IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received March 7, 2019, accepted March 17, 2019, date of publication March 27, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907571

A New Transfer Learning Algorithm in Semi-Supervised Setting

RAKESH KUMAR SANODIYA[1], (Member, IEEE), JIMSON MATHEW[1], (Senior Member, IEEE), SRIPARNA SAHA[1], (Senior Member, IEEE), AND MICHELLE DAVIES THALAKOTTUR[2]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801103, India
[2]Department of Computer Engineering, MKSSS Cummins College of Engineering, Pune 411052, India

Corresponding author: Rakesh Kumar Sanodiya (rakesh.pcs16@iitp.ac.in)

ABSTRACT Transfer Learning is an effective method of dealing with real-world problems where the training and test data are drawn from different distributions. Transfer learning methods use a labeled source domain to boost the task in a target domain that may be unsupervised or semi-supervised. However, the previous transfer learning algorithms use Euclidean distance or Mahalanobis distance formula to represent the relationships between instances and to try and capture the geometry of the manifold. In many real-world scenarios, this is not enough and these functions fail to capture the intrinsic geometry of the manifold that the data exists in. In this paper, we propose a transfer learning framework called Semi-Supervised Metric Transfer Learning with Relative Constraints (SSMTR), that uses distance metric learning with a set of relative distance constraints that capture the similarities and dissimilarities between the source and the target domains better. In SSMTR, instance weights are learned for different domains which are then used to reduce the domain shift while a Relative Distance metric is learned in parallel. We have developed SSMTR for classification problems as well, and have conducted extensive experiments on several real-world datasets; particularly, the PIE Face, Office-Caltech, and USPS-MNIST datasets to verify the accuracy of our proposed algorithm when compared to the current transfer learning algorithms.

INDEX TERMS Transfer Learning, metric learning, semi-supervised learning, relative distance constraints.

I. INTRODUCTION

The main assumption made in classical statistical learning is that the training and test data are drawn from the same data distribution. However, in many real-world applications, test examples are usually in a different context than the training data. Then new test samples need to be collected for training a classifier. For example, classic object category recognition requires a large number of training examples to ensure good generalization on test problems [1]. Usually, there exists some domain with a scarcity of labelled data, often called the target domain, and some related domain with an abundance of labelled data, often called the source data. The marginal and conditional variations in domain distributions are rarely small and hence, a source domain classifier cannot be applied directly to the target domain data. Minimizing the distribution shifts is necessary to accurately classify a given task and the task of learning a discriminative model by shifting the source and target domain data distributions is known as Transfer Learning (TA) [2]. However, previous approaches to TA have faced the following difficulties: a) the domain shift increases with minute changes in factors like the environment, etc. b) real-world datasets are expensive to manually review and label and there exist abundance of unlabelled target domain data and c) the geometries of the source and target domain data are not captured well and some information of the manifold is lost when re-weighting the domains or projecting them into a common subspace.

Based on the type of target domain data that is available, transfer learning algorithms can be classified as semi-supervised or unsupervised [3]. Semi-supervised transfer learning has well-labelled source domain data and a small amount of labelled target domain data while the rest of the target data is unlabelled [4]. Unsupervised transfer learning has well-labelled source domain data and unlabelled target domain data. Three common approaches to transfer learning are feature-based, instance-based and metric learning based [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Mohsin Jamil.

Feature-based transfer learning focuses on minimizing domain shift by either a) learning a transformation for the source and target domains so as to project the data from the two domains into a common subspace or b) only looks at the shared features of both domains, which can fail when the domain shift is large. Feature-based transfer learning is explored in [6]–[11] and [12]. Instance-based transfer learning involves re-weighting the labelled data in the source domain to better match the target domain data distribution. Instance-based transfer learning is used in [13]–[15], and [16]. Metric-Learning based transfer learning involves learning the metric of the target domain with the help of source domain. Metric-Learning based transfer learning is explored in [17]–[23] and [24].

The existing transfer learning algorithms generally make use of either the Euclidean distance formula or the Mahalanobis distance formula to capture the intrinsic geometry of the manifold and preserve it while projecting the source and target domain data to a subspace or while re-weighting the source domain to better match that of the target domain. The Euclidean and Mahalanobis distance formulas cannot accurately represent the similarities and dissimilarities of the domain in a manifold and this leads to an inaccurate representation of the source-target relationship and can lead to a limited knowledge transfer that reduces the accuracy of the transfer learning algorithm [24]. To deal with this problem we have developed a semi-supervised transfer metric learning framework called SSMTR that uses relative distance constraints to estimate a relative distance metric to find better projection of the data and the KL-divergence to minimize the distribution between the source and target domain data.

The major contributions of this paper are:

- To the best of our knowledge (after conducting a thorough literature survey), our proposed approach is a unique attempt in solving the problems detailed above, i.e. in the research gap, by estimating the relative distance metric and minimizing the distribution between both source and target domain are carried out simultaneously.
- In this paper, we have used Distance Metric Learning with Relative Distance comparison constraints instead of a Mahalanobis distance metric that uses Must-Link (ML) and Cannot-Link (CL) constraints which are commonly used in other algorithms (seen in the literature survey), and thus our approach is capable of quantifying the appropriate geometry of the data in different domains.
- We have compared the performance of SSMTR with 11 other transfer learning algorithms on the PIE Face, Office-Caltech, and USPS-MNIST datasets and our results have shown that the proposed algorithm achieves much greater accuracy when compared to other algorithms.
- Our proposed SSMTR approach achieved 77.94% mean accuracy for all tasks of the PIE Face dataset while none of the compared approaches achieved more than 43.74% accuracy.

In the following sections, we have detailed our proposed method and the results obtained while comparing it to other state of the art transfer learning methods. Section II we have presented an overview of relevant related literature and Section III explains the proposed method, in which we have detailed the problem statement as well as the proposed framework. In the next section, Section IV we have solved the optimization problem presented in Section III to arrive at the objective function for SSMTR. In Section V, we have first described the benchmark datasets that have been used to compare the results obtained by the proposed method with other state of the art methods. We have then analyzed parameter sensitivity of the proposed method and then analyzed the results obtained on the benchmark datasets. We have also provided a time complexity analysis of our method and compared it with other algorithms. Table 1 compares SSMTR with other algorithms and highlights the improved accuracy.

II. RELATED WORKS

In this section, we have discussed different transfer learning algorithms that are related to ours, and highlighted their differences from our proposed method. After conducting a thorough literature survey [25], we see that transfer learning approaches can be classified into three categories: feature-based, instance-based and transfer metric based transfer learning.

In the first category, a feature space is found where the divergence between the data distributions of the source and target domains is made the minimum. Pan et al. [6] proposed two algorithms, TCA (Transfer Component Analysis) and SSTCA (Semi-supervised Transfer Component Analysis) which minimize the distance between the domains means in a Reproducing Kernel Hilbert Space (RKHS) using Maximum Mean Discrepancy (MMD). The JDA (Joint Distribution Adaption) algorithm, proposed by Long *et al.* [7] improves on TCA and minimizes the marginal and conditional distribution shift between domains using principled dimensionality reduction methods like PCA (Principal Component Analysis). TJM (Transfer Joint Matching), proposed by Long *et al.* [8], addresses the issue of having large domain shift and thus improves on TCA. TJM identifies and re-weights the instances that are common across domains by jointly matching the features. It then constructs a feature representation in a RKHS using MMD, by using features that are common to both domains so that the subspace has minimum distribution and domain shift. In [9], Si *et al.* proposed algorithms that transfer knowledge from source to target domain by minimizing the Bregman divergence between the two distributions. GFK (Geodesic Flow Kernel), proposed by Gong *et al.* [10], is a kernel-based method that considers an infinite number of subspaces and models marginal and distributional shifts between domains. In [11], Zhang *et al.* proposed an algorithm called JGSA (Joint Geometrical and Statistical Alignment) that reduces the shift between domains

TABLE 1. Comparison of the accuracies of our proposed approach with various state of art algorithms over different tasks of the datasets.

PIE Face Dataset											
Tasks	**MTLF**	**TJM**	**JDA**	**GFK**	**JGSA**	**TCA**	**SSTCA**	**SMIDA**	**MIDA**	**TML**	**STML**	**SSMTR (Proposed)**
29_07	42.05±0.6	15.10 ±0.7	14.79±0.3	17.00 ±0.5	38.12±0.4	09.76±0.3	22.77±0.2	16.32 ±0.3	7.36±0.8	27.86±0.8	07.18±0.6	**74.27**±0.5
29_27	40.22±0.6	22.52±0.5	23.28±0.4	14.26±0.8	20.81±0.9	09.64±0.7	22.85±0.5	04.83±0.6	04.17±0.4	24.03±0.5	05.25±0.3	**77.50**±0.7
29_09	45.46±0.3	10.23±0.3	10.53±0.7	16.36±0.8	29.59±0.7	08.14±0.3	24.26±0.6	11.45±0.5	05.33±0.8	35.53±0.4	05.57±0.3	**76.28**±0.7
29_05	36.73±0.3	23.22±0.3	26.5±0.6	9.87±0.5	15.12±0.7	05.58±0.8	18.09±0.7	04.80±0.3	04.86±0.9	20.25±0.5	04.17±0.4	**73.67**±0.6
07_05	41.32±0.3	26.8±0.9	27.37±0.4	15.69±0.3	21.81±0.8	10.41±0.9	23.13±0.7	04.74±0.7	05.49±0.8	17.52±0.3	05.25±0.8	**76.53**±0.8
07_27	49.35±0.4	28.95±0.6	31.12±0.6	27.81±0.7	28.89±0.5	21.35±0.5	31.99±0.4	05.16±0.6	04.53±0.9	20.03±0.7	08.68±0.3	**83.95**±0.4
07_29	45.77±0.8	10.66±0.8	11.02±0.5	15.80±0.5	26.71±0.6	09.43±0.7	26.22±0.8	10.66±0.4	07.29±0.5	33.08±0.8	06.92±0.4	**77.94**±0.5
07_09	55.02±0.6	19.36±0.5	21.07±0.3	24.93±0.7	39.58±0.6	14.15±0.7	32.90±0.3	11.33±0.6	10.47±0.5	32.41±0.8	09.92±0.9	**82.10**±0.9
09_07	45.54±0.8	21.6±0.8	22.03±0.9	22.46±0.6	37.87±0.5	15.46±0.4	30.57±0.5	08.71±0.4	09.02±0.4	25.35±0.8	12.09±0.7	**79.68**±0.4
09_05	37.03±0.8	24.57±0.7	22.95±0.7	15.87±0.5	18.78±0.5	07.50±0.6	17.40±0.6	06.45±0.8	04.65±0.8	18.93±0.5	04.92±0.4	**73.19**±0.5
09_27	47.97±0.5	25.14±0.5	27.18±0.3	24.36±0.6	28.38±0.8	14.74±0.6	26.70±0.9	04.74±0.5	04.08±0.5	20.27±0.7	08.23±0.7	**84.16**±0.4
09_29	43.93±0.6	9.92±0.5	12.62±0.3	12.37±0.5	31.92±0.7	07.41±0.8	28.73±0.6	06.55±0.9	04.04±0.7	18.38±0.4	03.92±0.8	**75.36**±0.9
05_29	44.11±0.4	11.09±0.4	12.37±0.5	07.41±0.5	22.73±0.6	04.53±0.5	21.44±0.8	04.22±0.7	03.24±0.7	21.07±0.6	06.49±0.5	**77.08**±0.5
05_09	43.68±0.8	18.50±0.6	18.32±0.8	16.66±0.4	29.71±0.3	05.20±0.5	21.13±0.3	09.55±0.8	04.10±0.4	24.44±0.9	04.59±0.7	**79.65**±0.8
05_27	39.65±0.5	26.70±0.4	27.87±0.6	15.47±0.6	29.28±0.7	13.00±0.5	21.56±0.5	07.90±0.6	06.33±0.9	19.67±0.8	03.42±0.3	**77.53**±0.8
05_07	42.05±0.4	19.09±0.6	19.58±0.2	13.32±0.6	30.14±0.65	08.41±0.8	23.26±0.9	04.17±0.4	03.31±0.7	30.14±0.9	03.62±0.4	**78.20**±0.9
Office-Caltech Dataset												
W_D	57.32±0.6	68.78±0.4	67.51±0.3	61.14±0.4	53.50±0.5	63.69±0.7	**75.79**±0.4	68.15±0.3	55.41±0.8	38.85±0.7	39.49±0.7	**75.79**±0.6
W_C	27.11±0.7	26.17±0.7	26.53±0.4	25.82±0.5	13.35±0.3	21.28±0.5	26.00±0.8	25.64±0.4	24.57±0.3	15.85±0.4	14.78±0.6	**28.22**±0.7
W_A	31.41±0.3	26.82±0.3	29.95±0.7	26.09±0.8	15.03±0.6	20.98±0.5	34.55±0.8	32.15±0.6	22.54±0.4	22.02±0.6	18.26±0.7	**36.22**±0.3
A_W	45.76±0.6	34.23±0.5	33.55±0.7	26.44±0.5	22.71 ±0.4	26.77±0.3	52.18±0.6	41.35±0.7	20.33±0.8	30.84±0.6	24.40±0.5	**52.54**±0.4
A_D	37.57±0.3	35.03±0.4	36.30±0.6	29.93±0.6	28.02±0.4	32.48±0.5	**54.77**±0.4	54.14±0.7	33.75±0.8	35.03±0.5	26.75±0.3	49.04±0.8
A_C	32.41±0.3	31.34±0.4	30.36±0.8	28.76±0.7	11.84±0.5	29.65±0.5	34.63±0.8	34.90±0.7	27.60±0.5	21.01±0.5	18.52±0.45	**37.47**±0.7
C_A	35.07±0.7	33.4±0.3	32.88±0.4	27.24±0.6	11.48±0.8	26.93±0.5	36.11±0.6	35.69±0.7	30.68±0.5	24.21±0.3	20.04±0.5	**37.32**±0.5
C_W	43.38±0.7	27.11±0.6	27.45±0.5	13.55±0.5	13.89±0.8	15.25±0.6	44.06±0.3	40.00±0.8	15.59±0.4	24.74±0.3	15.59±0.3	**47.45**±0.3
C_D	07.64±0.4	33.75±0.4	32.48±0.5	26.11±0.3	19.74±0.6	18.47±0.8	41.40±0.3	35.66±0.5	19.74±0.4	31.21±0.4	14.64±0.7	**47.45**±0.2
D_C	27.07±0.7	25.46±0.3	25.46±0.7	25.28±0.5	13.62±0.4	22.35±0.8	27.51±0.4	27.51±0.3	21.90±0.5	17.98±0.4	13.08±0.5	**28.94**±0.7
D_A	33.08±0.6	31.31±0.2	31.31±0.4	23.79±0.4	19.72±0.3	24.21±0.7	36.32±0.7	31.00±0.9	26.93±0.6	24.32±0.3	17.11±0.8	**38.62**±0.5
D_W	52.54±0.5	65.08±0.3	64.74±0.4	52.20±0.3	53.55±0.5	49.15±0.7	65.76±0.8	66.10±0.5	49.83±0.2	34.91±0.4	37.96±0.6	**67.46**±0.4
Handwritten Digit Recognition (USPS, MNIST)												
U_M	67.20±0.4	57.60±0.3	58.55±0.5	42.40±0.3	47.40±0.4	46.45±0.5	61.50±0.7	36.70±0.3	36.10±0.6	33.60±0.2	38.70±0.6	**74.80**±0.5
M_U	83.83±0.4	72.11±0.5	71.50±0.3	58.00±0.3	54.22±0.2	54.88±0.6	69.61±0.4	57.22±0.3	57.77±0.5	34.27±0.2	33.72±0.2	**84.33**±0.6

both statistically and geometrically simultaneously. JGSA projects the source and target domain into lower dimensional subspaces while reducing the domain shift simultaneously. MIDA (Maximum Independence Domain Adaptation) and SMIDA (Semi-supervised MIDA), proposed by Yan *et al.* in [12], are for domain adaptation in the field of sensors and measurement. Their proposed algorithm treated instrumental variation and time-varying drift to be a discrete and continuous distributional change in the feature space.

In the second category, the objective is to re-weight the samples in the source domain to better match the target domain distribution. Bickel *et al.* [13] proposed a novel approach to discriminative learning using Co-variate Shift. In [14], data from both domains are aligned to the same space and then the weights of source domain are adjusted to better match those of the target domain. KLIEP, proposed by Sugiyamate at al. [15] directly estimates the importance of source domain data points by minimizing the Kullback-Leibler divergence from the actual input density to its estimate. Zhang *et al.* [16] proposed an algorithm that takes an adversarial-based approach by using a weighted adversarial net-based method when the source domain has a larger number of classes compared to the target data.

In the third category of transfer learning, the first model for metric learning was introduced by Zha *et al.* [17] by considering that the source task has sufficient labelled information in the form of prior metric and that the target task has some labelled information. Zha *et al.* proposed TML (Transfer Metric Learning) [18] and STML (Semi-supervised TML) [19] that use co-relations between tasks to formulate a task relationship between source and target task which boosts the performance of the target task. Oh *et al.* [20] used lifted structured feature embedding and pairwise distance constraints to learn a distance metric that is then used to train the neural network. This is a deep learning metric learning method. In [26], Sanodiya et improved the performance of the semi-supervised transfer metric learning algorithm STML [19] by generating the appropriate graph with relative distance constraints. In [21], Mahadevan *et al.* considered the Riemannian geometry of covariance matrices to minimize geometrical and statistical shifts between domains while learning a metric. Amid *et al.* [22], proposed an algorithm that learns a kernel matrix using the log-determinant divergence that is subject to a set of relative distance constraints. Dai *et al.* proposed an alg orithm called EigenTransfer [23] which learns the spectra of a graph that is obtained from the learning task to obtain eigenvectors that are able to accurately capture the structure of the graph. MTLF (Metric Transfer Learning Framework), proposed by Xu *et al.* [24], learns instance weights from the source domain and uses Mahalanobis distance to reduce conditional variance. These are learned in a parallel framework to reduce error propagation.

In recent years, researchers have proposed novel methods for feature selection, optimization and training which greatly improve the accuracy obtained on various benchmark datasets. Al *et al.* in citeal2018feature propose a feature selection method that can determine the optimal feature subset of a dataset for diagnosing coronary artery disease.

Support Vector Machines (SVM) are used for the classification problem while Grey Wolf Optimization (GWO) is used for feature selection. In citeal2018hybrid

These three transfer learning approaches make use of distance formulae that do not capture the manifold of the data very well. This can lead to a misrepresentation of the similarities and dissimilarities of the data, especially the class relationships in the manifold which can then lead to limited knowledge transfer that reduces the accuracy of the transfer learning algorithm. To deal with this problem, our proposed framework, SSMTR, uses relative distance constraints to estimate a relative distance metric and KL-divergence is used to minimize the distribution between the source and target domain data.

III. SEMI-SUPERVISED METRIC TRANSFER LEARNING WITH RELATIVE CONSTRAINTS

A. PROBLEM STATEMENT

Given a labelled source domain with n_s data samples: $\{X^s, Y^s\} = \{(x_1^s, y_1^s), (x_2^s, y_2^s), \ldots, (x_n^s, y_n^s)\}$ where $\{x_i^s \in R^d\}$ is a feature vector. Define $\{X_l^t, Y_l^t\} = \{(x_1^t, y_1^t), (x_2^t, y_2^t), \ldots, (x_l^t, y_l^t)\}$ as labelled data samples and $\{X_n^t\} = \{(x_1^t), (x_2^t), \ldots, (x_n^t)\}$ as unlabelled data samples in the target domain. Since the source and target data samples are distributed in different feature spaces, there exist marginal and conditional distributions between the two domains. Hence, $P_T(x) \neq P_S(x)$.

B. SSMTR FRAMEWORK

1) RELATIVE DISTANCE METRIC

In this paper, we have proposed a unified framework to learn instance weights v for the source domain data and a relative distance metric A_{D_t} for the target domain that captures the relationship between the two domains more accurately. To estimate the instance weights, the entire data distributions of the source and target domains are used.

To learn the Relative Distance Metric, we consider constraints that are applied on groups of three data samples i, j and k where, i, j, k belong to some domain. Unlike Must-Link and Cannot-Link (ML/CL) constraints [27] that are used to represent pair-wise similarities in and between classes, relative constraints are relative distance comparisons between data points and do not hold any information about the clustering structure. Relative Distance Constraints [28] are of two types, 1) k is an outlier in a group of i, j and k data points. This can be defined as a tuple $(i,j \mid k)$ where $\delta(i,j) < \delta(i,k)$ and $\delta(j,i) < \delta(j,k)$ and 2) i, j and k are equidistant from each other and $\delta(i,j) = \delta(i,k) = \delta(j,k)$.

Using these Relative Distance Constraints, we are able to capture the structure of the manifold better. Let A_{D_t} be the positive definite metric for the target domain and i, j, and k be three data points where i and j belong to the same class and k belongs to some other class for inequality constraints while all three points belong to the same class for equality constraints. Then the constraints can be defined as follows:

$$\sigma \left\|x_i - x_j\right\|_{A_{D_t}}^2 \leq \|x_i - x_k\|_{A_{D_t}}^2 \tag{1}$$

where σ is a constant factor. Equation 1 indicates that if there is an inequality constraint then the distance between similar data points, i and j, always is less than that between dissimilar data points, i and k, else the distance between them must be equal.

2) REGULARIZED TERM $\psi(V)$

Since the distributions in both domains are different, we require a co-variate shift adaptation to minimize the distribution accurately. Therefore, we define a regularization term, $\psi(v)$, to influence the co-variate shift as follows,

$$\psi(v) = \|v - v_0\|^2 \tag{2}$$

where $v_0(x_i)$ is the initial weight of data sample, x_i, in the source domain under the Euclidean metric. With the value $v_0(x_i)$, we can determine how much the data sample x_i is similar to source data or target data. If the value of $v_0(x_i)$ is high, it is more similar to source domain compared to target domain.

For calculating the value of $v_0(x_i)$, we adopt the method proposed in the paper [15], where the importance of $v_0(x_i)$ is determined by the following linear model, i.e., $v_0(x_i) = \sum i_{l=1}^{b} \delta_l \varphi_l(x)$, where $\{\omega_l\}_{l=1}^{b}$ are non-negative parameters to be learned from the available data and $\{\varphi_l(x_i)\}_{l=1}^{b}$ are a set of Gaussian kernel functions for all $l = 1, \ldots, b$. Thus, we can estimate the weight of $v_0(x)$ by minimizing the KL-divergence between $P_T(x)$ and $v_0(x)$ $P_S(x)$ as follows:

$$\min_{v_0} KL(P_T(x)||v_0(x)P_S(x)) = \int P_T(x)log\frac{P_T(x)}{v_0(x)P_S(x)}dx \tag{3}$$

According to [15], the problem in Eq 3 can be summarized as follows:

$$\begin{aligned} &\max_{\omega} \sum_{x_i \in D_T} \log \sum_{j=1}^{b} \omega_j \varphi_j(x_i) \\ &\text{s.t.} \sum_{x_i \in D_S} \sum_{j=1}^{b} \omega_j \varphi_j(x_i) = n_s, \quad \text{and } \omega > 0 \end{aligned} \tag{4}$$

The optimization problem in Eq. 4 is convex, hence the optimal solution can be found using gradient descent.

C. SSMTR FOR CLASSIFICATION PROBLEMS

When addressing the classification problems for target domain, we need to learn the target domain metric A_{D_t} from sufficient source data as well as few labelled data of target domain. For this, we obtain the specific optimization problem for classification as follows:

$$\begin{aligned} \min_{A_{D_t}, \hat{v}} \; & \operatorname{tr}(A_{D_t} A_{D_t}^T) + \delta \left\|\hat{v} - \hat{v}_0\right\|^2 \\ & + \beta(\sigma \sum_{i,j} \hat{v}(x_i)\hat{v}(x_j) \left\|x_i - x_j\right\|_{A_{D_t}}^2 \\ & - \sum_{i,k} \hat{v}(x_i)\hat{v}(x_k) \|x_i - x_k\|_{A_{D_t}}^2) \\ \text{s.t.} \; & \sum_{i=1}^{n_s} \hat{v}(x_i) = n_s, \text{ and } \hat{v}(x_i) \geq 0 \end{aligned} \tag{5}$$

where δ and β are the trade-off parameters, σ is a constant factor for constraints, and n_s is the number of samples in source domain.

IV. OPTIMIZATION

For solving the optimization problem in Eq.5, we need to minimize both A_{D_t} and $\hat{v}$ while satisfying the set of constraints. Using the Lagrange multiplier, we can write Eq. 5 as follows:

$$\begin{aligned}\min_{A_{D_t},\hat{v}} J &= \mathrm{tr}(A_{D_t}A_{D_t}^T) + \delta \left\|\hat{v} - \hat{v}_0\right\|^2 + \beta(\sigma \sum_{i,j} \hat{v}(x_i)\hat{v}(x_j) \\ &\quad \times \left\|x_i - x_j\right\|_{A_{D_t}}^2 - \sum_{i,k} \hat{v}(x_i)\hat{v}(x_k) \left\|x_i - x_k\right\|_{A_{D_t}}^2) \\ &\quad + \gamma((\hat{v}^T I - n_s)^2 + \sum_{i=1}^{n_s}(max(0, -\hat{v}(x_i)))^2) \qquad (6)\end{aligned}$$

γ is Lagrange multiplier, n_t is the number of samples in target domain, and I is a vector of size $(n_s + n_t^l) * 1$, where $I_i = 1$ if $i \leq n_s$ else $e_i = 0$.

For learning the values of A_{D_t} and $\hat{v}$, we use an alternative optimization approach.

Firstly, for optimizing the objective function J with respect to $\hat{v}$ while A_{D_t} is fixed: the partial derivative of optimization function J stated in Eq. 6 with respect to $\hat{v}$ is formulated as follows:

$$\begin{aligned}\frac{\partial J}{\partial \hat{v}} &= 2\delta(\hat{v} - \hat{v}_0) + 2\beta(\sigma \sum_{i,j} \hat{v}(x_j) \left\|x_i - x_j\right\|_{A_{D_t}}^2 \\ &\quad - \sum_{i,k} \hat{v}(x_k) \left\|x_i - x_k\right\|_{A_{D_t}}^2) + \gamma[2(\hat{v}^T I - n_s)I + \hat{v}^2\varepsilon] \qquad (7)\end{aligned}$$

where $\varepsilon_i = \mathrm{sign}(\max(0, -\hat{v}(x_i)))$

Secondly, optimizing objective function J with respect to A_{D_t} while $\hat{v}$ is fixed: the partial derivative of optimization function J stated in Eq. 6 with respect to A_{D_t} is formulated as follows:

$$\begin{aligned}\frac{\partial J}{\partial A_{D_t}} &= 2\mathrm{tr}(A_{D_t}) \\ &\quad + 2\beta(\sigma \sum_{i,j} \hat{v}(x_i)\hat{v}(x_j) A_{D_t}(x_i - x_j)(x_i - x_j)^T \\ &\quad - \sum_{i,k} \hat{v}(x_i)\hat{v}(x_k) A_{D_t}(x_i - x_k)(x_i - x_k)^T) \qquad (8)\end{aligned}$$

The values of $\hat{v}$ and A_{D_t} are alternatively updated till their values become less than some threshold values. The systematic algorithm is summarized in Algorithm 1.

Algorithm 1 The Outline of SSMTR Algorithm

```
Input : Target task labelled data $\{X_l^t, Y_l^t\}$, target task
        unlabelled data $\{X_n^t\}$, source task data $\{X^s, Y^s\}$,
        step sizes $\mu_1$ and $\mu_2$, regularized parameters $\delta$
        and $\beta$, a constant factor $\sigma$, maximum number
        of iterations $\tau$, Lagrange multiplier $\gamma$, initial
        value of target task regularized distance metric
        $A_{D_t}$, weight vector $\hat{v}_0$ and threshold $t$
Output: $A_{D_t}$, $\hat{v}$
for $i := 0$ to $\tau$ do
    Find out gradients $\frac{\partial J(A_{D_t}^i, \hat{v}^i)}{\partial A_{D_t}^i}$ and $\frac{\partial J(A_{D_t}^i, \hat{v}^i)}{\partial \hat{v}}$ using
    Eqs.7 and 8.
    Update the value of $\hat{v}^i$ by $\hat{v}^{i+1} = \hat{v}^i - \mu_1 \frac{\partial J(A_{D_t}^i, \hat{v}^i)}{\partial \hat{v}^i}$
    Update the value of $A_{D_t}^i$ by
    $A_{D_t}^{i+1} = A_{D_t}^i - \mu_2 \frac{\partial J(A_{D_t}^i, \hat{v}^{i+1})}{\partial A_{D_t}^i}$
    if $\left|J(A_{D_t}^{i+1}, \hat{v}^{i+1}) - J(A_{D_t}^i, \hat{v}^i)\right| < t$ then
        $A_{D_t}^i = A_{D_t}^{i+1}$
        $\hat{v} = \hat{v}^{i+1}$;
        Break;
    end
end
```

V. EXPERIMENTS

A. BENCHMARK DATASETS

To verify the effectiveness of our proposed algorithm, we have compared SSMTR[1] with 11 other transfer learning algorithms including both unsupervised and semi-supervised transfer learning algorithms like MTLF [24], TJM [8], JDA [7], GFK [10], JGSA [11], TCA and SSTCA [6], MIDA and SMIDA [12] and TML [18] and STML [19] on three well-known datasets, the CMU Multi-PIE Face Database, the Office-Caltech Dataset, and Handwriting Digit Recognition on the MNIST-USPS dataset. Fig.1 shows the sample images of Caltech Office, Handwritten digit (USPS-MNIST),

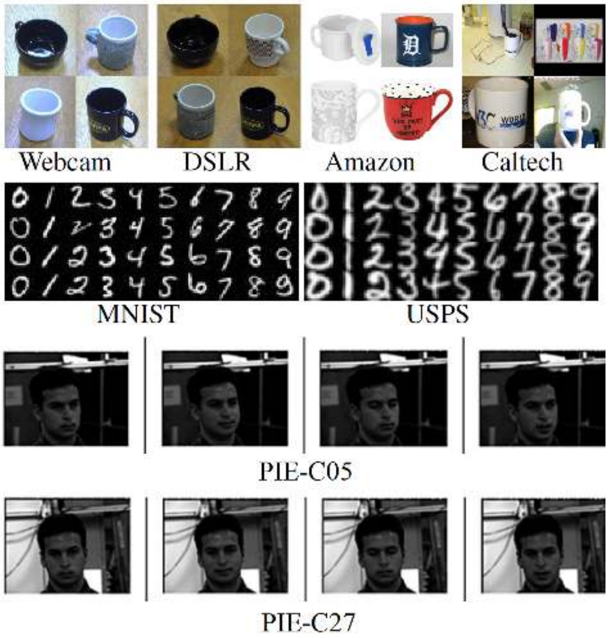
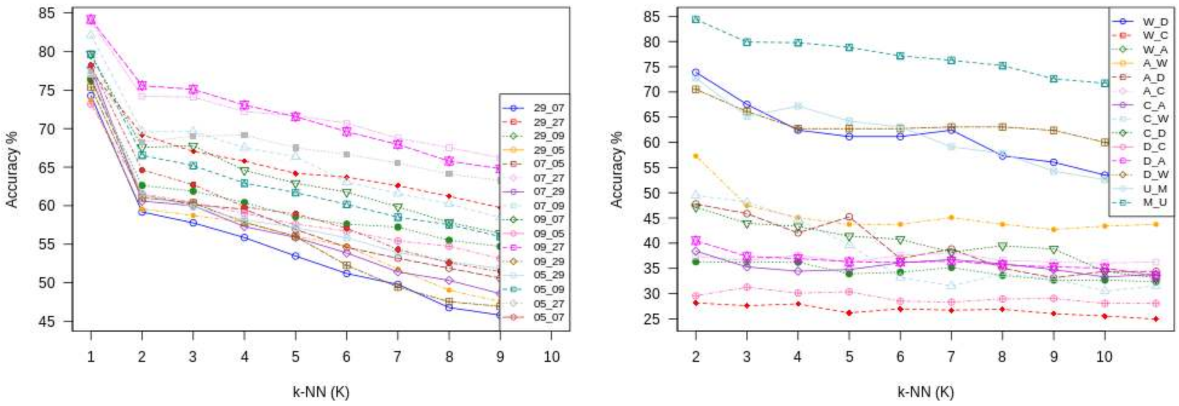


FIGURE 1. Sample images of Caltech Office, Handwritten digit (USPS-MNIST), and PIE Face (PIE-C05, PIE-C09) datasets.

[1] Source codes are available at https://github.com/rakesh1000/SSMTR



(a)

(b)

FIGURE 2. **Influence of *K* on SSMTR performance on the PIE Face, Office-Caltech and USPS-MNIST datasets, respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.**

and PIE Face (PIE-C05, PIE-C09) datasets. To train the K-NN classifier, we have randomly selected x labelled samples from the source domain and y samples per class from the target domain. The entire target domain has been considered for testing. The experiments were conducted for each task and the average results have been reported in Table 1.

The CMU Multi-PIE Face Database [29] consists of more than 600 images of 68 subjects with 13 camera views and 43 different illumination conditions. Additionally, the subjects display a range of facial expressions and the database has high-resolution frontal images as well. The images were re-sized to 32 × 32 pixels before the experiment and their vectorized, gray-scale images were used as feature vectors. For this experiment, we have generated 16 cross-domain tasks from the dataset where each task pair, e.g., 29_07 represents Pose PIE29 used as the source domain and PIE07 used as the target domain. We have selected 8 labelled instances from the source domain and 3 labelled instances are randomly selected from the target domain for training the model.

The Office-Caltech Dataset found in [10] contains images from the Caltech-256 dataset, Amazon (images downloaded from web-retailers), DSLR (high-resolution images) and Webcam (low-resolution images). We have selected 10 common images from each dataset: calculator, keyboard, mouse, mug, projector, backpack, headphones, monitor, bike and laptop, with the SURF feature dataset. For this experiment, we have generated 12 cross-domain tasks from the dataset where each task pair is composed of two domain datasets, source domain (S) and the target domain (T), e.g., W_D. Here too, we have selected 8 labelled instances per category and 3 labelled instances from the target domain for training the model.

MNIST and USPS [30], [31] are two popular datasets in data mining and pattern recognition applications. MNIST dataset was taken from the mixed American Census Bureau employees and an American high school. It has 60,000 training images and 10,000 testing images where each image size is 28 × 28. USPS dataset was collected by scanning envelops from the US Postal Service and has a total of 9298 labelled images where the size of each image is 16x16. However, for our experiment purpose we randomly choose 1800 samples from USPS dataset while 2000 samples from MNIST dataset. Here, both datasets have different marginal distributions, so there are two domain adaptation cases since we can use one domain as the source domain and the other one as the target domain for one task and vice-versa for the second task. Hence, in Table 1 and all the graphs, we have used M_U for the task MNIST-USPS and U_M for the task USPS-MNIST.

B. PARAMETER SENSITIVITY

1) EXPERIMENTAL ANALYSIS ON PARAMETER *K*

The performance of the KNN classifier depends on the parameter K. We vary the value of K from 1-10 while keeping the value of the other parameters constant for all three datasets, to find the value at which our proposed algorithm works most efficiently. For the PIE Face dataset, the values of the other parameters are kept constant at $C = 5, \sigma = 3, \delta = 1$, $\beta = 10^{-2}$, $\gamma = 1$, $d = 100$, $\mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. For the Office-Caltech and USPS-MNIST datasets, the values of the other parameters are kept constant at $C = 2$, $\delta = 1$, $\beta = 10^{-3}$, $\gamma = 1$, $d = 30$, $\mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. For Office-Caltech dataset, we consider $\sigma = 2$ while for USPS-MNIST dataset, we consider $\sigma = 3.5$. It is seen in the graphs in Fig. 2 that the performance of the algorithm keeps on reducing as we vary the value of K. At $K = 1$, the proposed algorithm shows maximum accuracy on all three datasets.

2) EXPERIMENTAL ANALYSIS ON CONSTRAINTS PARAMETER *C*

The number of constraints is fixed by the parameter C. We vary the C value from 2-10 while keeping the value of the other parameters constant. For the PIE Face dataset, the values of the other parameters are kept constant at $K = 1$, $\sigma = 3$, $\delta = 1$, $\beta = 10^{-2}$, $\gamma = 1$, $d = 100$, $\mu_1 = 10^{-4}$

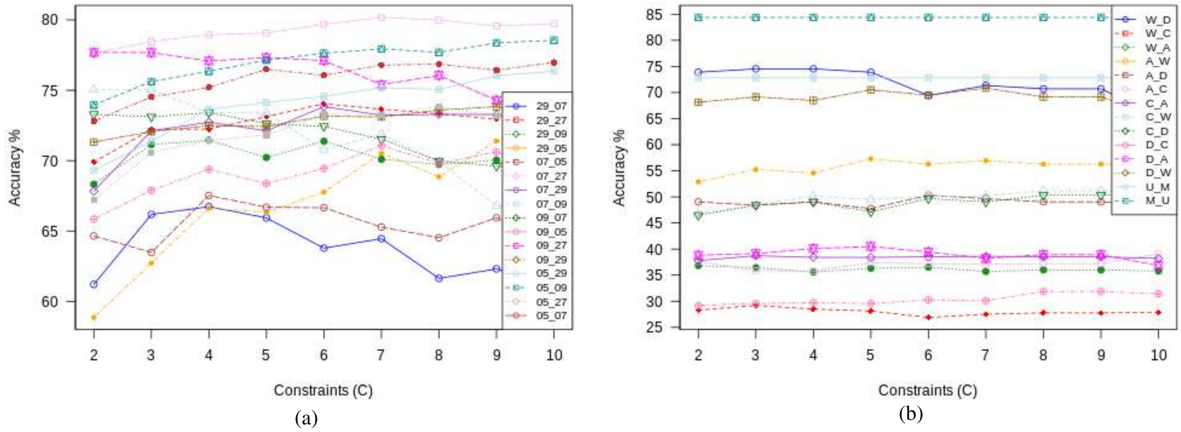


FIGURE 3. Influence of number of constraints, *C* on SSMTR performance on the PIE Face, Office-Caltech and USPS-MNIST datasets, respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.

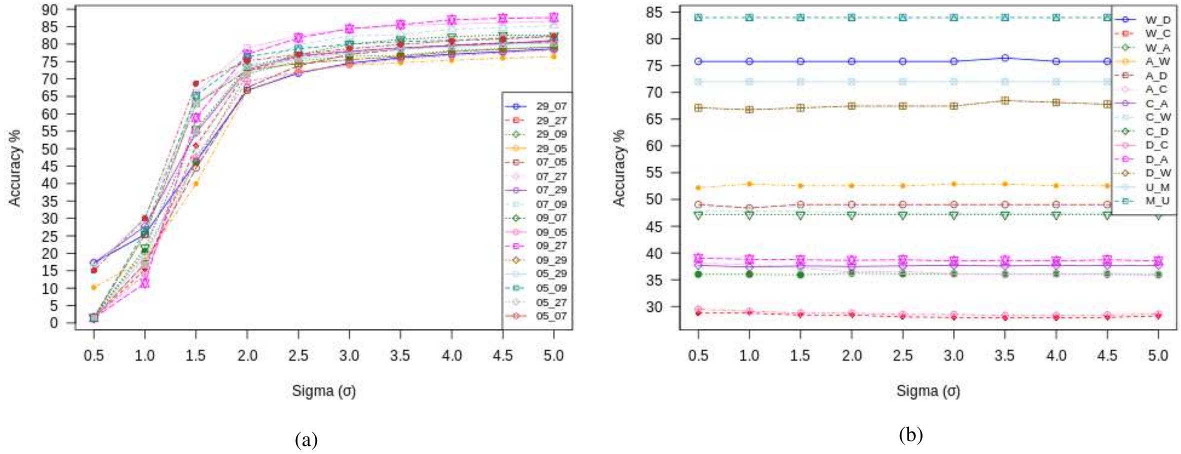


FIGURE 4. Influence of dimension parameter, sigma (σ), on SSMTR performance on the PIE Face, Office-Caltech and USPS-MNIST datasets, respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.

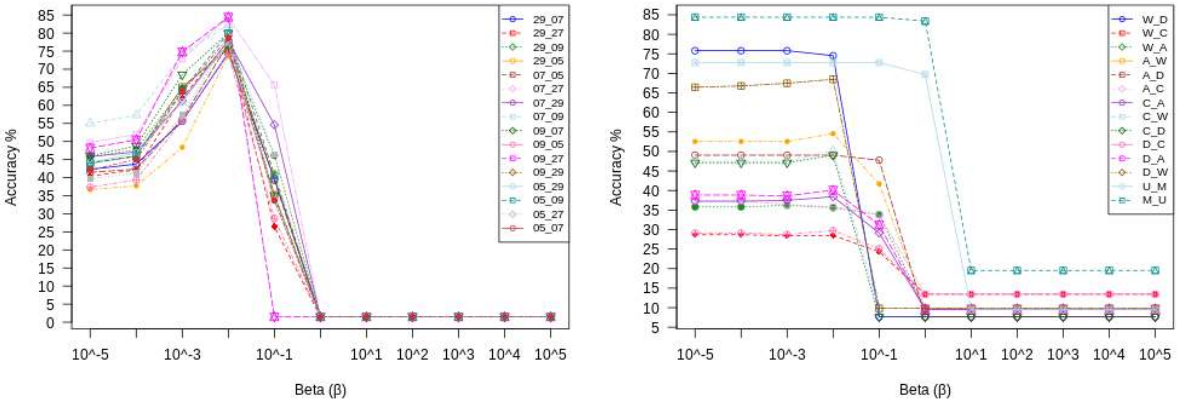
and $\mu_2 = 10^{-5}$. For the Office-Caltech and the USPS-MNIST datasets, the values of the other parameters are kept constant at $K = 1$, $\delta = 1$, $\beta = 10^{-3}$, $\gamma = 1$, $d = 30$, $\mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. For Office-Caltech dataset, we consider $\sigma = 2$ while for USPS-MNIST dataset, we consider $\sigma = 3.5$. In doing so we obtain Fig 3. From Fig. 3 (a), we can see that the accuracy of the algorithm peaks at $C = 5$ for the PIE Face dataset but for the Office-Caltech and USPS-MNIST datasets, from Fig. 3 (b), we can see that for all the tasks in the dataset, the algorithm shows maximum accuracy at $C = 2$.

3) EXPERIMENTAL ANALYSIS ON CONSTANT FACTOR σ FOR CONSTRAINTS

Similar to previous parameter analysis, the values of the other parameters are kept constant while the value of parameter, σ, was varied in the range: 0.5-5.0 and Fig. 4 was obtained. For the PIE Face dataset, the values of the other parameters were kept as: $K = 1, = 5$, $\delta = 1$, $\beta = 10^{-2}$, $\gamma = 1$, $d = 100$, $\mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. For the Office-Caltech and USPS-MNIST datasets, the values of the other parameters are kept constant as: $K = 1, = 2$, $\delta = 1$, $\beta = 10^{-3}$, $\gamma = 1$, $d = 30$, $\mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. From Fig. 4 (a), we can see that the accuracy of the proposed algorithm increases as the value of σ increases and all the tasks show maximum accuracy at $\sigma = 3$. From Fig. 4 (b), we can see that for Office-Caltech dataset, the accuracy does not change a lot when σ is varied and the tasks show maximum accuracy at $\sigma = 2$. For USPS-MNIST dataset, we can see that the tasks show maximum accuracy at $\sigma = 3.5$.

4) EXPERIMENTAL ANALYSIS ON TRADE-OFF PARAMETERS: δ, β AND A LAGRANGE MULTIPLIER, γ

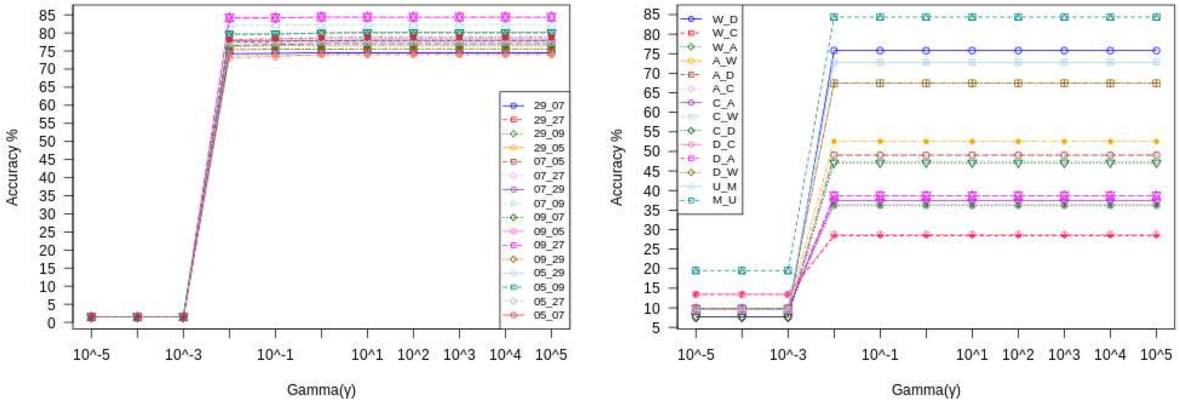
We analyze the performance of our proposed algorithm for the three trade-off parameters: δ, β and Lagrange multiplier, γ as seen in the objective function. To analyze the parameter behavior, we vary the value of these trade-off parameters as β from 10^{-5} to 10^{5} and γ from 10^{-5} to 10^{5}. For the PIE Face dataset, the values of the other parameters were kept as: $K = 1, = 5$, $\sigma = 3$, $d = 100$, $\mu_1 = 10^{-4}$



(a)

(b)

FIGURE 5. Influence of Lagrange multiplier, β, on SSMTR performance on the PIE Face, Office-Caltech and USPS-MNIST datasets, respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.



(a)

(b)

FIGURE 6. Influence of parameter Gamma (γ) to SSMTR performance on PIE Face and Office-Caltech datasets, respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.

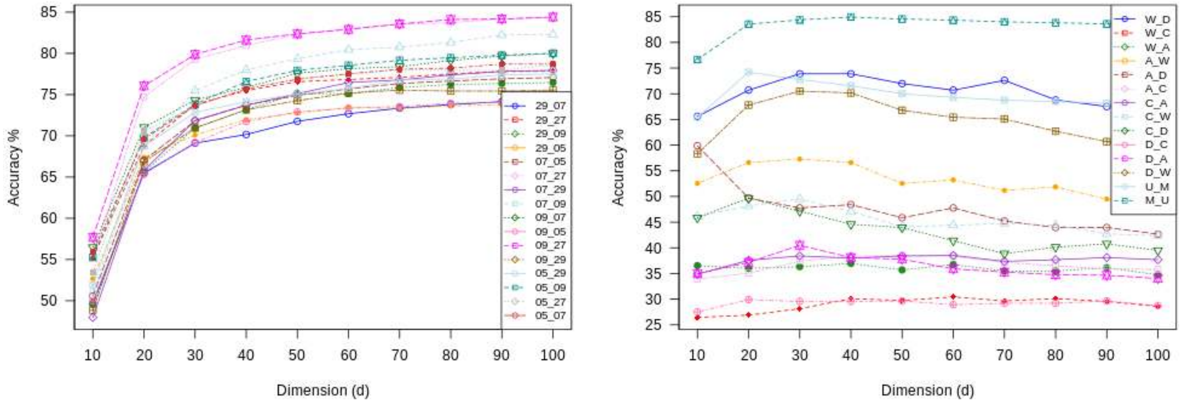
TABLE 2. Optimized parameter values after conducting the parameter sensitivity test on all three datasets.

Dataset	K	C	σ	δ	β	γ	d	μ_1	μ_2
PIE Face	1	5	3	1	10^{-2}	1	100	10^{-4}	10^{-5}
Office-Caltech	1	2	2	1	10^{-3}	1	30	10^{-4}	10^{-5}
USPS-MNIST	1	2	3.5	1	10^{-3}	1	30	10^{-4}	10^{-5}

and $\mu_2 = 10^{-5}$. For the Office-Caltech and USPS-MNIST datasets, the values of the other parameters are kept constant as: $K = 1, = 2, d = 30, \mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. For Office-Caltech we consider $\sigma = 2$ while for USPS-MNIST, we consider $\sigma = 3.5$. As seen in 6, at value $\gamma = 1$, the proposed algorithm shows maximum accuracy for all three datasets. From Fig 5, we see that for PIE Face dataset, $\beta = 10^{-2}$ and for Office-Caltech and USPS-MNIST datasets, $\beta = 10^{-3}$ provides us the maximum accuracy. From Table 2, it is revealed that on performing the same analysis on parameter δ, maximum accuracy is obtained at $\delta = 1$ for all three datasets.

5) EXPERIMENTAL ANALYSIS ON DIMENSIONALITY d

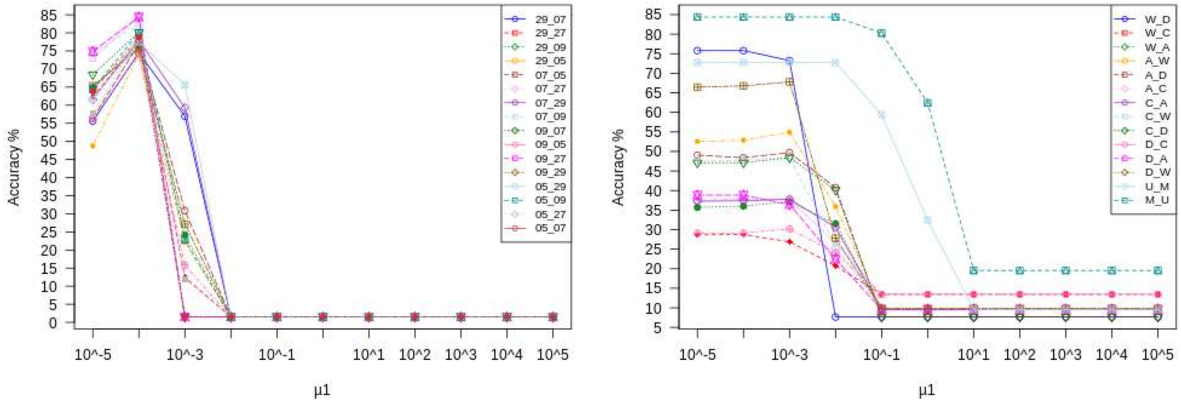
The dimensions of the PIE Face and Office-Caltech datasets are 1024 and 800, respectively, and the processing time required for performing the experiments is very high because of the high dimensionality. This is not desirable and we use PCA to project the data to lower dimensions to reduce the processing time. As seen in previous subsections, we vary the value of parameter, d, from 10-100 for both datasets while keeping the values of the other parameters constant and plot the Fig. 9. For the PIE Face dataset, the values of the other parameters were kept as: $K = 1, = 5, \sigma = 3, \delta = 1, \beta = 10^{-2}$, $\gamma = 1, \mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. For the Office-Caltech and USPS-MNIST datasets, the values of the other parameters are kept constant as: $K = 1, = 2, \sigma = 3$ and 3.5 (for Office-Caltech and USPS-MNIST datasets, respectively), $\delta = 1$, $\beta = 10^{-3}, \gamma = 1, \mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$. We see that the plot for PIE Face dataset keeps on increasing as d is varied and achieves maximum accuracy at $d = 100$. For both the Office-Caltech and USPS-MNIST datasets, the algorithm achieves maximum accuracy at $d = 30$.



(a)

(b)

FIGURE 7. Influence of dimension parameter, d, on SSMTR performance on the PIE Face, Office-Caltech and USPS-MNIST datasets respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.



(a)

(b)

FIGURE 8. Influence of Gradient parameter, μ_1, on SSMTR performance on the PIE Face, Office-Caltech and USPS-MNIST datasets, respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.

6) EXPERIMENTAL ANALYSIS ON PARAMETERS μ_1 AND μ_2

On performing similar analysis on parameters, μ_1 and μ_2, we keep the values of the other parameters constant and vary the values of μ_1 and μ_2. For the PIE Face dataset, the values of the other parameters were kept as: $K = 1, = 5, \sigma = 3, \delta = 1, \beta = 10^{-2}, \gamma = 1$ and $d = 100$. For the Office-Caltech and USPS-MNIST datasets, the values of the other parameters are kept constant as: $K = 1, = 2, \sigma = 3$ and 3.5 (for Office-Caltech and USPS-MNIST datasets, respectively), $\delta = 1$, $\beta = 10^{-3}$, $\gamma = 1$ and $d = 30$. We find that the proposed algorithm clearly attains maximum accuracy at $\mu_1 = 10^{-4}$ and $\mu_2 = 10^{-5}$ for all three datasets, as reported in Table 2.

7) DISCUSSION ON OPTIMAL PARAMETER VALUES

After conducting the above experiments on all the tuning parameters, we have reported the suggested ranges or values for all the parameters of our proposed SSMTR learning framework on different datasets in Table 2. This can be used as prior information for cross-validation to optimize parameters on specific datasets.

C. RESULTS AND DISCUSSION

Our proposed SSMTR framework utilizes the parameter values obtained after conducting the parameter sensitivity test, shown in Table 2. We conducted the experiments on PIE Face, Office-Caltech and USPS-MNIST datasets. The results have been reported in Table 1, and the findings are discussed below.

For the PIE Face dataset, it is seen that most methods show a very poor accuracy on the given tasks. The state of art algorithm, MTLF, has the best accuracy compared to all the other algorithms that we have compared SSMTR with. MTLF has a mean accuracy of 43.74% while our proposed framework has a mean accuracy of 77.94%. Thus, our proposed algorithm outperforms all other frameworks and attains the best accuracy on the given tasks.

For the Office-Caltech dataset, it is seen that for task W_D, SSTCA attains the same accuracy as SSMTR while for tasks A_D, SSTCA attains a greater accuracy than our proposed framework. But it is seen that for all other tasks in the dataset (W_A, A_W, A_C, C_A, C_W, C_D, D_C, D_A, D_W), SSMTR provides a greater accuracy than all other 11 algorithms. Thus we can say that on a whole, the SSMTR framework performs better than the other algorithms.

On the USPS-MNIST dataset, SSMTR performs better than all the other 11 algorithms on the U_M and M_U tasks. On U_M, SSMTR shows 74.8% accuracy while the greatest accuracy shown by the other algorithms is 67.20% by MTLF. On M_U, MTLF and SSMTR have similar accuracies of 83.83% and 84.33% respectively. Hence, we can conclude that SSMTR shows better performance on the USPS-MNIST dataset when compared to previous algorithms.

D. COMPLEXITY ANALYSIS OF OUR PROPOSED FRAMEWORK

Given a labelled source domain with n_s data samples, where each sample is a d dimensional vector, and a labelled target domain with n_t^l data samples, let us consider τ to be the maximum number of iterations, C to be the maximum number of constraints and g to be some constant parameter.

1) Note that in Eq. 5, the loss function value is determined by considering the pairwise relative distance constraints set C that contains both equality and inequality constraints. However, considering a large number of such pairwise relative distance constraints may increase the computational complexity of our proposed approach. Thus, the complexity of the loss function of the proposed approach is $O(Cd^2)$.

2) There are two gradients $\frac{\partial J(A^i_{D_t},\hat{v}^i)}{\partial A^i_{D_t}}$ and $\frac{\partial J(A^i_{D_t},\hat{v}^i)}{\partial \hat{v}}$ that have been computed using Eqs.7 and 8, respectively, where the first gradient is the partial derivative of function J with respect to A_{D_t} while other one is with respect to $\hat{v}$. The gradient $\frac{\partial J(A^i_{D_t},\hat{v}^i)}{\partial A^i_{D_t}}$ takes $O((n_s + n_t^l)Cd^2)$ time and other gradient takes $O(Cd^2)$ time. Thus, the total time complexity for computing both the gradients in step 2 of the Algorithm 1 is $(O((n_s + n_t^l)Cd^2) + O(Cd^2))$.

3) Moreover, from Lines 3-7 in Algorithm 1, the computational cost taken by other operations, is g.

4) Our proposed framework is iterated for τ number of iterations.

So, total run-time complexity of our proposed approach is, $\mathcal{O}(\tau((n_s + n_t^l)Cd^2 + Cd^2 + g))$

$$\begin{aligned} &\implies \mathcal{O}(\tau((n_s + n_t^l)Cd^2 + Cd^2 + g)) \\ &\implies \mathcal{O}(\tau((n_s + n_t^l + 1)Cd^2 + g)) \\ &\implies \mathcal{O}(\tau((n_s + n_t^l)Cd^2)) \end{aligned}$$

Thus, the total complexity of our proposed system is $\mathcal{O}(\tau((n_s + n_t^l)Cd^2))$

E. COMPARISON OF TIME COMPLEXITY WITH OTHER ALGORITHMS

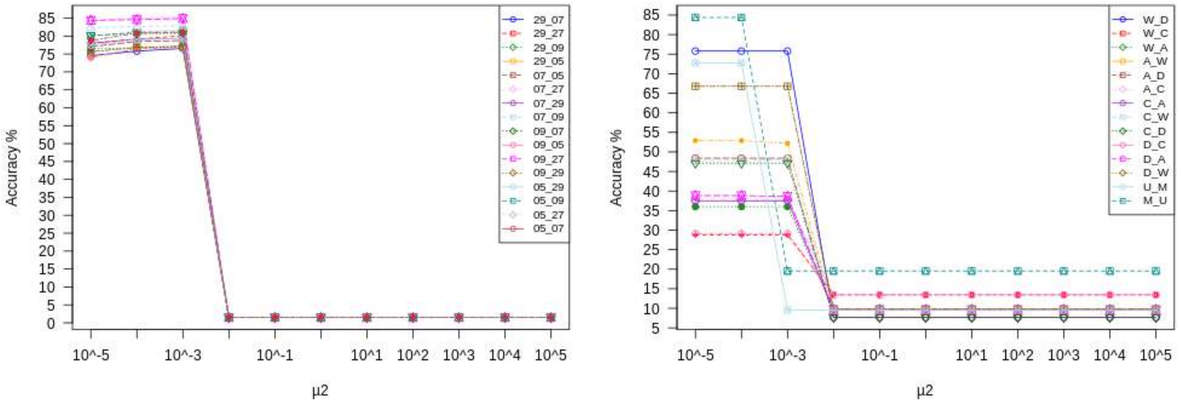
We have compared the time complexity of our proposed algorithm with the time complexities of the other algorithms and reported the comparative study in Table 3. The time complexities which are reported in Table 3, are directly taken from the respective reference papers. The time complexity of SSMTR as discussed in Section V-D, is dependent on various factors such as number of labelled data samples in both source and target domain and constraint factor C. From Table 3, we can see that our proposed method has a lesser time complexity compared to the other if we keep all the parameters constant except n_s, n_t, where n_s is the number of labelled samples in source domain and n_t is the number of labelled samples in the target domain, as the other algorithms have their time complexity $\mathcal{O}(d^2)$. However, keeping other parameters constant, proposed approach may not perform well. On comparing our algorithm with MTLF, TJM, JDA, TCA and SSTCA, it was found that our proposed approach has a little high computational time when compared to those by others. However, the performance of our approach is much better compared to all other existing approaches.

TABLE 3. **Complexities of various existing comparative transfer learning algorithms with our proposed SSMTR algorithm. Here, d is the number of features in each data point; n is the total samples available in the dataset; τ is the maximum number of iterations; C is the number of constraints; n_s and n_t^l are the number of samples in source domain and target domain, respectively; m is the number of shared features.**

Algorithm	Time complexity
SSMTR	$\mathcal{O}(\tau((n_s + n_t^l)Cd^2))$
MTLF	$\mathcal{O}(Cd^2)$
TJM	$\mathcal{O}(\tau dn^2 + mn^2)$
JDA	$\mathcal{O}(\tau dm^2 + \tau Cn^2 + \tau mn)$
GFK	not mentioned
JGSA	not mentioned
TCA and SSTCA	$\mathcal{O}(d(n_s + n_t^l)^2)$
MIDA and SMIDA	not mentioned
TML and SMTL	not mentioned

F. STATISTICAL SIGNIFICANCE TEST

In order to validate our obtained results statistically, we have conducted Welch's t-test [32] with the significance level of 0.05 to show whether the obtained results are statistically significant or they happened by chance. While conducting the experiments to determine p-values, we consider two groups; the first group consists of a list of values of accuracies attained by our proposed approach; while the second group consists of a list of values of accuracies attained by other approaches. We have considered two hypotheses, an alternative hypothesis and a null hypothesis. The first hypothesis assumes that there is a significant change between the median values of the two groups while the second hypothesis assumes that there is no significant difference between the median values of the two groups. We have reported the p-values obtained in Table 4 for all the tasks of Office-Caltech and USPS-MNIST datasets. However, the p-values for all the tasks of PIE Face dataset are less than 0.00001, which are significant. From Table 4, we can conclude that for the tasks, W_D, A_W and C_A, improvements obtained by STCA algorithm are not significant while



(a)

(b)

FIGURE 9. Influence of Gradient parameter, μ_2, on SSMTR performance on the PIE Face, Office-Caltech and USPS-MNIST datasets, respectively. (a) PIE Face dataset. (b) Office-Caltech and USPS-MNIST datasets.

TABLE 4. p-values obtained by t-test conducted on the accuracy values obtained by different compared algorithms for the Office-Caltech and USPS-MNIST datasets.

Office-Caltech Dataset											
Tasks	**MTLF**	**TJM**	**JDA**	**GFK**	**JGSA**	**TCA**	**STCA**	**SMIDA**	**MIDA**	**TML**	**STML**
W-D	0.00004	0.002423	0.000124	0.000011	0.000015	0.0001	**0.242278**	0.000253	0.00001	0.00001	0.00001
W-C	0.012701	0.006031	0.014262	0.019725	0.00001	0.000076	0.001396	0.033427	0.003093	0.00001	0.00001
W-A	0.002043	0.000014	0.000104	0.00002	0.00001	0.00001	0.088217	0.003069	0.00001	0.00001	0.00001
A-W	0.000153	0.00001	0.00001	0.00001	0.00001	0.00001	**0.353475**	0.00001	0.00001	0.00001	0.00001
A-D	0.000017	0.000012	0.000014	0.00001	0.00001	0.00001	0.00038	0.00037	0.00001	0.000012	0.00001
A-C	0.001594	0.002236	0.000729	0.000162	0.00001	0.00035	0.045224	0.046224	0.00001	0.00001	0.00001
C-A	**0.101193**	0.002145	0.000851	0.000123	0.00001	0.00011	**0.353475**	0.101213	0.00036	0.00001	0.00001
C-W	0.002145	0.00001	0.00011	0.00001	0.00011	0.00011	0.004964	0.000091	0.00001	0.00001	0.00001
C-D	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.000284	0.00001	0.00001	0.00001	0.00001
D-C	0.005217	0.000311	0.000312	0.000311	0.00001	0.00004	0.004354	0.004354	0.00001	0.00001	0.00001
D-A	0.000108	0.000101	0.000101	0.00001	0.00001	0.00001	0.001224	0.00009	0.00001	0.00001	0.00001
D-W	0.00001	0.019003	0.018003	0.00001	0.00001	0.00001	0.019211	0.093429	0.00001	0.00001	0.00001
Handwritten Digit Recognition (USPS, MNIST)											
U-M	0.000038.	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001
M-U	**0.301823**	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001

for task C_A, the improvement obtained by MTLF algorithm is not significant for the Office-Caltech dataset. Similarly, the improvement obtained by MTLF algorithm for the task M_U is not significant.

VI. CONCLUSION

In this paper, we have presented a new method for Semi-Supervised Metric Transfer Learning, SSMTR. By using the Distance Metric Learning with Relative Distance comparison constraints instead of a Mahalanobis distance metric that uses Must-Link (ML) and Cannot-Link (CL) constraints, we are able to express structures in greater detail. We have compared our proposed algorithm with 11 other transfer learning algorithms, MTLF, TJM, JDA, GFK, JGSA, TCA, SSTCA, MIDA, SMIDA, TML, and STML, on three well-known real-world datasets, PIE Face, Office-Caltech dataset and MNIST-USPS dataset. The results suggest that our proposed framework achieves greater accuracy when compared to other algorithms.

In many real-world applications, related tasks with very few labeled data and abandoned amount of unlabeled data are available. Therefore, in the future, to enhance the generalization performance of all the related tasks we will extend our model to multi-task metric learning while developing the predictive model.

REFERENCES

[1] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2003, p. II.

[2] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. Pennsylvania, PA, USA: IGI Global, 2010, pp. 242–264.

[3] 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.

[4] X. Zhu, "Semi-supervised learning literature survey," *Comput. Sci., Univ. Wisconsin-Madison*, vol. 2, no. 3, p. 4, 2006.

[5] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn.* New York, NY, USA: ACM, 2007, pp. 759–766.

[6] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[7] 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.

[8] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1410–1417.

[9] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.

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

[11] J. Zhang, W. Li, and P. Ogunbona. (2017). "Joint geometrical and statistical alignment for visual domain adaptation." [Online]. Available: https://arxiv.org/abs/1705.05498

[12] K. Yan, L. Kou, and D. Zhang, "Learning domain-invariant subspace using domain features and independence maximization," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 288–299, Jan. 2018.

[13] S. Bickel and M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, vol. 10, pp. 2137–2155, Sep. 2009.

[14] C. Wan, R. Pan, and J. Li, "Bi-weighting domain adaptation for cross-language text classification," in *Proc. IJCAI Proc.-Int. Joint Conf. Artif. Intell.*, 2011, vol. 22, no. 1, p. 1535.

[15] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1433–1440.

[16] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8156–8164.

[17] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua, ''Robust distance metric learning with auxiliary knowledge,'' in *Proc. IJCAI*. San Francisco, CA, USA, 2009, pp. 1327–1332.

[18] Y. Zhang and D.-Y. Yeung, ''Transfer metric learning by learning task relationships,'' in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: ACM, 2010, pp. 1199–1208.

[19] Y. Zhang and D.-Y. Yeung, ''Transfer metric learning with semi-supervised extension,'' in *Proc. ACM Trans. Intell. Syst. Technol. (TIST)*, 2012, vol. 3, no. 3, p. 54.

[20] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, ''Deep metric learning via lifted structured feature embedding,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4004–4012.

[21] S. Mahadevan, B. Mishra, and S. Ghosh. (2018). ''A unified framework for domain adaptation using metric learning on manifolds.'' [Online]. Available: https://arxiv.org/abs/1804.10834

[22] E. Amid, A. Gionis, and A. Ukkonen, ''A kernel-learning approach to semi-supervised clustering with relative distance comparisons,'' in *Machine Learning and Knowledge Discovery in Databases*, A. Appice *et al.*, Eds. Cham, Switzerland: Springer, 2015, pp. 219–234.

[23] W. Dai, O. Jin, G.-R. Xue, Q. Yang, and Y. Yu, ''Eigentransfer: A unified framework for transfer learning,'' in *Proc. 26th Annu. Int. Conf. Mach. Learn.* New York, NY, USA: ACM, 2009, pp. 193–200.

[24] Y. Xu *et al.*, ''A unified framework for metric transfer learning,'' *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1158–1171, Jun. 2017.

[25] J. Zhang, W. Li, and P. Ogunbona. (2017). ''Transfer learning for cross-dataset recognition: A survey.'' [Online]. Available: https://arxiv.org/abs/1705.04396

[26] R. K. Sanodiya, S. Saha, J. Mathew, and P. Bangwal, ''Semi-supervised transfer metric learning with relative constraints,'' in *Proc. Int. Conf. Neural Inf. Process.* Siem Reap, Cambodia: Springer, 2018, pp. 230–241.

[27] Z. Lu and T. K. Leen, ''Semi-supervised clustering with pairwise constraints: A discriminative approach,'' *J. Mach. Learn. Res.*, vol. 2, pp. 299–306, 2007.

[28] Y. Pei, X. Z. Fern, R. Rosales, and T. V. Tjahja. (2014). ''Discriminative clustering with relative constraints.'' [Online]. Available: https://arxiv.org/abs/1501.00037

[29] T. Sim, S. Baker, and M. Bsat, ''The CMU pose, illumination, and expression (PIE) database,'' in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2002, pp. 53–58.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[31] J. J. Hull, ''A database for handwritten text recognition research,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[32] E. A. Gehan, ''A generalized wilcoxon test for comparing arbitrarily singly-censored samples,'' *Biometrika*, vol. 52, nos. 1–2, pp. 203–224, 1965.

Authors' photographs and biographies not available at the time of publication.