DOMAIN ADAPTATION BY ITERATIVE IMPROVEMENT OF SOFT-LABELING AND MAXIMIZATION OF NON-PARAMETRIC MUTUAL INFORMATION

M.N.A. Khan, Douglas R. Heisterkamp

Department of Computer Science
Oklahoma State University, Stillwater, OK
{mohk, doug}@cs.okstate.edu

ABSTRACT

Domain adaptation (DA) algorithms address the problem of distribution shift between training and testing data. Recent approaches transform data into a shared subspace by minimizing the shift between their marginal distributions. We propose a method to learn a common subspace that will leverage the class conditional distributions of training samples along with reducing the marginal distribution shift. To learn the subspace, we employ a supervised technique based on non-parametric mutual information by inducing soft label assignment for the unlabeled test data. The approach presents an iterative linear transformation for subspace learning by repeatedly updating test data predictions via soft-labeling and consequently improving the subspace with maximization of mutual information. A set of comprehensive experiments on benchmark datasets is conducted to prove the efficacy of our novel framework over state-of-the-art approaches.

***Index Terms*—** Mutual information, soft-labeling, subspace, transfer learning.

1. INTRODUCTION

To build a model for object class detection or regression problem, it is generally assumed that training and testing data are sampled from the same distribution. This assumption is often challenged in real life scenario, i.e. the dataset on which a model is trained (referred to as *source* domain) may vary significantly from the test data distribution (*target* domain) (Figure 1). This may degrade test accuracy or performance of the trained model and entails the necessity of adapting that model such that it can overcome the distribution difference among training and testing datasets, widely known as *dataset bias*, *domain shift* or *domain adaptation* [1, 2, 3]. In this paper, we will refer the terms 'domain'and 'data'interchangeably. The ultimate goal is to compensate the distribution divergence between source and target domains such that a classifier trained using source data can also perform well on diversely distributed but related target data. Formally, source and target data are represented with same feature encodings, but their marginal probability distributions will be different. Here, both domains have the same set of class labels. One practical example is, learning an image classification model with images generated from high-resolution camera whereas deploying that application into a device with low-resolution camera.

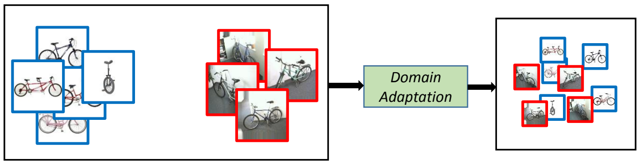

Fig. 1: Example of distribution differences for the same 'bicycle'class. DA methods will try to resolve this domain shift so that a classification model can perform effectively across domains.

To deal with the domain shift problem, two different settings are usually considered: i) unsupervised domain adaptation [4, 5, 6, 7], where no labeled data available in target domain and ii) semi-supervised domain adaptation [8, 9], where only a few labeled data are available in target domain along with abundant labeled data of source domain. In this paper, we will focus on the more challenging unsupervised case. One popular way to deal with the unsupervised case is to find a common feature subspace that expresses shared structures between source and target data. For example, Fernando *et al.*[5] proposed a linear projection function to align source and target distributions. Some other approaches focus in instance re-weighting of source data to match with target data in order to minimize their distribution differences [6, 10, 11, 2]. Most of these works deal with correcting marginal distribution shift [5, 4, 6], ignoring the class conditional distribution of source data. Therefore, the learned subspace may not be optimal in terms of class separation. This motivates us to utilize class discriminative information of source data to learn a common feature subspace with goals of resolving the distribution divergence and creating a discriminative subspace for unlabeled target data.

A supervised technique based on maximization of non-parametric mutual information (MI) between data and corresponding class labels has been proved effective in learning a discriminative subspace [12, 13]. Following prior work in domain adaptation setting with labeled source and unlabeled target data [7], an iterative method is proposed. At each iteration, a subspace is learned with MI maximization and then target data class predictions are created with *soft-labeling* (probability that a point belongs to a class) by utilizing neighboring source data in the learned subspace. The soft assignment of class labels for target data is integrated into the objective function [12] of MI maximization and influences the next iteration subspace learning. These two steps continue till converging to a final subspace.

In summary, the contributions of this work are i) utilizing class label distributions of source data along with all domain data distributions to develop a domain adaptation framework, ii) extending supervised method of MI to support unlabeled target data by inducing soft-labeling and iii) proposing an iterative approach of common subspace learning based on maximization of non-parametric MI induced with soft-labeling.

2. SUBSPACE LEARNING BY MAXIMIZING SOFT-LABELING INDUCED QUADRATIC MUTUAL INFORMATION (QMI-S)

According to information theoretic literature, *Mutual Information* (MI) is defined as a measure of dependence between random variables. Assume that X is a random variable representing d-dimensional data $\boldsymbol{x} \in \mathbb{R}^d$ and C is a discrete random variable representing class labels $c \in \{1, 2, \ldots, N_c\}$, where N_c is the total number of classes. Let $p(\boldsymbol{x})$ be the density function of $\boldsymbol{x}$ and $P(c)$ be the class prior probability. Using Renyi's entropy, quadratic mutual information (QMI) is a non-parametric estimation of MI and is defined as [13],

$$\begin{aligned}\mathcal{I}(X, C) &= \sum_c \int_{\boldsymbol{x}} (p(\boldsymbol{x}, c) - P(c)p(\boldsymbol{x}))^2 d\boldsymbol{x} \\ &= \sum_c \int_{\boldsymbol{x}} p(\boldsymbol{x}, c)^2 d\boldsymbol{x} + \sum_c \int_{\boldsymbol{x}} P(c)^2 p(\boldsymbol{x})^2 d\boldsymbol{x} \\ &\qquad - 2\sum_c \int_{\boldsymbol{x}} p(\boldsymbol{x}, c)P(c)p(\boldsymbol{x}) d\boldsymbol{x} \\ &= \mathcal{V}_{in} + \mathcal{V}_{all} - 2\mathcal{V}_{btw} \end{aligned} \tag{1}$$

Bouzas *et al.* proposed a subspace learning algorithm using trace-norm formulation of QMI [12]. The data distribution $p(\boldsymbol{x})$ is estimated by a Parzen window method using a Gaussian kernel. A multivariate Gaussian $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is,

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} e^{\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)}$$

A Parzen window density estimation of $p(\boldsymbol{x})$ with IID samples $\boldsymbol{x}_i$ is

$$p(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{1}{n} \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{x}_i, \sigma^2 \boldsymbol{I}\right)$$

where n is the cardinality of dataset.

We extend [12] by introducing a soft class labeling into the QMI formulation. The class prior probability $P(c)$ can be expressed as

$$P(c) = \sum_{i=1}^{n} P(c \mid \boldsymbol{x}_i) P(\boldsymbol{x}_i) = \sum_{i=1}^{n} P(c \mid \boldsymbol{x}_i) \frac{1}{n} = S_c$$

where $P(c|\boldsymbol{x}_i)$ is the probability that $\boldsymbol{x}_i$ belongs to class c (soft-labeling). For notational convenience, we will further refer $P(c)$ as S_c. The joint distribution $p(\boldsymbol{x}, c)$ can be expressed as

$$p(\boldsymbol{x}, c) = P(c \,|\boldsymbol{x}) p(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} P(c \,|\boldsymbol{x}_i) \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{x}_i, \sigma^2 \boldsymbol{I}\right)$$

Following [12]'s approach, a trace-norm derivation using the above expressions for $p(\boldsymbol{x})$, $P(c)$ and $p(\boldsymbol{x}, c)$ is presented in Table 1 and allows Eq.(1) to be rewritten as

$$\mathcal{I}(X, C) = \operatorname{tr}\left\{\boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi}\right\} \tag{2}$$

where

$$\begin{aligned}\boldsymbol{M} = \frac{1}{n^2}\left(\sum_c \boldsymbol{z}_c \boldsymbol{z}_c^T\right) + \left(\sum_c \frac{S_c^2}{n^2}\right) \boldsymbol{1}\boldsymbol{1}^T \\ - 2 \cdot \boldsymbol{1} \left(\sum_c \frac{S_c}{n^2}\right) \boldsymbol{z}_c^T \end{aligned} \tag{3}$$

where $\boldsymbol{z}_c = [P(c|\boldsymbol{x}_1), P(c|\boldsymbol{x}_2), \ldots, P(c|\boldsymbol{x}_n)]^T \in \mathbb{R}^{n\times 1}$ is the soft-labeling introduced into QMI and $\boldsymbol{\Phi} \in \mathbb{R}^{n\times m}$ represents the mapped data points from the original feature space to a kernel Hilbert space using a mapping function $\psi : \mathcal{X} \rightarrow \mathcal{H}$, i.e., $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$. This formulation is referred to as **QMI-S**.

A k-dimensional subspace is learned by finding a linear transformation $\boldsymbol{W} = \boldsymbol{\Phi}^T \boldsymbol{A}$, $\boldsymbol{A} \in \mathbb{R}^{n\times k}$, that maximizes **QMI-S**. That is, maximize $\mathcal{I}(X, C) = \operatorname{tr}\left\{\boldsymbol{W}^T \boldsymbol{\Phi}^T \boldsymbol{M} \boldsymbol{\Phi} \boldsymbol{W}\right\}$ which with unit covariance constraints becomes (see [12])

$$\boldsymbol{A}^* = \underset{\boldsymbol{A}^T \boldsymbol{K}\boldsymbol{A} = \boldsymbol{I}}{\operatorname{argmax}} \frac{\operatorname{tr}\left\{\boldsymbol{A}^T \boldsymbol{K}\boldsymbol{M}'\boldsymbol{K}\boldsymbol{A}\right\}}{\operatorname{tr}\left\{\boldsymbol{A}^T \boldsymbol{K}\boldsymbol{K}\boldsymbol{A}\right\}} \tag{4}$$

$$\text{with } \boldsymbol{M}' = (\boldsymbol{M} + \boldsymbol{M}^T)/2 \tag{5}$$

where $\boldsymbol{M}'$ is a symmetric form of $\boldsymbol{M}$. Centralized Gaussian kernel $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is defined as $\boldsymbol{K} = \boldsymbol{K}_g - \boldsymbol{E}_n \boldsymbol{K}_g - \boldsymbol{K}_g \boldsymbol{E}_n + \boldsymbol{E}_n \boldsymbol{K}_g \boldsymbol{E}_n$, where $\boldsymbol{K}_g(i, j) = \mathcal{N}\left(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{0}, 2\sigma^2 \boldsymbol{I}\right)$ and $\boldsymbol{E}_n \in \mathbb{R}^{n\times n}$ consists of elements each equal to $\frac{1}{n}$. The trace ratio problem of Eq.(4) can be approximated as a ratio trace optimization [14] and $\boldsymbol{A}^*$ is found using the generalized eigen value decomposition method [15]. After finding $\boldsymbol{A}^*$, the data is projected by $\boldsymbol{X}_p = \Phi \boldsymbol{W} = \boldsymbol{K}\boldsymbol{A}^{\boldsymbol{*}}$.

Table 1: Derivation of $\mathcal{V}_{in}$, $\mathcal{V}_{all}$ and $\mathcal{V}_{btw}$. Here $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$, $\boldsymbol{z}_c = [P(c|\boldsymbol{x}_1), P(c|\boldsymbol{x}_2), \ldots, P(c|\boldsymbol{x}_n)]^T \in \mathbb{R}^{n\times 1}$ and $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^{n\times 1}$

$\mathcal{V}_{in} = \sum_c \int_{\boldsymbol{x}} p(\boldsymbol{x}, c)^2 d\boldsymbol{x}$	$\mathcal{V}_{all} = \sum_c \int_{\boldsymbol{x}} P(c)^2 p(\boldsymbol{x})^2 d\boldsymbol{x}$	$\mathcal{V}_{btw} = \sum_c \int_{\boldsymbol{x}} p(\boldsymbol{x}, c) P(c) p(\boldsymbol{x}) d\boldsymbol{x}$
$= \frac{1}{n^2} \sum_c \sum_{i=1}^n \sum_{j=1}^n P(c\|\boldsymbol{x}_i) P(c\|\boldsymbol{x}_j) \mathcal{N}\left(\boldsymbol{x}_i - \boldsymbol{x}_j; 0, 2\sigma^2 \boldsymbol{I}\right)$	$= \sum_c S_c^2 \sum_{i=1}^n \sum_{j=1}^n \frac{1}{n^2} \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{x}_i, \sigma^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{x}_j, \sigma^2 \boldsymbol{I}\right)$	$= \sum_c S_c \sum_i^n \sum_j^n \left(\frac{1}{n^2}\right) P(c\|\boldsymbol{x}_i) \mathcal{N}\left(\boldsymbol{x}_i - \boldsymbol{x}_j; \mathbf{0}, 2\sigma^2 \boldsymbol{I}\right)$
$= \frac{1}{n^2} \sum_c {\boldsymbol{z}_c}^T \boldsymbol{K} \boldsymbol{z}_c$	$= \frac{1}{n^2} \left(\sum_c S_c^2\right) \mathbf{1}^T \boldsymbol{K} \mathbf{1}$	$= \frac{1}{n^2} \sum_c S_c {\boldsymbol{z}_c}^T \boldsymbol{K} \mathbf{1}$
$= \frac{1}{n^2} \sum_c \mathrm{tr}\{\boldsymbol{K} \boldsymbol{z}_c {\boldsymbol{z}_c}^T\}$	$= \frac{1}{n^2} \left(\sum_c S_c^2\right) \mathrm{tr}\{\boldsymbol{K} \mathbf{1} \mathbf{1}^T\}$	$= \frac{1}{n^2} \sum_c \mathrm{tr}\{S_c (\boldsymbol{K} \cdot \mathbf{1} \cdot \boldsymbol{z}_c^T)\}$
$= \frac{1}{n^2} \mathrm{tr}\left\{\boldsymbol{\Phi}^T \left(\sum_c \boldsymbol{z}_c \boldsymbol{z}_c^T\right) \boldsymbol{\Phi}\right\}$	$= \frac{1}{n^2} \left(\sum_c S_c^2\right) \mathrm{tr}\{\boldsymbol{\Phi}^T (\mathbf{1}\mathbf{1}^T) \boldsymbol{\Phi}\}$	$= \frac{1}{n^2} \mathrm{tr}\left\{\boldsymbol{\Phi}^T \left(\mathbf{1} \sum_c S_c \boldsymbol{z}_c^T\right) \boldsymbol{\Phi}\right\}$

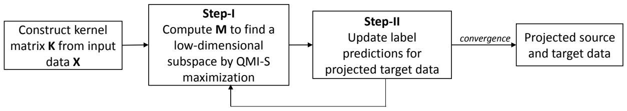
Construct kernel matrix **K** from input data **X**
Step-I
Compute **M** to find a low-dimensional subspace by QMI-S maximization
Step-II
Update label predictions for projected target data
convergence
Projected source and target data

Fig. 2: Proposed method for iterative subspace learning based on QMI-S maximization.

3. ITERATIVE IMPROVEMENT OF SOFT-LABELING AND QMI-S SUBSPACE

The data available are $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ consisting of source domain data $\boldsymbol{X}_s \in \mathbb{R}^{n_s\times d}$, target domain data $\boldsymbol{X}_t \in \mathbb{R}^{n_t\times d}$ and ground truth labels of source data $[y_1, y_2, \ldots, y_{n_s}]^T$, where n_s and n_t represent source and target data size respectively and $n = n_s + n_t$. $P(c|\boldsymbol{x}_i)$ is defined as follows. For $\boldsymbol{x}_i^s \in \boldsymbol{X}_s$, labels are known so hard labeling can be used (i.e, $P(c|\boldsymbol{x}_i^s) = 1$ if $c = y_i$ else 0). On the other hand, target data are unlabeled, so a full distribution is used. For target data, $P(c|\boldsymbol{x}_i^t)$ is initialized with uniform label distribution i.e., for $\boldsymbol{x}_i^t \in \boldsymbol{X}_t$, $P(c|\boldsymbol{x}_i^t) = \frac{1}{N_c}$ for each $c \in \{1, 2, \ldots, N_c\}$. If source labels are from a classifier instead of ground truth then the classifier's $P(c|\boldsymbol{x}_i^s)$ can be used instead of hard labels.

The proposed iterative **QMI-S** (Figure 2) consists of,

Step-I: $\boldsymbol{M'}$ is computed using Eq. (3) and (5). To find $\boldsymbol{A}^*$ for learning **QMI-S** subspace, Eq.(4) can be rewritten as $\boldsymbol{KM'KU} = \boldsymbol{KKU\Lambda}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigen values and $\boldsymbol{U}$ is a matrix of corresponding eigen vectors. For computational efficiency, we substitute $\boldsymbol{KU}$ with a new variable $\boldsymbol{V}$ *i.e.* $\boldsymbol{KU} = \boldsymbol{V}$, multiply both sides by $\boldsymbol{K}^{-1}$ and obtain $\boldsymbol{M'V} = \boldsymbol{V\Lambda}$. This is a standard eigen problem where $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ represent matrix of eigen vectors and eigen values respectively. As $\boldsymbol{M'}$ has rank $N_c - 1$, the $N_c - 1$ vectors with largest eigen values are selected from $\boldsymbol{V}$. Hence, $\boldsymbol{A}^*$ will be $\boldsymbol{A}^* = \boldsymbol{K}^{-\frac{1}{2}}\boldsymbol{V}$ and the projected data $\boldsymbol{X}_p \in \mathbb{R}^{n\times k}$ will be computed as $\boldsymbol{X}_p = \boldsymbol{KA}^* = \boldsymbol{K}^{\frac{1}{2}}\boldsymbol{V}$.

Step-II: Target data predictions $P(c|\boldsymbol{x}_i^t)$ are updated by applying a classifier trained with projected source data. This will eventually update $\boldsymbol{M'}$ of Step-I for the next iteration. $P(c|\boldsymbol{x}_i^s)$ will be remained same through out the iterations. The overall process is summarized in Algorithm 1.

Convergence criterion The proposed algorithm will reach convergence when subspace change in two successive iterations will be negligible. The subspace is defined by the basis vectors $\boldsymbol{V}$. The difference beween two k-dimensional subspaces can be approximated as a subspace distance on a Grassmannian [16]. One such distance metric measures the principal angle θ between $\boldsymbol{V}_i$ and $\boldsymbol{V}_{i+1}$ of iterations i and $i+1$ respectively [17, 16]. A convergence threshold ϵ is set and the algorithm terminates when $\theta \leq \epsilon$. At this state, class predictions for target data are stabled.

Algorithm 1 Subspace learning based on iterative **QMI-S**.

```
Input: Data matrix $X$=[$X_s$ ;$X_t$ ]∈ $\mathbb{R}^{n\times d}$ where source data $X_s \in \mathbb{R}^{n_s\times d}$ and target data $X_t \in \mathbb{R}^{n_t\times d}$, source data labels $[y_1, y_2, \ldots, y_{n_s}]^T$.
Output: $X_p \in \mathbb{R}^{n\times k}$, $k$-dimensional projected data.
Initialization: For $x_i^t \in X_t$, $P(c|x_i^t) = \frac{1}{N_c}$ for each $c \in \{1, 2, \ldots N_c\}$. For $x_i^s \in X_s$, $P(c|x_i^s) = 1$ if $c = y_i$ and $P(c|x_i^s) = 0$ otherwise, for each $c \in \{1, 2, \ldots N_c\}$.
Compute a centralized Gaussian kernel matrix, $K \in \mathbb{R}^{n\times n}$.
repeat
    Step-I:
    Compute $M'$ matrix using Eq.(3) and (5).
    Solve standard eigen problem, $M'V = V\Lambda$.
    Compute projected data $X_p = K^{\frac{1}{2}} V$.
    Step-II:
    Train a classifier $f$ using projected source data $X_p^s$ and apply it to update $P(c|x_i^t)$ with soft-labeling.
until convergence (defined in Section 3).
```

4. DATASET AND EXPERIMENTS

The proposed method is implemented and tested against popular benchmark datasets. **Office** is a widely used image database for domain adaptation [18]. It contains three different domains (Amazon, DSLR, Webcam) of images captured with varied settings and image conditions. Images of Amazon are downloaded from amazon site, DSLR contains images captured with high-resolution DSLR camera and Webcam contains images captured with low-resolution web camera. Additionally, a popular dataset for object recognition Caltech-256 [19] is used. The experiments will be conducted using these 4 domains with 10 common categories selected from each of them (Bike, BackPack, Calculator, Headphone, Keyboard, Laptop, Monitor, Mouse, Mug, Projector). From these 4 domains, a total of 12 DA sub-problems can be

Table 2: Comparative results in terms of classification accuracy(%) of target data for 12 different sub-problems. Each sub-problem consists of $source \rightarrow target$, where $source$ or $target$ represents any of the four domains: C(Caltech-256), A(Amazon), W(Webcam) and D(DSLR).

Methods	C → A	C → W	C → D	A → C	A → W	A → D	W → C	W → A	W → D	D → C	D → A	D → W	Avg
Origfeat	23.70	25.76	25.48	26.00	29.83	25.48	19.86	22.96	59.24	26.27	28.5	63.39	31.37
PCA	36.95	32.54	38.22	34.73	35.59	27.39	26.36	29.35	77.07	29.65	32.05	75.63	39.65
GFK	41.02	40.68	38.85	40.25	38.98	36.31	30.72	29.75	80.89	30.28	32.05	75.59	42.95
SA	42.07	32.2	45.86	39.8	37.63	36.94	28.76	34.34	88.54	32.5	34.24	88.47	45.11
TCA	45.82	30.51	35.67	40.07	35.25	34.39	29.92	28.81	85.99	32.06	31.42	86.44	43.03
TFL	44.78	41.69	45.22	39.36	37.97	39.49	**31.17**	32.78	89.17	31.52	33.09	**89.49**	46.31
TJM	46.76	38.98	44.59	39.45	42.03	**45.22**	30.19	29.96	**89.17**	31.43	32.78	85.42	46.33
QMI-H	55.95	49.49	45.86	**42.12**	42.71	37.58	30.37	35.8	80.89	35.71	38.31	61.02	46.32
QMI-S	**57.72**	**55.93**	**48.41**	41.76	**46.44**	38.85	30.72	**36.74**	83.44	**38.38**	**42.48**	77.63	**49.88**

created, each of which contains one source and one target domain.

Experimental setup The image representations published by Gong *et al.*[4] are used and the experimental protocol of [6, 7] is followed. Input data are whitened with PCA preserving 95% of the data variance. Gaussian kernel σ is set to median of the pair-wise distances of data in original feature space. A K-nn classifier with K set to $log(n_s) + 1$ heuristic [20] is used in line 9 of Algorithm 1. Convergence threshold $\epsilon = 1 \times 10^{-4}$ is used. The proposed **QMI-S** will be compared with 7 other methods (see Table 2). They can be categorized as follows,

- Without adaptation: **Origfeat** and **PCA** indicate the classification accuracy of target data in original feature space and PCA subspace respectively.
- Adaptation based on subspace alignment: It includes geodesic flow kernel (**GFK**) [4], subspace alignment (**SA**) [5], transfer component analysis (**TCA**) [21] and transfer feature learning (**TFL**) [7].
- Adaptation based on subspace alignment+instance reweighting: includes transfer joint matching (**TJM**) [6].

Analysis For each of the 12 sub-problems with source-target combination (C→A, C→W etc.), we reported the classification accuracy(%) of target domain data. For all methods, accuracy is determined by a one nearest neighbor classifier in the projected space or quoted from [6]. Our method shows improved performance compared to others and outperforms them in 7 out of 12 sub-problems. In terms of average accuracy over all 12 cases, our method is 3.75% ahead of the closest average accuracy (**TJM**). We also conducted experiments with hard-labeling (denoted as **QMI-H** in Table 2) for target predictions. Iterative **QMI-S** approach clearly shows its superiority as it conservatively updates target data labels, whereas **QMI-H** is aggressive and once a target point is falsely labeled, it is prone to stick with this label in successive iterations.

Figure 3 shows the affinity structure in the learned feature space for three different methods including ours for the sub-problem $C \rightarrow A$. Projected data are obtained using **Origfeat**, **TJM** and iterative **QMI-S** method. For illustration, 1051 projected data from 5 different classes are chosen with the first set is 584 data from source domain sorted by class and the second set is 467 data from target domain sorted by their predicted labels. A similarity matrix with 25-nearest neighbors is constructed (see Figure 3). The top-left and bottom-right sub-matrices of each sub-figure represent similarity inside a domain (source or target). Iterative **QMI-S** exhibits more compact block diagonal structure (Figure 3(c)) compared to **TJM** (Figure 3(b)). The top-right and bottom-left sub-matrices represent similarity across domains with better compact block diagonal structure generated by our method (which represents better within-class similarity).

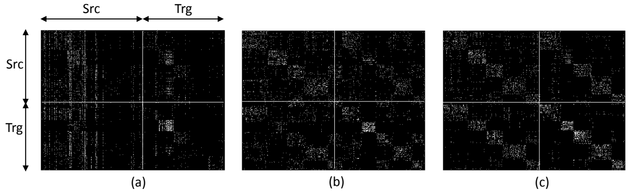

Fig. 3: Assesing the quality of feature subspace by similarity matrix for sub-problem $C \rightarrow A$ using (a) original feature space, (b) TJM and (c) iterative QMI-S.

In terms of computational cost, the average over all 12 sub-problems of the number of iterations of **QMI-S** until convergence is 25. The dominate cost in each iteration is solving an eigen value decomposition of a $n \times n$ for a set of largest eigenvalues and eigenvectors.

5. CONCLUSION

This paper proposes a domain adaptation algorithm based on soft-labeling induced quadratic mutual information. Unlike other subspace alignment methods, the goal is to utilize class conditional distribution of source domain to learn a common subspace with improved class separation such that a classifier trained with projected source data can be applied to projected target data effectively. In future work, we are planning to incorporate instance weighting into this framework in order to facilitate subspace learning only with closely related data samples across domains along with minimizing the impact of unrelated source samples.

6. REFERENCES

[1] Antonio Torralba, Alexei Efros, et al., "Unbiased look at dataset bias," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1521–1528.

[2] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola, "Correcting sample selection bias by unlabeled data," in *Proc. Advances in Neural Information Processing Systems*, 2006, pp. 601–608.

[3] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[4] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition, 2012*, 2012, pp. 2066–2073.

[5] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision, 2013*, 2013, pp. 2960–2967.

[6] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu, "Transfer joint matching for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1410–1417.

[7] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu, "Transfer feature learning with joint distribution adaptation," in *IEEE International Conference on Computer Vision, 2013*. IEEE, 2013, pp. 2200–2207.

[8] Jeff Donahue, Judy Hoffman, Erid Rodner, Kate Saenko, and Trevor Darrell, "Semi-supervised domain adaptation with instance constraints," in *IEEE Conference on Computer Vision and Pattern Recognition, 2013*, 2013, pp. 668–675.

[9] Hal Daumé III, Abhishek Kumar, and Avishek Saha, "Frustratingly easy semi-supervised domain adaptation," in *Proc. of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 2010, pp. 53–59.

[10] Minmin Chen, Kilian Q Weinberger, and John Blitzer, "Co-training for domain adaptation," in *Proc. Advances in Neural Information Processing Systems*, 2011, pp. 2456–2464.

[11] Wen-Sheng Chu, Fernando De la Torre, and Jeffrey F Cohn, "Selective transfer machine for personalized facial action unit detection," in *IEEE Conference on Computer Vision and Pattern Recognition, 2013*, 2013, pp. 3515–3522.

[12] Dimitrios Bouzas, Nikolaos Arvanitopoulos, and Anastasios Tefas, "Graph embedded nonparametric mutual information for supervised dimensionality reduction," *IEEE Trans. on Neural Networks and Learning Systems, vol. 26*, 2015.

[13] Kari Torkkola, "Feature extraction by non parametric mutual information maximization," *The Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, 2003.

[14] Yangqing Jia, Feiping Nie, and Changshui Zhang, "Trace ratio problem revisited," *IEEE Trans. on Neural Networks*, vol. 20, no. 4, pp. 729–735, 2009.

[15] Keinosuke Fukunaga, *Introduction to statistical pattern recognition*, Academic press, 2013.

[16] Jayaraman J Thiagarajan and Karthikeyan Natesan Ramamurthy, "Subspace learning using consensus on the grassmannian manifold," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2031–2035.

[17] Jihun Hamm and Daniel D Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *Proc. 25th International Conference on Machine learning*. ACM, 2008, pp. 376–383.

[18] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, "Adapting visual category models to new domains," in *Proc. ECCV 2010*, pp. 213–226. Springer, 2010.

[19] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, California Institute of Technology, 2007.

[20] Ulrike Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[21] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.