilde{\mathcal{L}}_{mc} + \lambda_{cd} \widetilde{\mathcal{L}}^g_{cd}$, $\widetilde{\mathcal{L}_l} = \lambda_x \mathcal{L}_x + \lambda_y \widetilde{\mathcal{L}_y} + \lambda_{mc} \widetilde{\mathcal{L}}_{mc} + \lambda_{cd} \widetilde{\mathcal{L}}^l_{cd}$. Different from $\mathcal{L}_y$, $\mathcal{L}_{mc}$, $\mathcal{L}^g_{cd}$, and $\mathcal{L}^l_{cd}$, $\widetilde{\mathcal{L}}_y$ formulates the reconstruction loss of text features only on the source domain, $\widetilde{\mathcal{L}}_{mc}$ formulates the modal correlation only in the source domain, and $\widetilde{\mathcal{L}}^g_{cd}$ and $\widetilde{\mathcal{L}}^l_{cd}$ consider the discrepancy between the source domain and the target domain only for image features. The solutions to these two optimization problems are similar to the process as introduced in Section IV.

After feature learning via our CDFL algorithm, we adopt the learned image features on Google domain to train a classifier and test it for event instances with the learned image features on Flickr domain. The experimental results shown in Table VI verify the effectiveness of our algorithm. We can see that both GFK and mSDA methods do not outperform the SVM which does not consider the domain discrepancy. This is due to the large discrepancy between Flickr domain and Google domain. Different from these methods, our CDFL method shows much better performance.

[4] http://www.mathworks.com/matlabcentral/fileexchange/42853-deep-neural-network

TABLE VI
AVERAGE EVENT RECOGNITION ACCURACIES IN IMAGE COLLECTION ON FLICKR

Methods	In-domain	SVM	GFK [29]	Landmark [30]	mSDA-1 [39]	mSDA [39]	CDFL-g	CDFL-l	CDFL
Accuracy	25.9	17.6	16.9	19.4	16.5	17.4	19.3	17.3	**21.0**

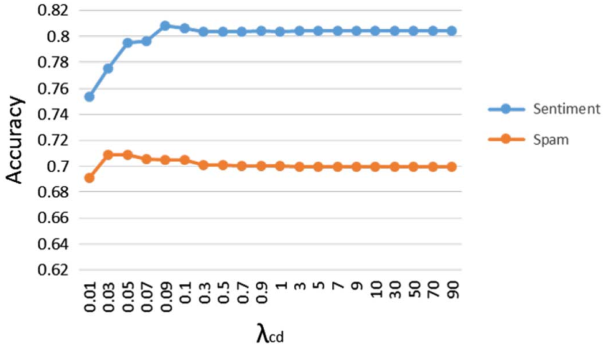

Fig. 5. Accuracy versus λ_{cd}. Accuracies of our CDFL method with different values of λ_{cd} on the sentiment and the spam experiments.

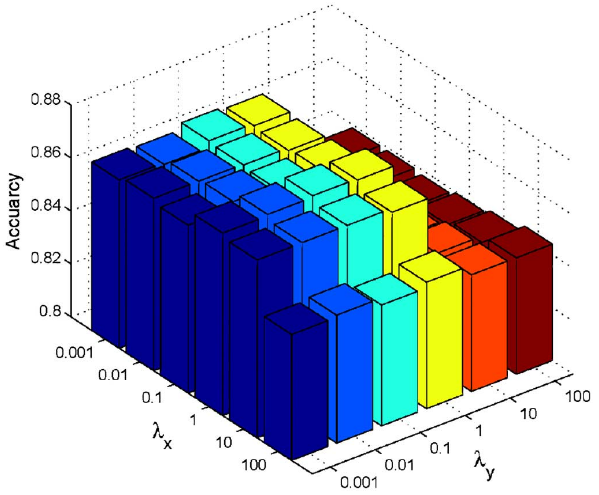

Fig. 6. Accuracy versus different λ_x and λ_y. λ_{mc} and λ_{cd} are both fixed to 1.

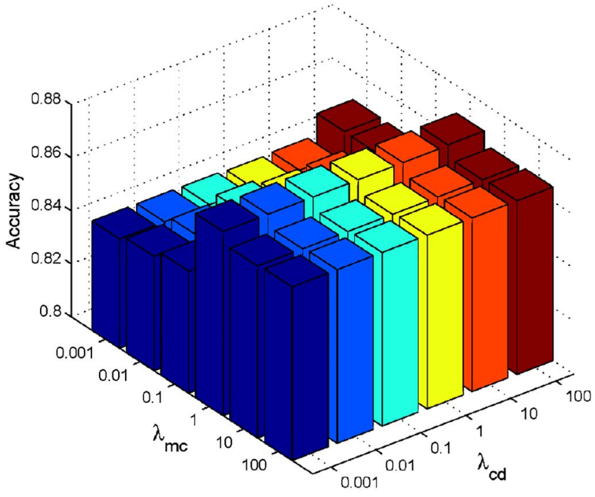

Fig. 7. Accuracy versus different λ_{mc} and λ_{cd}. λ_x and λ_y are both fixed to 1.

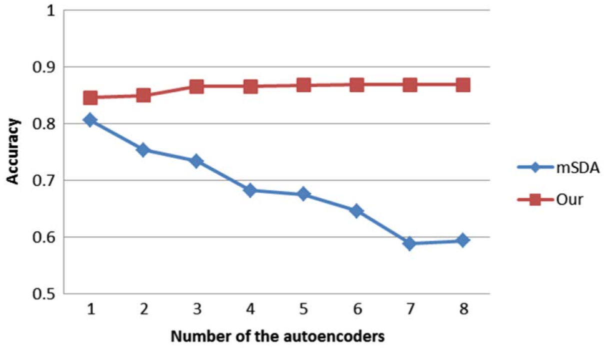

Fig. 8. Accuracy versus number of the auto-encoders for the experiment in Section V-C2. Accuracy of our method increases with the number of the auto-encoders.

D. Parameter Analysis

Several parameters play important roles in the proposed algorithm. In this section, we show how to determine their values and their effects on accuracy performance.

1) Effect of λ *:* The four parameters λ_x, λ_y, λ_{mc} and λ_{cd} in Eq. (1) are used to determine the weights of corrupted feature reconstruction, modal correlation constraint, and cross-domain constraint, respectively.

For the first two experiments as shown in Section V-A and Section V-B, only the λ_{cd} related to the cross-domain constraint needs to be set up while the λ of the reconstruction item is fixed. In Fig. 5, we fix $\lambda_x = 1$ and show the accuracies with different values of λ_{cd}. We can see that λ_{cd} has a small effect on the performance of our CDFL method. Overall, the results are quite stable. Based on these results, we can see that it is good to set λ_{cd} to 0.09 and 0.03 for the sentiment experiment and the spam experiment, respectively, due to their performances.

For the third experiment as shown in Section V-C, tuning all these four parameters is a time consuming task. Here, we only adopt an approximation method by tuning two parameters while the remaining two parameters are fixed. In Fig. 6, we show the performances of the sub-experiment in Section V-C2 using different parameters λ_x and λ_y while λ_{mc} and λ_{cd} are fixed to 1. In Fig. 7, we show the performances using different parameters λ_{mc} and λ_{cd} while λ_x and λ_y are fixed to 1. We can see that a better performance can be achieved by setting λ_{mc} to 1. Meanwhile, λ_{cd} can be set to any values from 1 to 100 because it has a little effect on the final performance.

Effect of p *:* In our implementation, the corruption probability p is chosen according to the best performance on the sentiment dataset reported in mSDA [39]. Following the setting in mSDA [39], the p is set to 0.7 in all experiments.

2) Effect of the Number of Layers: By sequentially concatenating outputs of different layers of denoisers, we get different results. For all methods, we choose the layer number to be 5 according to the best results reported in mSDA [39]. For the event classification experiment, a single layer autoencoder is adopted for the mSDA method because its performance will decrease with the increase of the number of layers as shown in Fig. 8. To further analyze the performance of our CDFL method,

TABLE VII
COMPUTATIONAL COST (SECONDS) FOR THREE EXPERIMENTS

Methods	SVM	GFK [29]	Landmark [30]	mSDA-1 [39]	mSDA [39]	CDFL-g	CDFL-l	CDFL
Sentiment (Section V-A)	82.0	139	208k	341	328	115k	436	398
Spam (Section V-B)	0.57	0.12	240	2.88	2.29	135	2.93	2.54
Event1 (Section V-C2)	4.06	17.1	79.6k	217	192	68.4k	229	212
Event2 (Section V-C3)	3.41	3.88	12.7k	185	168	5.05k	196	184

we show the accuracies of the sub-experiment in Section V-C2 when using a different number of layers in Fig. 8. These results are obtained by concatenating different layers of latent features learned by mSDA and our CDFL on the event dataset. We can see that the accuracies by mSDA feature learning are decreased with the increase of the number of layers. Different from the mSDA, the accuracies of our CDFL will be improved with more layers of auto-encoders. The reason is that the conventional mSDA does not consider the modal differences between image and text features. As a result, some noise will be introduced in the learned deep layer features of the mSDA method. Different from the mSDA, our CDFL algorithm considers the difference and maximizes the correlations among different modalities, and achieves better performance.

E. Computational Cost Analysis

In Table VII, we show the computational cost of all methods. For each method, we give the average computational cost of all domain transfer tasks on the preceding three experiments. For simplicity, the sentiment classification experiment in Section V-A is denoted as sentiment, the spam filtering experiment in Section V-B as spam. The two sub-experiments of the third experiment about cross-domain event classification in Section V-C are denoted as Event1 and Event2, respectively. For the Event1 experiment, we only show the time cost of all methods using combined features. Though two additional constraints are used, our CDFL is only slightly slower than the mSDA. In CDFL, we only use the linear kernel to approximate the MMD. Thus, we can obtain the closed form solutions through marginalization which could be implemented by simple matrix operation as in the mSDA. The CDFL-g can not be solved in closed form due to the Gaussian kernel. The gradient descent method dominates most of the solving time. For both mSDA and our CDFL, the total time cost for training features is linearly proportional to the number of auto-encoders. All the experiments are carried out on a PC with 16G RAM and 3.1 GHZ CPU.

VI. CONCLUSION

In this paper we proposed CDFL, an algorithm for obtaining representations of multimedia data that have multiple modalities and come from different domains. In the formulation of our CDFL, we introduce the modal correlation and cross-domain constraints in the conventional marginalized denoising auto-encoders. We evaluate our CDFL on three cross-domain applications: sentiment classification, spam filtering and event classification. The experimental results demonstrate the effectiveness and generality of our CDFL.

APPENDIX A
CONVEXITY OF $\mathcal{L}_{mc}$

Firstly, we explain the convexity of the third item $\mathcal{L}_{mc}$ in Eq. (1). We rewrite $\mathcal{L}_{mc}$ as

$$\begin{aligned}\mathcal{L}_{mc} &= Tr\left(W_x C_{xx} W_x^\top\right) + Tr\left(W_y C_{yy} W_y^\top\right) \\ &\quad - 2Tr\left(W_x C_{xy} W_y^\top\right) \\ &= Tr\left(W_x \tilde{X}\tilde{X}^\top W_x^\top\right) + Tr\left(W_y \tilde{Y}\tilde{Y}^\top W_y^\top\right) \\ &\quad - 2Tr\left(W_x \tilde{X}\tilde{Y}^\top W_y^\top\right) \\ &= Tr\left(\left(W_x\tilde{X} - W_y\tilde{Y}\right)\left(W_x\tilde{X} - W_y\tilde{Y}\right)^\top\right) \\ &= \|W_x\tilde{X} - W_y\tilde{Y}\|_2^2. \end{aligned} \tag{30}$$

Since $\|W_x\tilde{X} - W_y\tilde{Y}\|_2^2$ is convex, $\mathcal{L}_{mc}$ is also a convex quadratic function.

APPENDIX B
PARTIAL GRADIENTS

A. Partial gradients of $\mathcal{L}_x$ and $\mathcal{L}_y$

$$\begin{aligned}\frac{\partial \mathcal{L}_x}{\partial W_x} &= \frac{\partial Tr\left(\tilde{X}^T W_x^T W_x \tilde{X}\right)}{\partial W_x} - \frac{\partial Tr\left(X^\top W_x \tilde{X}\right)}{\partial W_x} \\ &\quad - \frac{\partial Tr\left(\tilde{X}^\top W_x^\top X\right)}{\partial W_x} \\ &= \left[W_x\tilde{X}\tilde{X}^\top + W_x\tilde{X}\tilde{X}^\top - X(\tilde{X})^\top - X\tilde{X}^\top\right] \\ &= 2\left[W_x\tilde{X}\tilde{X}^\top - X(\tilde{X})^\top\right] = 2\left[W_x C_{xx} - \bar{C}_{xx}\right] \end{aligned} \tag{31}$$

Similarly, we get the partial gradient for W_y

$$\frac{\partial \mathcal{L}_y}{\partial W_y} = 2\left[W_y\tilde{Y}\tilde{Y}^\top - Y(\tilde{Y})^\top\right] = 2\left[W_y C_{yy} - \bar{C}_{yy}\right] \tag{32}$$

B. Partial Gradients of $\mathcal{L}_{mc}$

$$\begin{aligned}\frac{\partial \mathcal{L}_{mc}}{\partial W_x} &= W_x C_{xx}^\top + W_x C_{xx} - 2W_y C_{xy}^\top \\ &= 2(W_x C_{xx} - W_y C_{xy}^\top) \end{aligned} \tag{33}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{mc}}{\partial W_y} &= W_y C_{yy}^\top + W_y C_{yy} - 2W_x C_{xy} \\ &= 2(W_y C_{yy} - W_x C_{xy}) \end{aligned} \tag{34}$$

C. Partial Gradients of $\mathcal{L}^l_{cd}$

$$\frac{\partial \mathcal{L}^l_{cd}}{\partial W_x} = 2W_x G_x G_x^\top, \frac{\partial \mathcal{L}^l_{cd}}{\partial W_y} = 2W_y G_y G_y^\top \tag{35}$$

D. Partial Gradients of $\mathcal{L}^l$

Compute the partial derivative of $\mathcal{L}^l$ with respect to W_x

$$\begin{aligned}\frac{\partial \mathcal{L}^l}{\partial W_x} &= 2\lambda_x \left[W_x C_{xx} - \bar{C}_{xx}\right] + 2\lambda_c W_x C_{xx} \\ &\quad - \lambda_c W_y C_{xy}^\top + 2\lambda_m W_x G_x G_x^\top \\ &= W_x Q_x - 2\lambda_c W_y C_{xy}^\top - 2\lambda_x \bar{C}_{xx}\end{aligned} \tag{36}$$

Similarly, we get the partial derivative of $\mathcal{L}^l$ with respect to W_y

$$\begin{aligned}\frac{\partial \mathcal{L}^l}{\partial W_y} &= 2\lambda_y \left[W_y C_{yy} - \bar{C}_{yy}\right] + 2\lambda_c W_y C_{yy} \\ &\quad - \lambda_c W_x C_{xy} + 2\lambda_m W_y G_y G_y^\top \\ &= W_y Q_y - 2\lambda_c W_x C_{xy} - 2\lambda_y \bar{C}_{yy}\end{aligned} \tag{37}$$

References

[1] M. Naaman, "Social multimedia: Highlighting opportunities for search and mining of multimedia data in social media applications," *Multimedia Tools Appl.*, vol. 56, no. 1, pp. 9–34, 2012.

[2] J. Sang and C. Xu, "Browse by chunks: Topic mining and organizing on web-scale social media," *TOMCCAP*, vol. 7, p. 30, 2011.

[3] S. D. Roy, T. Mei, W. Zeng, and S. Li, "Socialtransfer: Cross-domain transfer learning from social streams for media applications," in *Proc. ACM Multimedia*, 2012, pp. 649–658.

[4] J. Sang and C. Xu, "Right buddy makes the difference: An early exploration of social relation analysis in multimedia applications," in *Proc. 20th ACM Multimedia Conf.*, Nara, Japan, Oct.–Nov. 2012, pp. 19–28.

[5] X. Jin, A. C. Gallagher, L. Cao, J. Luo, and J. Han, "The wisdom of social multimedia: Using Flickr for prediction and forecast," in *Proc. ACM Multimedia*, 2010, pp. 1235–1244.

[6] G. Ma and J. Luo, "Is a picture worth 1000 votes? Analyzing the sentiment of election related social photos," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2013, pp. 1–6.

[7] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li, "User-level sentiment analysis incorporating social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, New York, NY, USA, 2011, pp. 1397–1405.

[8] B.-K. Bao, W. Min, K. Lu, and C. Xu, "Social event detection with robust high-order co-clustering," in *Proc. 3rd ACM Conf. Int. Conf. Multimedia Retrieval*, New York, NY, USA, 2013, pp. 135–142.

[9] M. Zaharieva, M. Zeppelzauer, and C. Breiteneder, "Automated social event detection in large photo collections," in *Proc. 3rd ACM Conf. Int. Conf. Multimedia Retrieval*, New York, NY, USA, 2013, pp. 167–174.

[10] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris, "Social event detection using multimodal clustering and integrating supervisory signals," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, New York, NY, USA, 2012, pp. 23:1–23:8.

[11] M. Brenner and E. Izquierdo, "Social event detection and retrieval in collaborative photo collections," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, New York, NY, USA, 2012, pp. 21:1–21:8.

[12] Y. Wang, H. Sundaram, and L. Xie, "Social event detection with interaction graph modeling," in *Proc. 20th ACM Int. Conf. Multimedia*, New York, NY, USA, 2012, pp. 865–868.

[13] S. Orlando, F. Pizzolon, and G. Tolomei, "Seed: A framework for extracting social events from press news," in *Proc. 22nd Int. Conf. World Wide Web Companion*, Geneva, Switzerland, 2013, pp. 1285–1294.

[14] T. Reuter and P. Cimiano, "Event-based classification of social media streams," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, New York, NY, USA, 2012, pp. 22:1–22:8.

[15] X. Liu and B. Huet, "Heterogeneous features and model selection for event-based media classification," in *Proc. 3rd ACM Conf. Int. Conf. Multimedia Retrieval*, New York, NY, USA, 2013, pp. 151–158.

[16] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, 2010, vol. 4, pp. 213–226.

[17] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," in *Proc. ECCV*, 2012, vol. 2, pp. 702–715.

[18] B. Gong, K. Grauman, and F. Sha, "Reshaping visual datasets for domain adaptation," in *Proc. NIPS*, 2013, pp. 1286–1294.

[19] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[20] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. NIPS*, 2006, pp. 153–160.

[21] M. Ranzato, C. S. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Proc. NIPS*, 2006, pp. 1137–1144.

[22] H. Theil and C.-F. Chung, "Relations between two sets of variates: The bits of information provided by each variate in each set," *Statist. Probability Lett.*, vol. 6, no. 3, pp. 137–139, Feb. 1988.

[23] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput,*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

[24] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proc. ACM Multimedia*, 2010, pp. 251–260.

[25] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos, "Bridging the gap: Query by semantic example," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 923–938, Aug. 2007.

[26] M. Guillaumin, J. J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 902–909.

[27] J. Blitzer, R. T. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. EMNLP*, 2006, pp. 120–128.

[28] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *Proc. IJCAI*, 2009, pp. 1187–1192.

[29] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2066–2073.

[30] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. ICML*, 2013, pp. 222–230.

[31] A. Habrard, J.-P. Peyrache, and M. Sebban, "Boosting for unsupervised domain adaptation," in *Proc. ECML/PKDD*, 2013, pp. 433–448.

[32] Z. Ma, Y. Yang, Y. Cai, N. Sebe, and A. G. Hauptmann, "Knowledge adaptation for ad hoc multimedia event detection with few exemplars," in *Proc. ACM Multimedia*, 2012, pp. 469–478.

[33] W. Lu, J. Li, T. Li, W. Guo, H. Zhang, and J. Guo, "Web multimedia object classification using cross-domain correlation knowledge," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1920–1929, Dec. 2013.

[34] G. Qi, C. C. Aggarwal, and T. S. Huang, "Towards semantic knowledge propagation from text corpus to web images," in *Proc. WWW*, 2011, pp. 297–306.

[35] Y. Bengio, "Deep learning of representations: Looking forward," in *Proc. SLSP*, 2013, pp. 1–37.

[36] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, vol. 1, ch. 6, pp. 194–281.

[37] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008, pp. 1096–1103.

[38] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. ICML*, 2011, pp. 513–520.

[39] M. Chen, Z. E. Xu, K. Q. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. ICML*, 2012, pp. 767–774.

[40] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. ICML*, 2011, pp. 689–696.

[41] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," in *Proc. ISMB*, 2006, pp. 49–57.

[42] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, "Covariate shift by kernel mean matching," *Dataset Shift Mach. Learn.*, vol. 3, no. 4, p. 5, 2009.

[43] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 769–776.

[44] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.

[45] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 67–93, 2001.

[46] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, 2014.

[47] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[48] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. ACL*, 2007, pp. 440–447.

[49] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Mach. Learn.: Proc. 13th Int. Conf.*, 1996, pp. 148–156.

[50] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[51] J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proc. NIPS*, 2007, pp. 601–608.

[52] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot, "TRECVID 2013—An overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proc. TRECVID*, May 2013, pp. 1–45.

[53] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.

[54] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[55] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[56] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2486–2493.

[57] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 1794–1801.

[58] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, vol. 2, pp. 2169–2178.

[59] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

Xiaoshan Yang received the M.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2012, and is currently pursuing the Ph.D. degree from the Multimedia Computing Group, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

He was an intern with the China-Singapore Institute of Digital Media, Singapore, from September 2013 to April 2014. His research interests include multimedia and computer vision.

Tianzhu Zhang (S'09–M'11) received the bachelor's degree in communications and information technology from the Beijing Institute of Technology, Beijing, China, in 2006, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2011.

He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and multimedia, especially action recognition, object classification, and object tracking.

Changsheng Xu (M'97–SM'99–F'14) is a Professor with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and Executive Director of China-Singapore Institute of Digital Media, Singapore. He holds 30 granted and pending patents and has published over 200 refereed research papers. He is an Associate Editor for *ACM Transactions on Multimedia Computing, Communications, and Applications* and *ACM/Springer Multimedia Systems Journal*. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition, and computer vision.

Dr. Xu is an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA. He has served as Associate Editor, Guest Editor, General Chair, Program Chair, Area/Track Chair, Special Session Organizer, Session Chair, and TPC Member for over 20 IEEE and ACM prestigious multimedia journals, conferences, and workshops. He served as Program Chair of ACM Multimedia 2009. He received the Best Associate Editor Award of the*ACM Transactions on Multimedia Computing, Communications, and Applications* in 2012 and the Best Editorial Member Award of the *ACM/Springer Multimedia Systems Journal* in 2008. He is an IAPR Fellow and ACM Distinguished Scientist.