Adapting Instance Weights For Unsupervised Domain Adaptation Using Quadratic Mutual Information And Subspace Learning

M.N.A. Khan, Douglas R. Heisterkamp
Department of Computer Science
Oklahoma State University, Stillwater, OK
Email: {mohk, doug}@cs.okstate.edu

***Abstract*—Domain adaptation (DA) algorithms utilize a label-rich old dataset (domain) to build a machine learning model (classification, detection etc.) in a label-scarce new dataset with different data distribution. Recent approaches transform cross-domain data into a shared subspace by minimizing the shift between their marginal distributions. In this paper, we propose a novel iterative method to learn a common subspace based on non-parametric quadratic mutual information (QMI) between data and corresponding class labels. We extend a prior work of discriminative subspace learning based on maximization of QMI and integrate instance weighting into the QMI formulation. We propose an adaptive weighting model to identify relevant samples that share underlying similarity across domains and ignore irrelevant ones. Due to difficulty of applying cross-validation, an alternative strategy is integrated with the proposed algorithm to setup model parameters. A set of comprehensive experiments on benchmark datasets is conducted to prove the efficacy of our proposed framework over state-of-the-art approaches.**

I. Introduction

To build an object recognition or classification model, sufficient number of labeled images are necessary. Such models are tested using images sampled from the same distribution as training one. In real life scenario, training and testing data distributions might be different. Also collecting and annotating training data by manual intervention is often expensive, hence building a supervised machine learning model in a new domain becomes challenging. Therefore, the need for transferring knowledge from a related domain (*source*) to a novel one (*target*) becomes inevitable. One practical example is, learning a classification model using images captured with canonical viewpoints in a studio environment and deploying that application to recognize images taken in natural surroundings (see Figure 1). According to the literature, this problem area is widely known as *transfer learning*, *dataset bias*, *domain shift* or *domain adaptation* [1], [2], [3]. Researchers focus on leveraging a label-rich source domain to build an adaptive classifier for the target domain where i) label information is in poor supply and ii) marginal data distribution is different from source domain.

To deal with the domain adaptation 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 most challenging unsupervised case. Recent works in this area are mainly based on 1) subspace learning which focuses to learn a shared subspace by discovering the underlying common structures across domains and 2) instance weighting on source data to match their distribution with target data. Some works focused on unifying both strategies in a single framework and reported better performance than employing a single one [6], [10]. As an example of the first strategy, Gong *et al.* [4] proposed a kernel based method that models the underlying low-dimensional structure along the geodesic path from source to target domain. All available source data are utilized in their approach. Practically, not all the source data are useful for transfer learning [10] i.e. some source samples share very little structural similarity with target domain data. We propose an iterative framework for common subspace learning incorporated with instance weighting to model this scenario. An adaptive weight update recipe is applied to down-weight irrelevant source samples and up-weight cross-domain data with shared similarity.

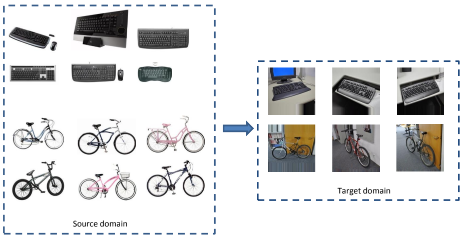


Fig. 1: Two domains with different data distributions.

Learning a common discriminative subspace by utilizing source information has been proved effective in domain adaptation context [11]. This work was inspired by [12], [13] and proposed a linear transformation based on maximization of non-parametric quadratic mutual information (QMI) between data and corresponding class labels. Also refinement of linear transformation with updated prediction of target data via soft-labeling (distribution of class labels for a point) was integrated with their algorithm. We improved their approach to leverage the benefit of discriminative subspace and induced instance weighting for source and target data in QMI formulation.

Nevertheless, we propose an unsupervised technique for setting model parameters, as traditional cross-validation approach is difficult due to unavailability of labeled data in target domain. The main contributions of this paper are summarized as follows,

(i) Learning a linear transformation to create common discriminative subspace based on maximization of QMI induced with instance weighting.

(ii) Proposing an adaptive weighting model to identify cross-domain data with shared similarity to assign them with higher weights compared to others.

(iii) Proposing an alternative approach of parameter selection in an unsupervised fashion without requiring labeled data from target domain.

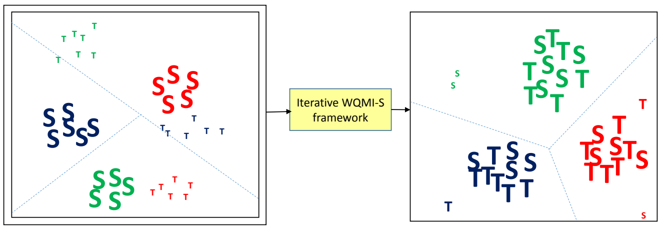
In Figure 2, the initial and final stage of the proposed iterative algorithm is illustrated.

II. Related work

Domain adaptation algorithms based on subspace learning has become popular in recent years [6], [5], [4], [7]. Most of these methods are developed based on the assumption of common underlying structure shared across domains. In [7], a common feature representation is proposed in a principled dimensionality reduction process. This work is further enhanced in [6] by introducing re-weighting of source samples to match them with target data. The motivation of our work is similar to [6]. However, we employed non-uniform instance weighting for cross-domain samples, as opposed to [6] where only source samples are weighted. Recent work based on landmark selection also focuses on identifying relevant samples from both source and target domains [10]. Their approach chooses landmarks by matching pair-wise samples across domains in kernel space and used them to align the two domains. Despite its efficiency, it is observed that landmark selection is a static process and requires to build the subspace from scratch in case of availability of new data. In contrast, we propose the selection of cross-domain similar instances via an adaptive weighting model. This model is coupled with subspace learning with an iterative feedback loop and leverages two benefits: i) refinement of the linear transformation through updated instance weights, and ii) assessing the cross-domain data similarity in the low-dimensional learned subspace, as opposed to kernel space [10].

A. *Prior work on QMI based domain adaptation*

Our work adopted the procedure of discriminative subspace learning by maximization of QMI from [11] i.e. the objective function for optimization and its solution process are adopted from this paper. We will provide a brief description of the derivation of objective function. According to information theoretic literature, *Mutual Information* (MI) is defined as a measure of independence 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. Also let $p(\boldsymbol{x})$ is the marginal density function of $\boldsymbol{x}$ and $P(c)$ is the class prior probability. Following Renyi's entropy, a non-parametric quadratic estimation of MI (denoted as quadratic mutual information or QMI) is defined as [13],



(a) Initialization (b) Final learned subspace

Fig. 2: 'S' and 'T' represent data from source and target domain respectively, colors represent different classes and font-size is proportional to individual sample weight. Initially target data are unknown and assigned with negligible or zero weights (left sub-figure). In the learned subspace (right sub-figure), data with shared similarity are closely projected, while ignoring irrelevant samples by down-weighting.

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

The subspace learning algorithm of [11] followed the trace-norm formulation of QMI proposed by Bouzas *et al.* [12]. They extended QMI by inducing soft class labeling, referred to as QMI-S.

Now $P(c)$, $p(\boldsymbol{x}, c)$ and $p(\boldsymbol{x})$ in Eq. (1) is evaluated by a Parzen window method based on Gaussian kernel. A Gaussian distribution function $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is,

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

and the Gaussian kernel matrix is $\boldsymbol{K}_g$ with $\boldsymbol{K}_g(i, j) = \mathcal{N}\left(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{0}, 2\sigma^2\boldsymbol{I}\right)$. Assume $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ represents d-dimensional data points of size n and $\boldsymbol{\Phi} \in \mathbb{R}^{n\times m}$ represents mapped data from raw feature space to a kernel Hilbert space, where m is the dimension of kernel space. A linear transformation function (alternatively, a projection matrix) is learned to create a k-dimensional subspace ($k < d$) by maximizing QMI-S between projected data and corresponding class labels. The final optimization objective takes the following form,

$$\boldsymbol{A}^* = \arg \max_{\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{A}=\boldsymbol{I}} \frac{\mathrm{tr}\{\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{M'}\boldsymbol{K}\boldsymbol{A}\}}{\mathrm{tr}\{\boldsymbol{A}^T\boldsymbol{K}\boldsymbol{K}\boldsymbol{A}\}} \tag{2}$$

$$\boldsymbol{M'} = (\boldsymbol{M} + \boldsymbol{M}^T)/2 \tag{3}$$

The matrix $\boldsymbol{M}$ is extracted from the formulation of QMI-S (see Eq. (2) of [11]). Also $\boldsymbol{W}$ is a projection matrix which is restricted to be in the range of $\boldsymbol{\Phi}$ i.e. $\boldsymbol{W} = \boldsymbol{\Phi}^T\boldsymbol{A}$, where $\boldsymbol{A} \in \mathbb{R}^{n\times k}$ is a co-efficient matrix. A 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 $\boldsymbol{E}_n \in \mathbb{R}^{n\times n}$ consists of elements each equal to $\frac{1}{n}$. See [12] for a detailed derivation of this objective function. The trace ratio problem of Eq. (2) can be approximated as a ratio trace optimization and solved for $\boldsymbol{A}^*$ using generalized eigen value decomposition method [14]. Hence Eq. (2) 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. Substituting $\boldsymbol{KU}$ with $\boldsymbol{V}$ and multiplying both sides by $\boldsymbol{K}^{-1}$, it becomes $\boldsymbol{M'V} = \boldsymbol{V\Lambda}$. 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 $\boldsymbol{A}^* = \boldsymbol{K}^{-\frac{1}{2}}\boldsymbol{V}$. Finally, the projected data $\boldsymbol{X}_p \in \mathbb{R}^{n\times k}$ will be computed as $\boldsymbol{X}_p = \boldsymbol{KA}^* = \boldsymbol{K}^{\frac{1}{2}}\boldsymbol{V}$.

III. QUADRATIC MUTUAL INFORMATION INTEGRATED WITH INSTANCE WEIGHTING

In [11], soft class labeling is induced in the QMI formulation (Eq. (1)). We further extend it by scaling the soft-labeling of a point with corresponding weight. We define weight of a sample $\boldsymbol{x}_i$ as the discrete probability distribution $P(\boldsymbol{x}_i)$ over the training samples $\boldsymbol{X}$ i.e. $w_i = P(\boldsymbol{x}_i)$. The expressions for $P(c)$, $p(\boldsymbol{x}, c)$ and $p(\boldsymbol{x})$ of Eq. (1) are evaluated as follows,

$$\begin{aligned}
p(\boldsymbol{x}) &= \sum_{i=1}^{n} P(\boldsymbol{x}_i)\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2\boldsymbol{I}\right) = \sum_{i=1}^{n} w_i\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2\boldsymbol{I}\right)\\
P(c) &= \sum_{i=1}^{n} P(c \mid \boldsymbol{x}_i) w_i = S_c\\
p(\boldsymbol{x}, c) &= P(c)p(\boldsymbol{x} \mid c)\\
&= \sum_{i=1}^{n} P(c|\boldsymbol{x}_i) w_i \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2\boldsymbol{I}\right)\\
&= \sum_{i=1}^{n} z_{c,i}\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2\boldsymbol{I}\right)
\end{aligned}$$

where $z_{c,i} = P(c|\boldsymbol{x}_i)w_i$ represents soft-labeling scaled with individual sample weight. Also for notational convenience, $P(c)$ is referred to as S_c. Now inspired by [11] and using the expressions for $p(\boldsymbol{x})$, $P(c)$ and $p(\boldsymbol{x}, c)$ derived above, we can further elaborate Eq. (1) (see Table I) which is rewritten as,

$$\begin{aligned}
\mathcal{I}(X, C) &= \operatorname{tr}\left\{\boldsymbol{\Phi}^T \left(\left(\sum_c \boldsymbol{z}_c \boldsymbol{z}_c^T\right) + \left(\sum_c S_c^2\right)\boldsymbol{w}\boldsymbol{w}^T - 2\boldsymbol{w}\left(\sum_c S_c \boldsymbol{z}_c^T\right)\right)\boldsymbol{\Phi}\right\}\\
&= \operatorname{tr}\{\boldsymbol{\Phi}^T \boldsymbol{M}\boldsymbol{\Phi}\}
\end{aligned}$$

where,

$$\boldsymbol{M} = \left(\sum_c \boldsymbol{z}_c \boldsymbol{z}_c^T\right) + \left(\sum_c S_c^2\right)\boldsymbol{w}\boldsymbol{w}^T - 2\cdot\boldsymbol{w}\left(\sum_c S_c\right)\boldsymbol{z}_c^T \qquad (4)$$

Here, $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]^T \in \mathbb{R}^{n\times 1}$ is a weight vector consisting of all sample weights with $\sum_{i=1}^n w_i = 1$ and $\boldsymbol{z}_c = [z_{c,1}, z_{c,2}, \ldots, z_{c,n}]^T \in \mathbb{R}^{n\times 1}$. We refer to this expression as WQMI-S. This is a generic formulation of QMI and by choosing $w_i = \frac{1}{n}$, we can obtain QMI-S of [11].

IV. PROPOSED DA FRAMEWORK BASED ON WQMI-S

We propose an iterative algorithm (referred to as iterative WQMI-S) based on subspace learning by maximization of WQMI-S. Input data $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ consists of source domain data $\boldsymbol{X}_s \in \mathbb{R}^{n_s\times d}$ and target domain data $\boldsymbol{X}_t \in \mathbb{R}^{n_t\times d}$. Ground truth labels of source data is $[y_1, y_2, \ldots, y_{n_s}]^T$, where n_s and n_t represent source and target data size respectively ($n = n_s + n_t$). As an initialization step of the algorithm, soft-labeling $P(c|\boldsymbol{x}_i)$ of a sample $\boldsymbol{x}_i$ and weight vector $\boldsymbol{w}$ are set. For source point $\boldsymbol{x}_i \in \boldsymbol{X}_s$,

$$P(c|\boldsymbol{x}_i) = \begin{cases} 1 & \text{if } c = y_i\\ 0 & \text{otherwise}\end{cases}$$

for each $c \in \{1, 2, \ldots, N_c\}$. On other hand, target data are unlabeled and hence are initialized with uniform label distribution i.e. for target point $\boldsymbol{x}_i \in \boldsymbol{X}_t$, $P(c|\boldsymbol{x}_i) = \frac{1}{N_c}$ for each $c \in \{1, 2, \ldots, N_c\}$. Initialization of $\boldsymbol{w}$ is subject to choice of an instance weighting model. We proposed one such model for $\boldsymbol{w}$ later in this section.

Each iteration of the proposed algorithm will learn a subspace, update label predictions of target data via soft-labeling and adjust instance weights. Subspace learning procedure is described earlier in Section II-A, where $\boldsymbol{M}$ will be used from Eq (4). After obtaining projected data, each target sample's prediction is updated by the weighted average prediction of its neighboring samples in WQMI-S subspace. Denoting $L_K(\boldsymbol{x})$ as the set of K neighboring points of a sample $\boldsymbol{x}$, the update rule will be following,

$$P(c|\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_j \in L_k(\boldsymbol{x}_i)} w_j P(c|\boldsymbol{x}_j) \quad \text{for each } \boldsymbol{x}_i \in \boldsymbol{X}_t \qquad (5)$$

Finally, instance weights are updated through a weight adaptation process.

A. Instance weight adaptation

We propose a weight adaptation formula to adjust weights of source and target samples at each iteration. Weight vector $\boldsymbol{w}$ is initialized: for each source sample $\boldsymbol{x}_i \in \boldsymbol{X}_s$, $w_i = \frac{1}{n_s}$ and for each target sample $\boldsymbol{x}_j \in \boldsymbol{X}_t$, $w_j = 0$. With this assignment, the initial learned subspace is dominated by $\boldsymbol{X}_s$ and further refined through out the iterations. The goal is to assign higher weights to relevant samples compared to others. To do this, a fraction from each instance weight is shrinked and then the total shrinked weight is re-distributed among *candidate* sets that contain cross-domain data with shared underlying similarity. Three sets of *candidate* points are defined as follows,

$$\begin{aligned}
\Omega_s &= \{\boldsymbol{x}|(\boldsymbol{x} \in \boldsymbol{X}_s) \cap (\boldsymbol{x} \in L_K(\boldsymbol{x}_i),\ \exists \boldsymbol{x}_i \in \boldsymbol{X}_t)\}\\
\Omega_{ta} &= \{\boldsymbol{x}|\boldsymbol{x} \in \boldsymbol{X}_t\}\\
\Omega_{th} &= \{\boldsymbol{x}|\boldsymbol{x} \in \boldsymbol{X}_t \cap \max P(c|\boldsymbol{x}) \geq \tau\}
\end{aligned}$$

τ is chosen to construct Ω_{th} which is defined as a set of target points with *confidence* in corresponding label predictions. Source and target data with underlying shared similarity are

TABLE I: Derivation of $\mathcal{V}_{in}$, $\mathcal{V}_{all}$ and $\mathcal{V}_{btw}$. Here $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$, $\boldsymbol{z}_c = [z_{c,1}, z_{c,2}, \ldots, z_{c,n}]^T \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]^T \in \mathbb{R}^{n\times 1}$

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

projected in a close proximity on the subspace. Hence, members of Ω_s (source *candidate* set) are up-weighted compared to other source data. The rest two sets consist of target data. Now weight shrinking and re-distributing take the following form,

$$w_i' = w_i - \alpha w_i. \tag{6}$$

$$w_i^{new} = \begin{cases} w_i' + \frac{\eta\alpha}{|\Omega_s|}, & \text{if } \boldsymbol{x}_i \in \Omega_s \\ w_i' + \frac{((1-\eta)\alpha)\beta}{|\Omega_{th}|}, & \text{if } \boldsymbol{x}_i \in \Omega_{th} \\ w_i' + \frac{((1-\eta)\alpha)(1-\beta)}{|\Omega_{ta}|}, & \text{if } \boldsymbol{x}_i \in \Omega_{ta}. \end{cases} \tag{7}$$

Here a fraction α is extracted from each instance weight (weight shrinking) resulting in total shrinked weight as $\sum_{i=1}^n \alpha w_i = \alpha$. This shrinked weight α is distributed among *candidate* points according to Eq. (7). The parameter η controls the partition of α among source and target *candidate* points and β controls the partition of allotted α among the members of Ω_{th} and Ω_{ta}. All target points are assigned with uniform weights (as members of Ω_{ta}) except some of them also gain 'bonus'weights (as members of Ω_{th}).

The weight mass is initially concentrated on source points and will eventually be shifted towards *candidate* points through out the iterations of iterative WQMI-S algorithm. This is a generic weighting scheme where α, β and η are chosen appropriately from $0 < \alpha, \eta, \beta \leq 1$.

B. Unsupervised parameter selection

To update weight vector $\boldsymbol{w}$ in iterative WQMI-S algorithm, parameters α, η, β and τ are set. For α, we argue that it is linearly proportional to the ratio $r = \frac{|\Omega_{th}|}{|\Omega_{ta}|}$. High value of r indicates the growth of target *confident* points and hence larger α is necessary to be shrinked for weight re-adjustment in target domain. In our implementation, $\alpha = \frac{r}{2}$ is used. Next, η is set to 0.5 to maintain a balance in weight re-adjustment among source and target *candidate* points. The 'bonus'weight assignment for the members of Ω_{th} is controlled by β. The intension is to assign *confident* target points with higher weights compared to other target ones. Any function characterizing this behavior would suffice to define β. In this work, $\beta = max(r, e^{-r})$ is used. Note that, when $|\Omega_{th}|$ is small, most points are non-confident and assigned with very small weights and vice versa. Lastly, τ is used to construct Ω_{th}. For tasks with moderate number of categories, $\tau = 0.7$ might be a simple choice. Instead in our work, an alternative strategy is applied. Note that, each iteration of iterative WQMI-S involves a subspace learning, update of label predictions for target data and weight vector $\boldsymbol{w}$. The idea is to unify subspace learning and weight update as follows,

- Choose parameter τ from a set of possible choices, e.g. $\{0.55, 0.6, 0.65, 0.7, 0.75\}$. Update $\boldsymbol{w}$ using Eq. (6), (7).
- Compute $\boldsymbol{M}$, learn a WQMI-S subspace and obtain projected data.
- K-means clustering is applied to projected target data for each WQMI-S subspace, where K is set to the number of unique categories. From each of these clusterings, a scatter metric $\mathcal{G} = \frac{J_b}{J_w}$ is computed with J_w and J_b representing trace-norm of within-cluster scatter and between-cluster scatter matrices respectively [15].

The idea is to select the *best* subspace generated by using corresponding τ. Assuming projected data will be tightly clustered around their class means, the WQMI-S subspace (alternatively, projected data) with maximum $\mathcal{G}$ is chosen. We refer to this scheme as *unsupervised parameter selection*, UPS (Figure 3). This projected data chosen in current iteration are utilized in next one for label predictions of target data. The overall approach is summarized in Algorithm 1.

C. Termination criteria

The algorithm will terminate when weight re-adjustment in successive iterations is negligible i.e. any of these two criteria is satisfied: i) weight change for each target point in successive iterations is negligible or ii) $\sum_{i=1}^{n_s} w_i = \sum_{j=1}^{n_t} w_j$ with sum of weights for non-candidate source points is negligible. After termination, a target point $\boldsymbol{x}$ is annotated with class c, where $c = \arg\max_c P(c|\boldsymbol{x})$.

V. Dataset and Experiments

'Office'is a widely used image database for domain adaptation [16]. 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'[17] 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,

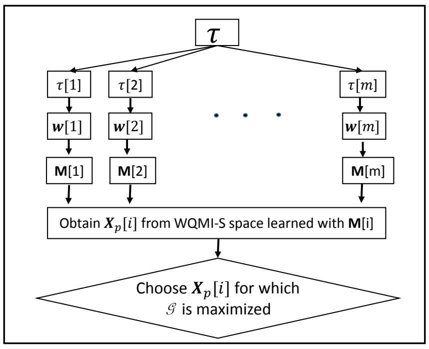

τ
$\tau[1]$
$\tau[2]$
$\tau[m]$
$w[1]$
$w[2]$
$w[m]$
M[1]
M[2]
M[m]
Obtain $X_p[i]$ from WQMI-S space learned with M[i]
Choose $X_p[i]$ for which $\mathcal{G}$ is maximized

Fig. 3: Block diagram for unsupervised parameter selection.

Algorithm 1 Iterative WQMI-S: Subspace learning algorithm with maximization of WQMI-S.

```
**Input**: Data matrix $X$=[$X_s$ ,$X_t$ ]$\in \mathbb{R}^{n\times d}$ where source domain data $X_s \in \mathbb{R}^{n_s \times d}$ and target domain data $X_t \in \mathbb{R}^{n_t \times d}$ with $n = n_s + n_t$, label vector for source data $[y_1, y_2, \dots, y_n]^T \in \mathbb{R}^{n_s \times 1}$.
**Output**: Projected data in final learned subspace $X_p \in \mathbb{R}^{n\times k}$.

Initialize weight vector $w$ and $P(c|x)$.
Compute a centralized Gaussian kernel matrix, $K \in \mathbb{R}^{n\times n}$.
Call eigenM($w$, $P(c|x)$) to obtain projected data $X_p$ .
**repeat**
    **for all** projected target point $x_i$ **do**
        Update $P(c|x_i)$ using Eq. (5).
    **end for**
    Choose possible values for $\tau$ e.g. $\tau \in \{0.55, 0.6, 0.65, 0.7, 0.75\}$.
    **for** each $\tau$ **do**
        Construct $\Omega_s$, $\Omega_{th}$ and $\Omega_{ta}$.
        Apply Eq. (6) and (7) to update $w$ and assign $w[i] \leftarrow w$.
        Call eigenM($w$[i], $P(c|x)$) to obtain projected data $X_p$ [i].
        Apply $K$-means clustering to the projected target data.
        Compute scatter metric $\mathcal{G}[i]$.
    **end for**
    Choose $X_p$ [j] and $w$[j] such that $j = \arg\max_j \mathcal{G}(j)$.
    Assign $X_p \leftarrow X_p[j]$ and $w \leftarrow w[j]$.
**until** termination criteria (Section IV-C) satisfied

**procedure** eigenM ($w$, $P(c|x)$)
    Compute $M'$ matrix using Eq.(4) and (3).
    Solve standard eigen problem, $M'V = V\Lambda$.
    Compute projected data $X_p = K^{\frac{1}{2}} V$.
```

Projector). Total 7 DA sub-problems are created, each of which contains one source and one target domain with $n_s > n_t$ or $n_s \approx n_t$. Our algorithm assumes a balance between source and target datasize, as smaller source domain is impractical for knowledge transfer.

Experimental setup The image representations published by Gong *et al.*[4] are used in our experiments. Also the experimental protocol of [6], [7] is followed. For other methods, a nearest neighbor classifier was trained with source data to classify target points. For fair comparison, each target point is annotated by the label of its nearest point in the learned subspace in iterative WQMI-S algorithm. Input data are whitened with PCA preserving 95% of the data variance. Following heuristic, σ of Gaussian kernel is set to median of the pair-wise distances of data in raw feature space [10]. Also K is set to $(log(n_s) + 1)$ [18] to construct L_k. The iterative WQMI-S algorithm is compared with 7 other methods (see Table II). They can be categorized as follows,

TABLE II: Comparative results in terms of classification accuracy(%) of target data for 7 different sub-problems of Office+Caltech dataset. Each sub-problem is in the form of $source \rightarrow target$, where C(Caltech-256), A(Amazon), W(Webcam) and D(DSLR) indicate four different domains.

Methods	C → A	C → W	C → D	A → C	A → W	A → D	W → D	Avg
Origfeat	23.70	25.76	25.48	26.00	29.83	25.48	59.24	30.78
PCA	36.95	32.54	38.22	34.73	35.59	27.39	77.07	40.36
GFK	41.02	40.68	38.85	40.25	38.98	36.31	80.89	45.28
TFL	44.78	41.69	45.22	39.36	37.97	39.49	**89.17**	48.24
TJM	46.76	38.98	44.59	39.45	42.03	45.22	**89.17**	49.46
QMI-S	**57.72**	55.93	48.41	**41.76**	46.44	38.85	83.44	53.22
WQMI-S	57.48	**58.64**	**50.07**	41.48	**50.51**	**45.61**	82.8	**55.23**

- 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] and transfer feature learning (TFL) [7].
- Adaptation based on subspace alignment + instance reweighting: it includes transfer joint matching (TJM) [6].

Pascal-Sun-Caltech dataset

Another experiment is conducted with three different datasets namely PASCAL2007(Ps), Caltech-101(Cl) and SUN09(Sn) [19]. Five common object classes (Bird, Car, Chair, Dog, Person) are selected from each of them. The image representations released by [19] are used, where each image is encoded with 5376 dimensional feature vector. Classification accuracy in target domain is reported in Table III.

Analysis For Office+Caltech dataset, the iterative WQMI-S method shows improved performance compared to other methods and outperforms them in 4 out of 7 sub-problems by significant margin. Also the average accuracy over all sub-problems is 2.01% higher than the second best performance. QMI-S is essentially a variant of WQMI-S, which is based on QMI maximization using soft-labeling with unweighted instances; accuracy of QMI-S is quoted from [11]. The accuracy data for GFK, TFL and TJM are quoted from [6] and [7]. It is noted that using both instance weighting and soft-labeling is very effective in learning a domain adaptive discriminative subspace. Similar behavior is observed for the Pascal-Sun-Caltech dataset (Table III). In this case, for GFK, TFL and TJM, classification accuracies are computed using different subspace dimensions and best results are reported. For both experiments, because of unsupervised clustering involved in UPS technique of our iterative WQMI-S algorithm, each sub-problem is run 5 times and the average accuracy is provided.

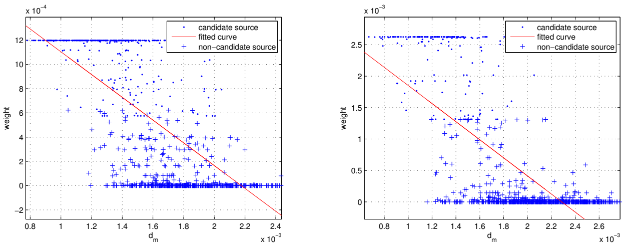
According to the proposed weight update model, source points residing in the neighborhood of target data in WQMI-S space are assigned with higher weights compared to other source points. As non-candidate source points are distantly projected from target data cloud, they are down-weighted through out the iterations. This phenomena is analyzed by measuring the minimum of distances between each source point and all the target points, i.e. for a source point $x_i \in X_s$,

TABLE III: Comparative results in terms of classification accuracy(%) of target data for 4 different sub-problems of Pascal-Sun-Caltech dataset. Each sub-problem is in the form of *source* → *target*, where Cl(Caltech-101), Sn(SUN '09), Ps(Pascal 2007) indicate three different domains.

Methods	Sn → Cl	Sn → Ps	Ps → Cl	Ps → Sn	Avg
Origfeat	20.28	28.92	55.51	41.3	36.5
PCA	21.59	31.03	64.25	42.17	39.76
GFK	33.65	35.06	64.86	**44.71**	44.57
TFL	39.16	39.71	61.89	44.08	46.21
TJM	41.00	35.68	57.34	41.35	43.84
QMI-S	38.46	39.19	62.24	43.46	45.84
WQMI-S	**42.31**	**42.29**	**69.14**	44.56	**49.58**

$d_m(\boldsymbol{x}_i) = \min dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$, for $\forall \boldsymbol{x}_j \in \boldsymbol{X}_t$. Here $dist(.,.)$ measures a distance between two points. The weight of $\boldsymbol{x}_i$ and $d_m(\boldsymbol{x}_i)$ follow negative correlation i.e. the increase of $d_m(\boldsymbol{x}_i)$ results in down-weighting of corresponding $\boldsymbol{x}_i$ and vice versa. This behavior is illustrated in Figure 4 for two sub-problems ($A \to W$, $A \to D$). The line fitted in $d_m(\boldsymbol{x}_i)$ vs. weight scatter plot indicates that source *candidate* points are higher weighted and projected in close proximity of target data (hence $d_m(\boldsymbol{x}_i)$ is lower). Weight decreases with the growth of $dm(\boldsymbol{x}_i)$ for *non-candidate* points.

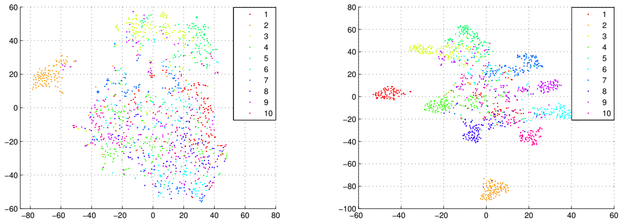
Finally, we used a visualization tool named t-SNE [20] to plot the projected source and target samples in 2d space (Figure 5). Target data are plotted with corresponding predicted labels. It is observed that data in WQMI-S subspace is tightly clustered compared to TJM. This also supports the unsupervised clustering of UPS process as a reasonable alternative strategy of parameter selection.



(a) $A \to W$

(b) $A \to D$

Fig. 4: Scatter plot for weight vs. d_m value of each source point $\boldsymbol{x}_i \in \boldsymbol{X}_s$ in two sub-problems, where $d_m(\boldsymbol{x}_i) = \min dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$, for $\forall \boldsymbol{x}_j \in \boldsymbol{X}_t$.



(a) $\mathcal{G} = 146.36$ (TJM)

(b) $\mathcal{G} = 296.62$ (WQMI-S)

Fig. 5: 2d visuallization of source and target data in learned subspace using t-SNE [20] for sub-problem $A \to W$ using two different methods (a) TJM and (b)WQMI-S. Values of scatter metric $\mathcal{G}$ are also provided.

VI. Conclusion

We proposed a subspace learning framework for domain adaptation based on maximization of quadratic mutual information between data and class labels. A generic formulation of QMI is proposed incorporating soft-class labeling and instance weights. Both source and target data are weighted based on their underlying shared similarity. Source data that share little similarity with target distribution are down-weighted to reduce their impact on subspace learning. Through an iterative refinement, the linear transformation to build a common subspace is optimized with the use of relevant samples across domains. An adaptive instance weighting model is provided to identify such samples automatically. We propose an alternative strategy to deal with model parameter setup without requiring labeled data from target domain. In future, we plan to extend this work for detection of novel category not present in training phase.

References

[1] A. Torralba, A. Efros *et al.*, "Unbiased look at dataset bias," in *In Proc. of IEEE Conference on CVPR, 2011*, pp. 1521–1528.

[2] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *In Proc. of Advances in Neural Information Processing Systems, 2006*, pp. 601–608.

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

[4] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *In Proc. of IEEE Conference on CVPR, 2012*.

[5] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *In Proc. of IEEE Conference on CVPR, 2013*, pp. 2960–2967.

[6] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *In Proc. of IEEE Conference on CVPR, 2014*, pp. 1410–1417.

[7] ——, "Transfer feature learning with joint distribution adaptation," in *In Proc. of IEEE Conference on ICCV, 2013*.

[8] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *In Proc. of IEEE Conference on CVPR, 2013*, pp. 668–675.

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

[10] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban, "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation," in *In Proc. of IEEE Conference on CVPR, 2015*, pp. 56–63.

[11] D. R. H. M.N.A. Khan, "Domain adaptation by iterative improvement of soft-labeling and maximization of non-parametric mutual information," in *In Proc. of IEEE Conference on ICIP, 2016*.

[12] D. Bouzas, N. Arvanitopoulos, and A. Tefas, "Graph embedded non-parametric mutual information for supervised dimensionality reduction," *IEEE Trans. on Neural Networks and Learning Systems, vol. 26*, 2015.

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

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

[15] L. Rokach and O. Maimon, *Data Mining and Knowledge Discovery Handbook*. Springer US, 2005, ch. Clustering Methods, pp. 321–352.

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

[17] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

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

[19] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *In Proc. of IEEE Conference on ECCV 2012*. Springer, 2012, pp. 158–171.

[20] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.