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.

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 corre-
sponding 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 $x \in \mathbb{R}^d$ 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 $p(x)$ be the density function of $x$ 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],

$$I(X, C) = \sum_c \int x (p(x, c) - P(c)p(x))^2 dx$$

$$= \sum_c \int x p(x, c)^2 dx + \sum_c \int x P(c)^2 p(x)^2 dx$$

$$- 2 \sum_c \int x p(x, c) P(c) p(x) dx$$

$$= V_{in} + V_{all} - 2 V_{btw} \tag{1}$$

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

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

A Parzen window density estimation of $p(x)$ with IID samples $x_i$ is,

$$p(x) = \frac{1}{n} \sum_{i=1}^{n} N(x; x_i, \sigma^2 I)$$

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

$$P(c) = \sum_{i=1}^{n} \frac{P(c|x_i) P(x_i)}{\sum_{i=1}^{n} P(c|x_i) \frac{1}{n} = S_c}$$

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

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

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

$$I(X, C) = tr \{ \Phi^T M \Phi \} \tag{2}$$

where

$$M = \frac{1}{n^2} \left( \sum_c z_c z_c^T \right) + \left( \sum_c \frac{S_c}{n^2} \right) I - 2 \cdot 1 \left( \sum_c \frac{S_c}{n^2} \right) z_c^T \tag{3}$$

where $z_c = [P(c|x_1), P(c|x_2), \ldots, P(c|x_n)]^T \in \mathbb{R}^{n \times 1}$ is the soft-labeling introduced into QMI and $\Phi \in \mathbb{R}^{n \times m}$ 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., $\Phi = \Phi^T$. This formulation is referred to as QMI-S.

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

$$A^* = \arg\max_{A^T K A = I} \frac{tr \{ A^T K M' K A \}}{tr \{ A^T K K A \}} \tag{4}$$

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

where $M'$ is a symmetric form of $M$. Centralized Gaussian kernel $K \in \mathbb{R}^{n \times n}$ is defined as $K = K_g - E_n K_g - K_g E_n + E_n K_g E_n$, where $K_g(i, j) = N(x_i - x_j; 0, 2\sigma^2 I)$ and $E_n \in \mathbb{R}^{n \times n}$ 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 $A^*$ is found using the generalized eigenvalue decomposition method [15]. After finding $A^*$, the data is projected by $X_p = \Phi W = K A^*$. 


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

The data are available as \( X \in \mathbb{R}^{n \times d} \) consisting of source domain data \( X_s \in \mathbb{R}^{n_s \times d} \), target domain data \( X_t \in \mathbb{R}^{n_t \times d} \) 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 \( n = n_s + n_t \). \( P(c|x_i) \) is defined as follows. For \( x_i \in X_s \), labels are known so hard labeling can be used (i.e., \( P(c|x_i) = 1 \) if \( c = y_i \) else 0). On the other hand, target data are unlabeled, so a full distribution is used. For target data, \( P(c|x_i) \) is initialized with uniform label distribution i.e., for \( x_i \in X_t \), \( P(c|x_i) = \frac{1}{N_c} \) for each \( c \in \{1, 2, \ldots, N_c\} \). If source labels are from a classifier instead of ground truth then the classifier’s \( P(c|x_i) \) can be used instead of hard labels.

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

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

**Step-II:** Target data predictions \( P(c|x_p^t) \) are updated by applying a classifier trained with projected source data. This will eventually update \( M' \) of Step-I for the next iteration. \( P(c|x_p^t) \) 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 \( V \). The difference between two \( k \)-dimensional subspaces can be approximated as a subspace distance on a Grassmannian [16]. One such distance metric measures the principal angle \( \theta \) between \( V_i \) and \( V_{i+1} \) of iterations \( i \) and \( i + 1 \) respectively [17, 16]. A convergence threshold \( \epsilon \) is set and the algorithm terminates when \( \theta \leq \epsilon \). At this state, class predictions for target data are stable.

**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, 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).

<table>
<thead>
<tr>
<th>Methods</th>
<th>C $\rightarrow$ A</th>
<th>C $\rightarrow$ W</th>
<th>C $\rightarrow$ D</th>
<th>A $\rightarrow$ C</th>
<th>A $\rightarrow$ W</th>
<th>A $\rightarrow$ D</th>
<th>W $\rightarrow$ C</th>
<th>W $\rightarrow$ A</th>
<th>W $\rightarrow$ D</th>
<th>D $\rightarrow$ C</th>
<th>D $\rightarrow$ A</th>
<th>D $\rightarrow$ W</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origfeat</td>
<td>23.70</td>
<td>25.76</td>
<td>25.48</td>
<td>26.00</td>
<td>29.83</td>
<td>25.48</td>
<td>19.86</td>
<td>22.96</td>
<td>59.24</td>
<td>26.27</td>
<td>28.5</td>
<td>63.39</td>
<td>31.37</td>
</tr>
<tr>
<td>PCA</td>
<td>36.95</td>
<td>32.54</td>
<td>38.22</td>
<td>34.73</td>
<td>35.59</td>
<td>27.39</td>
<td>26.36</td>
<td>29.35</td>
<td>77.07</td>
<td>29.65</td>
<td>32.05</td>
<td>75.63</td>
<td>39.65</td>
</tr>
<tr>
<td>GFK</td>
<td>41.02</td>
<td>40.68</td>
<td>38.88</td>
<td>40.25</td>
<td>38.98</td>
<td>36.31</td>
<td>30.72</td>
<td>29.75</td>
<td>80.89</td>
<td>30.28</td>
<td>32.05</td>
<td>75.59</td>
<td>42.95</td>
</tr>
<tr>
<td>SA</td>
<td>42.07</td>
<td>37.32</td>
<td>45.86</td>
<td>39.8</td>
<td>37.63</td>
<td>36.94</td>
<td>28.76</td>
<td>34.34</td>
<td>88.54</td>
<td>32.78</td>
<td>34.24</td>
<td>88.47</td>
<td>45.11</td>
</tr>
<tr>
<td>RCA</td>
<td>45.82</td>
<td>30.51</td>
<td>35.67</td>
<td>40.07</td>
<td>35.25</td>
<td>34.19</td>
<td>29.92</td>
<td>28.81</td>
<td>85.99</td>
<td>32.06</td>
<td>31.42</td>
<td>86.44</td>
<td>43.03</td>
</tr>
<tr>
<td>TFL</td>
<td>46.76</td>
<td>38.98</td>
<td>44.59</td>
<td>39.45</td>
<td>42.03</td>
<td>45.22</td>
<td>30.19</td>
<td>29.96</td>
<td>89.17</td>
<td>31.43</td>
<td>32.78</td>
<td>85.42</td>
<td>46.33</td>
</tr>
<tr>
<td>QMI-H</td>
<td>55.95</td>
<td>49.49</td>
<td>45.86</td>
<td>42.12</td>
<td>42.71</td>
<td>37.58</td>
<td>30.37</td>
<td>35.8</td>
<td>80.89</td>
<td>35.71</td>
<td>38.31</td>
<td>61.02</td>
<td>46.32</td>
</tr>
<tr>
<td>QMI-S</td>
<td>57.72</td>
<td>55.93</td>
<td>48.41</td>
<td>41.76</td>
<td>46.44</td>
<td>38.85</td>
<td>30.72</td>
<td>36.74</td>
<td>83.44</td>
<td>38.38</td>
<td>42.48</td>
<td>77.63</td>
<td>49.88</td>
</tr>
</tbody>
</table>

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 $\sigma$ is set to median of the pair-wise distances of data in original feature space. A $K$-nn classifier with $K$ set to $\log(n_A) + 1$ 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 re-weighting: includes transfer joint matching (TJM) [6].

Analysis For each of the 12 sub-problems with source-target combination (C$\rightarrow$A, C$\rightarrow$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 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).

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


