Supervised Transfer Sparse Coding

Maruan Al-Shedivat[1], **Jim Jing-Yan Wang**[2], **Majed Alzahrani**[1], **Jianhua Z. Huang**[3] and **Xin Gao**[1,*]
[1]Computer, Electrical and Mathematical Sciences and Engineering Division,
King Abdullah University of Science and Technology (KAUST), Thuwal, Jeddah 23955, Saudi Arabia
[2]University at Buffalo, The State University of New York, Buffalo, NY 14203, United States
[3]Department of Statistics, Texas A&M University, College Station, TX 77843, United States
{maruan.shedivat, majed.alzahrani, xin.gao}@kaust.edu.sa,
jimjywang@gmail.com, jianhua@stat.tamu.edu

Abstract

A combination of the sparse coding and transfer learning techniques was shown to be accurate and robust in classification tasks where training and testing objects have a shared feature space but are sampled from different underlying distributions, i.e., belong to different domains. The key assumption in such case is that in spite of the domain disparity, samples from different domains share some common hidden factors. Previous methods often assumed that all the objects in the target domain are unlabeled, and thus the training set solely comprised objects from the source domain. However, in real world applications, the target domain often has some labeled objects, or one can always manually label a small number of them. In this paper, we explore such possibility and show how a small number of labeled data in the target domain can significantly leverage classification accuracy of the state-of-the-art transfer sparse coding methods. We further propose a unified framework named supervised transfer sparse coding (STSC) which simultaneously optimizes sparse representation, domain transfer and classification. Experimental results on three applications demonstrate that a little manual labeling and then learning the model in a supervised fashion can significantly improve classification accuracy.

1 Introduction

The classification theory assumes that the training objects and the testing objects are sampled from a single shared distribution. Moreover, it assumes that both marginal and conditional distributions should be identical for training and testing sets. These assumptions are necessary in order to ensure the generalization of a statistically derived classification model. On the contrary, for the real-world data these assumptions may not hold: Training and testing data might come from entirely different domains, usually called the source and the target, respectively. This results in a lack of generalization power of the model trained on the source domain objects only or on a set derived from both domains. According to the classification theory, for every new distribution new data should be acquired and labeled and a new model should be learned. In most of the applications, this process tends to be expensive and time consuming.

In order to address this problem, a variety of methods were proposed (Pan and Yang 2010), where are referred to as transfer learning or domain knowledge adaptation. The main assumption behind these methods is that data have some common latent factors shared across the domains. Given this assumption, one can use these factors to transfer information from the source domain and to leverage the model accuracy on the target domain. In recent years, transfer learning was shown to be effective and efficient in image (Zhu et al. 2011; Wang et al. 2011; Long et al. 2013a) and text (Zhuang et al. 2012) classification tasks, object recognition (Gopalan, Li, and Chellappa 2011), sentiment analysis (Pan et al. 2010a), and collaborative filtering (Pan et al. 2010b).

An intuitively appealing general approach for transfer learning was proposed by Pan, Kwok, and Yang (2008). The idea is to find a latent space where the marginal distributions of the data between different domains are close to each other. A classification model trained on the source domain mapped to such a space can be general enough to be able to make predictions for the target domain data mapped to the same space. Subsequently, different methods based on this idea were proposed, such as transfer component analysis (Pan et al. 2011), domain adaptation for pattern recognition (Gopalan, Li, and Chellappa 2011), metric learning (Geng, Tao, and Xu 2011), and more recent joint distribution adaptation (Long et al. 2013b).

One of the effective techniques to find easily interpretable representations that can capture high-level features from data is sparse coding (Huang and Aviyente 2006; Yang et al. 2009). Instead of using linear or non-linear parametric mapping functions, sparse coding seeks to find a dictionary built of objects from the original space, the sparse combinations of which approximate the data. To improve the quality of the representations found via sparse coding, different modifications to the sparsity constraint were proposed (Liu et al. 2010; Gao et al. 2010; Wang et al. 2010; 2013; Wang, Bensmail, and Gao 2014). Moreover, sparse coding has also been adopted for transfer learning scenarios where labeled and unlabeled images are sampled from different distributions. In order to be able to learn a unified dictionary and a coherent representation for data samples from both domains, Long et al. (2013a) proposed to

*All correspondence should be addressed to Xin Gao.

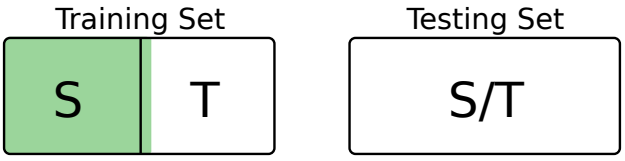


Figure 1: The training and testing set layouts studied in this paper: The training set has samples from both domains, where the source objects are labeled and the target objects are almost unlabeled; the testing set has a mix of source and target objects with unknown labels and domains.

enhance the sparse coding optimization function with an additional regularization constraint called minimum mean discrepancy (MMD) (Gretton et al. 2006). They named their method transfer sparse coding (TSC) and showed that this enhancement along with Graph-Laplacian smoothing (Zheng et al. 2011) produced the state-of-the-art results for semi-supervised transfer learning in image classification.

Transfer sparse coding as well as the other aforementioned transfer learning methods assumed that none of the target domain objects is labeled. Moreover, their training set consisted of samples from the source domain only, and the testing set comprised samples only from the target. This reproduces a real life situation where it is often very expensive to obtain a sufficient amount of labeled data. However, in real world it seems more plausible to have a small number of labeled objects in the target domain or even manually label some if it leads to a significantly more accurate and robust model. Then, the gain in the performance improvement pays back the labeling expenses.

In this paper, we relax all the constraints and study the case where both training and testing sets can have objects from both domains. We assume that, in the training set, objects from the source domain are entirely labeled and those from the target are almost unlabeled. The testing objects are not labeled and also belong to an unknown domain, i.e., either to the source or to the target (Figure 1). We pose the problem as a supervised cross-domain learning task with a soft constraint that only a small number of target domain samples in the training set is labeled.

The proposed setting is natural in various applications that inherently deal with multi-domain mixed datasets. One of the examples is classification of images in social networks and media. In this case, one might aim to classify photographs of people on different backgrounds. Moreover, backgrounds are known for the training data, but unknown for the testing. Another example is bilingual speech and text recognition tasks, important for bilingual countries or international congresses, which again requires the algorithm perform well on mixed-domain data.

It was shown that a sparse coding done in a supervised fashion usually learns highly discriminative dictionaries which facilitates subsequent classification and recognition tasks (Mairal et al. 2009; Jiang, Lin, and Davis 2011). In this paper, we first show that a small number of labeled objects from the target domain can significantly leverage the classification accuracy of the state-of-the-art sparse coding

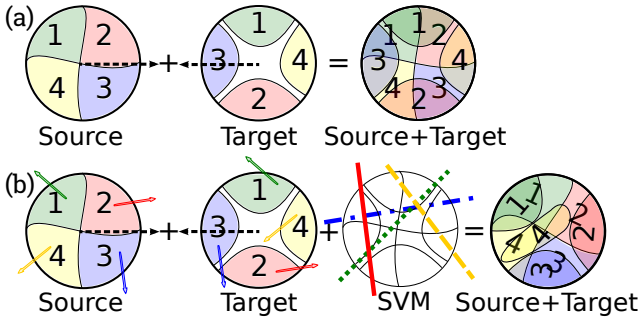


Figure 2: (a) Illustration of TSC. Both the source and target domains have the same four classes. Dashed arrows show the direction of domain merging by TSC. The merged domain is difficult to classify. (b) Illustration of STSC. Arrows with the same colors as SVM decision boundaries regularize the way the domains are merged. The resulting merged domain is much easier to be classified.

methods. We further introduce a unified framework, supervised transfer sparse coding (STSC), that utilizes label information in the target domain in order to build a better discriminative representation of objects from both domains. Our method combines transfer learning and sparse coding with a supervised support vector machine (SVM) term and benefits from simultaneous learning of all the components of the model. As shown in Figure 2, the supervised learning component assists the domains to be transferred in a better manner. Experiments on classification of handwritten digits (MNIST, USPS, and MADBase), and objects (Caltech-256 and Office) demonstrate that our method yields better performance than the state-of-the-art transfer sparse coding approach under the introduced supervised cross-domain learning setting.

2 Related Work

Sparse coding became an attractive method for learning cross-domain transfer models since Raina et al. (2007) demonstrated its ultimate effectiveness in leveraging large amounts of unlabeled data from various domains to improve the final classification model in the target domain. Lee et al. (2007) designed efficient optimization methods for sparse coding which significantly accelerated the learning process. Some of the subsequent works concentrated on applying sparse coding to the problem of cross-domain transfer semi-supervised learning with rigid constraint on target domain objects labeling. It was demonstrated how one can modify the sparsity constraint to improve the model further by preserving the geometrical relationships between the samples (Zheng et al. 2011), or also by minimizing the empirical distance measure between marginal distributions in the source and target domains (Long et al. 2013a).

Although these models worked in an unsupervised manner, it was known that sparse coding can utilize label information to learn better discriminative representations (Mairal et al. 2009). It was also demonstrated how a kernel-based model can be adopted in order to perform domain transfer (Duan et al. 2009; Duan, Tsang, and Xu 2012).

To the best of our knowledge, this paper is the first work that attempts to deviate from the classical transfer learning constraints on the source and target domains in order to learn discriminative sparse representations for both domains in a supervised fashion. In the rest of this paper, we will define a cross-domain learning problem, propose a unified supervised transfer sparse coding framework, and demonstrate how a small number of labeled objects in the target domain can significantly improve the classification accuracy.

3 Supervised Transfer Sparse Coding

In this section, we first define a cross-domain supervised learning problem. Then, we introduce a unified framework to solve the problem, which we called supervised transfer sparse coding (STSC). We split it into separate components and discuss them one after another: sparse coding, domain transfer, and SVM-based transfer correction. By introducing each of the components, we step-by-step refine the original sparsity constraint in the sparse coding objective function, and finally present the overall STSC framework.

3.1 Problem Definition

We denote the training set as $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N},\ \mathcal{D} \subset \mathbb{R}^D$, where N is the number of data samples, $\mathbf{x}_i$ is the vector of features of the i-th sample, and D is the dimensionality of the feature vector for each data sample. Further, the data is represented as matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$, where the i-th column is the feature vector of the i-th sample.

The training set is composed of the source domain subset $\mathcal{D}^s$ and the target domain subset $\mathcal{D}^t$, i.e. $\mathcal{D} = \mathcal{D}^s \sqcup \mathcal{D}^t$. We also denote N^s and N^t as the number of objects in source and target domains, respectively. All the samples from the source domain $\mathcal{D}^s$ are labeled, while only a few from the target domain $\mathcal{D}^t$ are. Both source and target domains share the same class space. The set of labeled samples is denoted as $\mathcal{D}_l$, and the set of unlabeled samples as $\mathcal{D}_u$, and their cardinalities are denoted as N_l and N_u, respectively. Data labels are represented by a vector $\mathcal{Y} = [y_1, \cdots, y_{N_l}] \in \mathbb{R}^{N_l}$. The number of classes is denoted by m.

We define the testing set in the same way and denote it $\tilde{\mathcal{D}}$. All the objects in $\tilde{\mathcal{D}}$ are considered as unlabeled, and they are also from both domains, i.e., $\tilde{\mathcal{D}} = \tilde{\mathcal{D}}^s \sqcup \tilde{\mathcal{D}}^t$. However, we suppose that the original domain of each object in $\tilde{\mathcal{D}}$ is unknown (Figure 1). In the following sections, we introduce additional notations, which are summarized in Table 1.

Given the data, the classification problem is to learn a model from the training set and apply it to the testing set with the maximum possible prediction accuracy for objects in both domains. In order to design such a robust model while using simple classifiers, we should be able to learn a new data representation.

3.2 Sparse Coding

Given a D-dimensional feature vector of a data sample $\mathbf{x} \in \mathbb{R}^D$ and a dictionary matrix $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_K] \in \mathbb{R}^{D \times K}$, where the k-th column is the k-th codeword and K is the number of codewords in the dictionary, sparse coding tries

Table 1: Summary of the notations used in this paper.

Notations	Descriptions
$\mathcal{D}, \mathbf{X}, \mathcal{Y}$	training dataset, its matrix, its vector of labels
$\tilde{\mathcal{D}}, \tilde{\mathbf{X}}, \tilde{\mathcal{Y}}$	testing dataset, its matrix, its vector of labels
$\mathcal{D}^s, \mathcal{D}^t, \tilde{\mathcal{D}}^s, \tilde{\mathcal{D}}^t$	training source, training target, test source, test target
$\mathcal{D}_u, \mathcal{D}_l, \tilde{\mathcal{D}}_u, \tilde{\mathcal{D}}_l$	unlabeled train, labeled train, unlabeled test, labeled test
$N_u, N_l, \tilde{N}_u, \tilde{N}_l$	# unlabeled/labeled samples in the training/testing set
$N^s, N^t, \tilde{N}^s, \tilde{N}^t$	# source/target samples in the training/testing set
m, K	# classes, # codewords in the dictionary
$\mathbf{U}, \mathbf{V}$	dictionary matrix, sparse codes matrix
$\mathbf{M}, \mathbf{L}, \tilde{\mathbf{M}}$	MMD, graph reg., unified reg. matrices
$\mathbf{W}, \mathbf{B}$	SVM hyperplane normal vectors, SVM intercept terms
$\boldsymbol{\Xi}, \mathbf{Y}$	SVM margins, one-hot encoding matrix of labels
λ, α, μ	sparsity penalty, MMD, and Graph-Laplacian weights
κ, c	SVM term weight, SVM coefficient

to reconstruct $\mathbf{x}$ by a sparse linear combination of the dictionary codewords

$$\mathbf{x} \approx \sum_{k=1}^{K} v_k \mathbf{u}_k = \mathbf{U}\mathbf{v}, \tag{1}$$

where $\mathbf{v} = [v_1, \cdots, v_K]^\top \in \mathbb{R}^K$ is the reconstruction coefficient vector – the sparse code – for the sample $\mathbf{x}$. In the above introduced notation, if we denote the matrix of sparse codes that corresponds to the objects $\mathbf{X}$ as $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_N] \in \mathbb{R}^{K \times N}$, the sparse coding optimization problem can be written as following

$$\begin{aligned} &\underset{\mathbf{U},\mathbf{V}}{min} \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 \\ &s.t.\ \|\mathbf{v}_j\|_0 \le T,\ j = 1, \cdots, N, \end{aligned} \tag{2}$$

where T is a positive constant, by $\|\cdot\|_0$ we denote l_0-norm, which is the number of non-zero elements in a vector, and by $\|\cdot\|_F^2$ we denote squared Frobenius norm, which is a sum of squares of matrix elements. It was shown that the exact determination of the sparsest possible codes is an NP-hard problem (Davis, Mallat, and Avellaneda 1997). Although, there were some effective greedy algorithms proposed (Chen, Billings, and Luo 1989; Tropp 2004), Donoho (2006) showed that the minimal l_1 solution is also the sparsest solution for the most of the large, underdetermined systems. This allows us to switch to a different problem

$$\begin{aligned} &\underset{\mathbf{U},\mathbf{V}}{min} \left\{ \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \lambda \sum_{i=1}^{N} \|\mathbf{v}_i\|_1 \right\} \\ &s.t.\ \|\mathbf{u}_k\|_2^2 \le 1,\ k = 1, \cdots, K, \end{aligned} \tag{3}$$

where λ is a regularization parameter. Due to the switch to l_1-norm, the problem happens to be convex in either $\mathbf{U}$ or $\mathbf{V}$. Hence, it can be solved by an iterative algorithm that alternates between l_1- and l_2-regularized least square problems that could be efficiently solved (Lee et al. 2007). The constraint $\|\mathbf{u}_k\|_2^2 \le 1$ is necessary due to the fact that reconstruction errors $\|\mathbf{x} - \mathbf{U}\mathbf{v}\|_2^2$ are invariant to simultaneous scaling of $\mathbf{U}$ by a scalar and $\mathbf{v}$ by its inverse. This constraint prevents $\mathbf{U}$ from an unbounded growth which defeats the purpose of the sparsity regularization.

3.3 Domain Transfer

To enforce seeking unified sparse codes for both domains, we regularize the model (3) with additional terms.

Maximum Mean Discrepancy (MMD) Regularization for sparse coding was recently introduced by Long et al. (2013a) and is based on the following additional term

$$\text{MMD} = \left\| \frac{1}{N^s} \sum_{i:\mathbf{x}_i \in \mathcal{D}^s} \mathbf{v}_i - \frac{1}{N^t} \sum_{j:\mathbf{x}_j \in \mathcal{D}^t} \mathbf{v}_j \right\|_2^2, \quad (4)$$

which is the l_2-norm of the difference between mean samples of each of the domains in the sparse coding space. Gretton et al. (2006) showed that MMD will approach zero if the distributions of the domains are the same.

MMD can be also rewritten in the following matrix form

$$\text{MMD} = \text{Tr}\left(\mathbf{V}\mathbf{M}\mathbf{V}^\top\right), \quad (5)$$

where the M-matrix is determined as following

$$M_{ij} = \begin{cases} 1/(N^s)^2, & \mathbf{v}_i, \mathbf{v}_j \in \mathcal{D}^s, \\ 1/(N^t)^2, & \mathbf{v}_i, \mathbf{v}_j \in \mathcal{D}^t, \\ -1/(N^s N^t), & otherwise. \end{cases} \quad (6)$$

Graph-Laplacian Regularization is another technique introduced by Zheng et al. (2011) which preserves the intrinsic geometrical properties of the data distributions. It is also used as an additional term for the sparse coding optimization function (3). Let $\mathbf{Q}$ be the k-nearest neighbor graph matrix of the data

$$\mathbf{Q}_{ij} = \begin{cases} 1, & \mathbf{v}_i \text{ is among k-nearest to } \mathbf{v}_j \text{ or vice versa}, \\ 0, & otherwise. \end{cases} \quad (7)$$

The Graph-Laplacian matrix is then defined as $\mathbf{L} = \mathbf{Q} - \mathbf{D}$, where $\mathbf{D} = diag(d_1, \cdots, d_N)$ and $d_i = \sum_{j=1}^N \mathbf{Q}_{ij}$. According to Zheng et al., the Graph-Laplacian regularization term will have the following form

$$\text{GL} = \text{Tr}\left(\mathbf{V}\mathbf{L}\mathbf{V}^\top\right). \quad (8)$$

Comparing the matrix form of the MMD term (5) with the Graph-Laplacian term, one can generalize these two in the following way

$$\text{Tr}\left(\mathbf{V}\tilde{\mathbf{M}}\mathbf{V}^\top\right) = \text{Tr}\left(\mathbf{V}\left(\alpha\mathbf{M} + \mu\mathbf{L}\right)\mathbf{V}^\top\right), \quad (9)$$

where α and μ are the tuning parameters for MMD and Graph-Laplacian terms, respectively. The final modified sparse coding optimization problem has the following form

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} \left\{ \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \lambda \sum_{i=1}^N \|\mathbf{v}_i\|_1 + \text{Tr}\left(\mathbf{V}\tilde{\mathbf{M}}\mathbf{V}^\top\right) \right\} \\ &s.t.\ \|\mathbf{u}_k\|_2^2 \le 1,\ k = 1, \cdots, K. \end{aligned} \quad (10)$$

Long et al. (2013a) introduced this model, named transfer sparse coding (TSC), and showed that the representations learned by TSC can be used for building a robust and accurate image classifier.

Supervised Transfer Correction is an SVM-based term which further enhances the model (10) by using label information. It exploits the relaxation of the cross-domain transfer learning problem that allows having some labeled objects from the target domain in the training set.

For every class, we introduce a binary linear SVM classifier, the objective function of which we integrate into (10). SVM hyperplane normal vectors for every class are coupled together as columns of matrix $\mathbf{W} \in \mathbb{R}^{D \times m}$, and intercept parameters are stacked into vector $\mathbf{b} \in \mathbb{R}^m$ which is copied for every labeled object in the training set and grouped into matrix $\mathbf{B} = [\mathbf{b}, \cdots, \mathbf{b}] \in \mathbb{R}^{m \times N_l}$. We also group together all the margins for the labeled training objects with respect to all the classes into matrix $\mathbf{\Xi} \in \mathbb{R}^{m \times N_l}$. This notation allows us to write the final model, we named supervised transfer sparse coding (STSC), as following

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V},\mathbf{W}} \left\{ \overbrace{\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \lambda \sum_{i=1}^N \|\mathbf{v}_i\|_1}^{\text{sparse coding}} + \overbrace{\text{Tr}\left(\mathbf{V}\tilde{\mathbf{M}}\mathbf{V}^\top\right)}^{\text{transfer \& geometry}} + \right. \\ &\left. \underbrace{\kappa\left(\frac{1}{2}\|\mathbf{W}\|_F^2 + c\mathbf{1}^T\mathbf{\Xi}\mathbf{1}\right)}_{\text{multi-class SVM}} \right\} \\ &s.t.\ \|\mathbf{u}_k\|_2^2 \le 1,\ k = 1, \cdots, K, \\ &\quad \mathbb{1} - \mathbf{\Xi} \preceq \mathbf{Y} \circ (\mathbf{W}^\top\mathbf{V} + \mathbf{B}),\ \mathbf{\Xi} \succeq 0, \end{aligned} \quad (11)$$

where κ is a tuning parameter, c is SVM coefficient, $\mathbb{1}$ is a matrix of ones, $\mathbf{1}$ is a vector of ones, $\circ$ denotes Hadamard product, and $\preceq, \succeq$ stand for element-wise inequalities.

It is important to notice that for different sparse representations different classes will become better or worse separable. This model can sacrifice separability of some classes in order to better classify the others. For simplicity, we have introduced single κ parameter. However, one can choose to assign each class a parameter κ_c in order to capture relative importance of classes. One can also notice that due to the multi-class SVM term, STSC model (11) does not reduce to TSC model (10) even when all the target objects in the training set are unlabeled, i.e. it remains different even in the classical transfer learning setting.

4 Three-Step Optimization

We propose a three-step iterative algorithm for efficiently solving the STSC optimization problem. The Lagrangian function for (11) is the following

$$\begin{aligned} &\mathcal{L}(\mathbf{W}, \mathbf{B}, \mathbf{\Xi}, \mathbf{U}, \mathbf{V}, \mathbf{\Gamma}, \mathbf{\Theta}, \boldsymbol{\nu}) = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \frac{\kappa}{2}\|\mathbf{W}\|_F^2 + \\ &\text{Tr}\left(\mathbf{V}\tilde{\mathbf{M}}\mathbf{V}^\top\right) + \lambda \sum_{i=1}^N \|\mathbf{v}_i\|_1 + \sum_{k=1}^K \nu_k \left(\|\mathbf{u}_k\|_2^2 - 1\right) + \\ &\mathbf{1}^T\left[(\kappa c\mathbb{1} + \mathbf{\Theta})\mathbf{\Xi} + \mathbf{\Gamma} \circ \left(\mathbb{1} - \mathbf{\Xi} - \mathbf{Y} \circ (\mathbf{W}^\top\mathbf{V} + \mathbf{B})\right)\right]\mathbf{1}, \end{aligned} \quad (12)$$

where $\mathbf{\Gamma}, \mathbf{\Theta} \in \mathbb{R}^{m \times N_l}, \boldsymbol{\nu} \in \mathbb{R}^K$ are the dual variables associated with corresponding inequality constraints. According

to the duality theory, we can solve the following problem

$$\begin{aligned}\max_{\boldsymbol{\Gamma},\boldsymbol{\Theta},\boldsymbol{\nu}} \quad & \min_{\mathbf{W},\mathbf{B},\boldsymbol{\Xi},\mathbf{U},\mathbf{V}} \mathcal{L}(\mathbf{W},\mathbf{B},\boldsymbol{\Xi},\mathbf{U},\mathbf{V},\boldsymbol{\Gamma},\boldsymbol{\Theta},\boldsymbol{\nu}) \\ s.t.\ & \boldsymbol{\Gamma} \succeq 0, \boldsymbol{\Theta} \succeq 0,\ \boldsymbol{\nu} \succeq 0.\end{aligned} \tag{13}$$

The first order optimality conditions over $\mathbf{W}, \mathbf{B}, \boldsymbol{\Xi}$, will have the following form

$$\begin{aligned}\mathbf{W} &= (\boldsymbol{\Gamma} \circ \mathbf{Y})\mathbf{V}^T, \\ 0 &= (\boldsymbol{\Gamma} \circ \mathbf{Y})\mathbf{1}, \\ \boldsymbol{\Gamma} &\preceq \kappa c,\end{aligned} \tag{14}$$

Plugging (14) and (12) into (13), we end up with the following optimization problem

$$\begin{aligned}\max_{\boldsymbol{\Gamma},\boldsymbol{\nu}} \min_{\mathbf{U},\mathbf{V}} \ & \mathcal{L}_{\mathcal{D}}(\mathbf{U},\mathbf{V},\boldsymbol{\Gamma},\boldsymbol{\nu}) \\ & s.t.\ (\boldsymbol{\Gamma} \circ \mathbf{Y})\,\mathbf{1} = 0, \\ & \quad 0 \preceq \boldsymbol{\Gamma} \preceq \kappa c,\ \boldsymbol{\nu} \succeq 0,\end{aligned} \tag{15}$$

where $\mathcal{L}_{\mathcal{D}}(\mathbf{U},\mathbf{V},\Gamma,\boldsymbol{\nu})$ has the following form

$$\begin{aligned}&\mathcal{L}_{\mathcal{D}}(\mathbf{U},\mathbf{V},\boldsymbol{\Gamma},\boldsymbol{\nu}) = \\ &\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \lambda \sum_{i=1}^{N} \|\mathbf{v}_i\|_1 + \mathbf{1}^T\boldsymbol{\Gamma}\mathbf{1} + \\ &\sum_{k=1}^{K} \nu_k \left(\|\mathbf{u}_k\|_2^2 - 1\right) + \mathrm{Tr}\left(\mathbf{V}\left(\tilde{\mathbf{M}} - \frac{1}{2}\boldsymbol{\Psi}\right)\mathbf{V}^\top\right),\end{aligned} \tag{16}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\Gamma} \circ \mathbf{Y})^\top (\boldsymbol{\Gamma} \circ \mathbf{Y})$. Problem (15) can be efficiently solved via the following three-step iterative algorithm (summarized in Algorithm 1).

1. **Sparse Codes Learning** is done by optimizing

$$\min_{\mathbf{V}} \left\{ \begin{array}{l} \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \lambda \sum_{i=1}^{N} \|\mathbf{v}_i\|_1 + \\ \mathrm{Tr}\left(\mathbf{V}\left(\tilde{\mathbf{M}} - \frac{1}{2}\boldsymbol{\Psi}\right)\mathbf{V}^\top\right) \end{array} \right\}. \tag{17}$$

   This problem can be efficiently solved by a modified feature-sign algorithm (Zheng et al. 2011).

2. **Dictionary Learning** is performed by solving

$$\begin{aligned}\max_{\boldsymbol{\nu}} \min_{\mathbf{U}} & \left\{ \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2 + \sum_{k=1}^{K} \nu_k \left(\|\mathbf{u}_k\|_2^2 - 1\right) \right\} \\ & s.t.\ \boldsymbol{\nu} \succeq 0,\end{aligned} \tag{18}$$

   using the algorithm proposed by Lee et al. (2007).

3. **Learning SVM**. Finally, we search for the optimal classifier parameters

$$\begin{aligned}\min_{\boldsymbol{\Gamma}} & \left\{ \frac{1}{2}\,\mathrm{Tr}\left(\mathbf{V}\boldsymbol{\Psi}\mathbf{V}^\top\right) - \mathbf{1}^T\boldsymbol{\Gamma}\mathbf{1} \right\} \\ s.t.\ & (\boldsymbol{\Gamma} \circ \mathbf{Y})\,\mathbf{1} = 0, \\ & 0 \preceq \boldsymbol{\Gamma} \preceq \kappa\, c,\end{aligned} \tag{19}$$

   which is a convex quadratic programming (QP) problem that can be efficiently solved by an interior-point method.

Algorithm 1 STSC: Supervised Transfer Sparse Coding

Input: $\mathbf{X}$ – training data, $\mathcal{Y}$ – labels.
Input: $\alpha, \mu, \kappa, \lambda, c$ – parameters, *iter_num* – number of iterations.
1: Build the MMD matrix $\mathbf{M}$, Graph-Laplacian matrix $\mathbf{L}$, and one-hot encoding matrix $\mathbf{Y}$ of labels for the labeled objects.
2: $\mathbf{U} \leftarrow$ uniform random matrix with zero mean for each column.
3: $\boldsymbol{\Gamma} \leftarrow 0$, $\boldsymbol{\Psi} \leftarrow 0$.
4: **for** $t = 1, \cdots, iter\_num$ **do**
5: Find $\mathbf{V}$ by solving **Sparse Codes Learning** subproblem.
6: Find $\mathbf{U}$ by solving **Dictionary Learning** subproblem.
7: Find $\boldsymbol{\Gamma}$ and compute $\boldsymbol{\Psi}$ by solving **SVM** subproblem.
Output: $\mathbf{U}$ – dictionary, $\mathbf{V}$ – sparse codes.

It is important to notice that $\boldsymbol{\Psi}$ matrix learned on the third step of the algorithm is actually subtracted from the transfer and geometry matrix $\tilde{\mathbf{M}}$. We call this process *supervised transfer correction*, since a supervised model directly influences the transfer matrix and eventually allows us to learn a more discriminative dictionary $\mathbf{U}$. This dictionary is further used to construct sparse representations for the testing data.

5 Experiments

In this section, we present experimental results that verify both of our hypothesis: (1) a small number of labeled data can significantly improve TSC's accuracy; (2) the proposed STSC is able to further improve the performance on the subsequent classification tasks.

5.1 Data Description

In order to be consistent with Long et al. (2013a), the following well known benchmark datasets were selected: handwritten digits (USPS, MNIST, and MADBase), Amazon and Caltech-256 dataset of object images.

USPS[1] comprises 9,298 images of hand-written Arabic digits of size 16×16 pixels.

MNIST[2] contains 70,000 images of hand-written Arabic digits. Each image has size of 28×28 pixels.

MADBase[3] is a less known dataset of 70,000 images of hand-written Hindi digits. It was designed to have similar parameters as MNIST, so the images from MADBase have size of 28×28. We decided to work with this dataset since learning a unified model to accurately classify both Arabic and Hindi digits is a vivid example of the power of transfer learning (Figure 3).

All the digit datasets were rescaled to size of 16×16, grayscaled and then normalized. In order to accurately evaluate the methods, we randomly sampled training and testing subsets from USPS, MNIST, and MADBase several times with different random seeds. We used USPS database as the source domain, and MNIST or MADBase as the target. For the training set, we sampled 100 objects per class from the source, and 100 objects per class from the target; for the testing set, we used 100 objects per class from the source as well as from the target (Table 2).

[1] http://www-i6.informatik.rwth-aachen.de/∼keysers/usps.html
[2] http://yann.lecun.com/exdb/mnist
[3] http://datacenter.aucegypt.edu/shazeem/

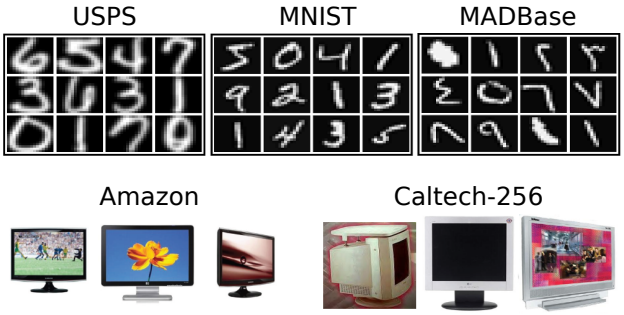


Figure 3: Samples from digits (MNIST, USPS, MADBase) and objects (Amazon and Caltech-256) datasets.

Table 2: Statistics of the benchmark datasets.

Dataset	#Obj. p/class (src / tgt)	#Features	#Classes
USPS	100 / 100	256	10
MNIST	100 / 100	256	10
MADBase	100 / 100	256	10
Caltech	20 / 20	800	10
Amazon	20 / 20	800	10

Amazon is the part of the **Office** (Gong et al. 2012) dataset that has images downloaded from online merchants.

Caltech-256 is a standard database of object images of 256 categories (Griffin, Holub, and Perona 2007).

In our experiments, we used a preprocessed version of Amazon and Caltech-256 taken from Office+Caltech constructed by Gong et al. (2012). Since the number of objects per class in this dataset was about 200 in total, we sampled 20 objects per class for each the source and the target domain for each the training and the testing sets (Table 2).

5.2 Baseline Methods

For the baseline methods we selected the following

- Logistic Regression (LR)
- Support Vector Machine (SVM)
- Transfer Sparse Coding (TSC) + LR
- Transfer Sparse Coding (TSC) + SVM

Performances of all the methods were evaluated for different ratios of labeled objects in the target subset of the training set. Baseline methods performances were compared against our proposed Supervised Transfer Sparse Coding (STSC) followed by SVM and by LR.

5.3 Experimental and Evaluation Details

According to the relaxed cross-domain transfer learning, we trained each of the baseline methods on a sampled training set and then tested on the corresponding testing set. Training and testing sets were re-sampled several times, and the average performance is reported here.

Following Long et al. (2013a), we applied PCA and kept 98% of information in the largest eigenvectors to reduce the data dimensionality. We performed all algorithms in the reduced PCA space.

We fixed the number of basis vectors $k = 128$ and the number of nearest neighbors used for Laplacian graph construction $p = 5$. We also performed a grid search for optimal parameters on a grid used in Long et al. (2013a) for TSC method. Once we found a set of optimal parameters for our supervised classification case: $\lambda = 0.1, \alpha = 10^4, \mu = 1$, we fixed them also for STSC. Then, we tuned the SVM term weight κ and got an optimal value for it $\kappa = 0.35$. SVM coefficient was set $c = 1$. The number of iterations for TSC and STSC was $T = 100$. Tuning was performed with respect to USPS as the source domain and MNIST as the target.

It is important to notice the difference between semi-supervised setting used by Long et al. (2013a) and purely supervised used in this paper. On the learning step, we only can obtain the dictionary and the sparse codes for the training objects. Hence, on the testing step, for TSC and STSC we have to first learn sparse representations for the testing data by solving (17), and only then we can apply a supervised model (LR or SVM). We also should notice that we are solving (17) for the testing objects with $\mathbf{\Psi} = 0$ and $\tilde{\mathbf{M}} = GL$, since no information is available regarding their labels or their domains. Once we obtained sparse codes for the testing objects, we trained generic SVM and LR models on the sparse representations of the training data and applied them to the testing sparse codes.

Finally, we used classification accuracy on the source and the target domain of the testing data to measure performance

$$Accuracy_s = \frac{|\mathbf{x} : \mathbf{x} \in \tilde{\mathcal{D}}^s \wedge \hat{y}(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} : \mathbf{x} \in \tilde{\mathcal{D}}^s|},$$

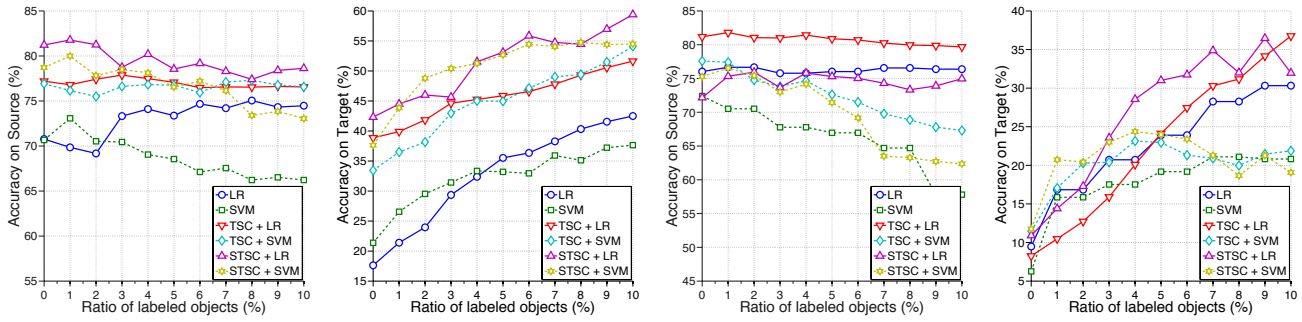
$$Accuracy_t = \frac{|\mathbf{x} : \mathbf{x} \in \tilde{\mathcal{D}}^t \wedge \hat{y}(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} : \mathbf{x} \in \tilde{\mathcal{D}}^t|},$$

where $\tilde{\mathcal{D}}^s$ and $\tilde{\mathcal{D}}^t$ are the target and the source domains of the testing set, $y(\mathbf{x})$ and $\hat{y}(\mathbf{x})$ are the true and the predicted label of $\mathbf{x}$, respectively.

5.4 Experimental Results Discussion

Experimental results for digit datasets are presented in Figure 4. Along the X-axis the ratio of training labeled objects from the target domain is changed. One can see that even if a small number of the training target domain objects is labeled, classification accuracy significantly increases for all the methods on the target domain. On the other hand, the accuracy on the source domain does not improve. This justifies the relaxation of the cross-domain learning that allows to label a fraction of the target domain objects in order to significantly gain classification accuracy even when using unsupervised methods such as TSC. Moreover, representations learned by STSC eventually yield superior performance over the baseline methods. This indicates that such representations work better for classification, and that the supervised transfer correction works as expected.

For all the datasets with 5% labeled target domain training objects, experimental results are summarized in Table 3. One can see that STSC performance remains superior for all the considered datasets.



(a) USPS & MNIST. (b) USPS & MNIST. (c) USPS & MADBase. (d) USPS & MADBase.

Figure 4: Classification accuracy of STSC+LR, STSC+SVM, and the baseline methods for different ratios of labeled target domain objects in the training set. (a) Accuracy on the source domain of the testing set for USPS & MNIST; (b) accuracy on the target domain of the testing set for USPS & MNIST; (c) accuracy on the source domain of the testing set for USPS & MADBase; (d) accuracy on the target domain of the testing set for USPS & MADBase. Methods with LR classifiers are drawn as solid lines; methods with SVM classifiers are depicted as dashed lines.

Table 3: Classification accuracy on the target domain of the test set (5% labeled target domain objects in the training set). Accuracy is averaged over five resamplings of the training and testing sets. Standard deviations are also presented.

Dataset	USPS – MNIST	USPS – MADBase	Caltech – Amazon
LR	35.5 ± 0.8	24.1 ± 3.0	39.7 ± 1.9
SVM	33.2 ± 1.5	19.3 ± 4.2	34.6 ± 3.7
TSC+LR	45.8 ± 1.8	24.1 ± 3.8	38.3 ± 2.1
TSC+SVM	44.9 ± 2.4	22.9 ± 4.3	32.5 ± 1.6
STSC+SVM	52.6 ± 3.8	24.0 ± 4.8	41.5 ± 2.5
STSC+LR	$\mathbf{53.1 \pm 2.2}$	$\mathbf{31.0 \pm 3.5}$	$\mathbf{43.0 \pm 2.1}$

One might doubt that the achieved performance is the merit of knowledge transfer, arguing that 10% of target subset of the training data labeled is almost enough to be able to generalize. This is indeed true: The higher the target labeled ratio, the less transfer plays role. However, our experiments explicitly suggest that for labeled ratios below 5%, transfer from the source domain is crucial: TSC+LR trained only on the labeled 2% of the training target data yields only 28% of accuracy on the testing target objects. If we train the same method with the training source domain assisting in an unsupervised fashion, the accuracy jumps up to 41%. If use STSC+LR and enable supervised correction of the transfer, we get 46% of the accuracy on the testing target domain.

5.5 Limitations

Although STSC outperforms other methods in the given setting of a relaxed supervised cross-domain transfer classification, it is important to mention that STSC's additional tuning parameter (SVM term weight κ) is sensitive in some cases. One should keep it relatively small, to make the method finally converge. Otherwise, supervised transfer correction would possibly be too large, and the three-step optimization procedure will remain oscillating without convergence. In such case, STSC will perform relatively poor in comparison with classical TSC method.

6 Conclusion and Future Work

In this paper, we demonstrate that a small number of labeled objects from the target domain can significantly improve performance of the state-of-the-art transfer sparse coding methods. We propose a supervised transfer sparse coding (STSC) framework for learning discriminative representations in a relaxed cross-domain transfer learning setting. Using STSC, we show that by simultaneously optimizing sparse representations, domain transfer, and supervised classification, learned representations can further improve the subsequent classification accuracy.

In the future, we plan to extend STSC framework with different supervised transfer correction terms based on other classifiers, e.g., logistic regression and linear discriminant analysis (LDA).

7 Acknowledgments

The authors thank Xiangliang Zhang for a helpful discussion, and Mingsheng Long, Honglak Lee, and Boqing Gong for making their code and data available. Support for this work was provided in part by King Abdullah University of Science and Technology (KAUST), Saudi Arabia, and US National Science Foundation (Grant No. DBI-1322212).

References

Chen, S.; Billings, S. A.; and Luo, W. 1989. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control* 50(5):1873–1896.

Davis, G.; Mallat, S.; and Avellaneda, M. 1997. Adaptive greedy approximations. *Constructive approximation* 13(1):57–98.

Donoho, D. L. 2006. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics* 59(6):797–829.

Duan, L.; Tsang, I. W.; Xu, D.; and Maybank, S. J. 2009. Domain transfer svm for video concept detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1375–1381. IEEE.

Duan, L.; Tsang, I. W.; and Xu, D. 2012. Domain transfer multiple kernel learning. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, volume 34, 465–479. IEEE.

Gao, S.; Tsang, I. W.; Chia, L.-T.; and Zhao, P. 2010. Local features are not lonely–laplacian sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3555–3561. IEEE.

Geng, B.; Tao, D.; and Xu, C. 2011. Daml: Domain adaptation metric learning. *Image Processing, IEEE Transactions on* 20(10):2980–2989.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2066–2073. IEEE.

Gopalan, R.; Li, R.; and Chellappa, R. 2011. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 999–1006. IEEE.

Gretton, A.; Borgwardt, K.; Rasch, M. J.; Scholkopf, B.; and Smola, A. J. 2006. A kernel method for the two-sample problem. In *Advances in Neural Information Processing Systems 20, NIPS*.

Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 object category dataset.

Huang, K., and Aviyente, S. 2006. Sparse representation for signal classification. In *Advances in Neural Information Processing Systems*, 609–616.

Jiang, Z.; Lin, Z.; and Davis, L. S. 2011. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1697–1704. IEEE.

Lee, H.; Battle, A.; Raina, R.; and Ng, A. 2007. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems* 19:801.

Liu, Y.; Wu, F.; Zhang, Z.; Zhuang, Y.; and Yan, S. 2010. Sparse representation using nonnegative curds and whey. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3578–3585. IEEE.

Long, M.; Ding, G.; Wang, J.; Sun, J.; Guo, Y.; and Yu, P. S. 2013a. Transfer sparse coding for robust image representation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 407–414. IEEE.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2013b. Transfer feature learning with joint distribution adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*.

Mairal, J.; Ponce, J.; Sapiro, G.; Zisserman, A.; and Bach, F. R. 2009. Supervised dictionary learning. In Koller, D.; Schuurmans, D.; Bengio, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems 21*, 1033–1040. Curran Associates, Inc.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on* 22(10):1345–1359.

Pan, S. J.; Ni, X.; Sun, J.-T.; Yang, Q.; and Chen, Z. 2010a. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, 751–760. ACM.

Pan, W.; Xiang, E. W.; Liu, N. N.; and Yang, Q. 2010b. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, volume 10, 230–235.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on* 22(2):199–210.

Pan, S. J.; Kwok, J. T.; and Yang, Q. 2008. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, 677–682.

Raina, R.; Battle, A.; Lee, H.; Packer, B.; and Ng, A. Y. 2007. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, 759–766. ACM.

Tropp, J. A. 2004. Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE Transactions on* 50(10):2231–2242.

Wang, J. J.-Y.; Bensmail, H.; and Gao, X. 2014. Feature selection and multi-kernel learning for sparse representation on a manifold. *Neural Networks* 51(0):9–16.

Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; and Gong, Y. 2010. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3360–3367. IEEE.

Wang, H.; Nie, F.; Huang, H.; and Ding, C. 2011. Dyadic transfer learning for cross-domain image classification. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 551–556. IEEE.

Wang, J. J.-Y.; Bensmail, H.; Yao, N.; and Gao, X. 2013. Discriminative sparse coding on multi-manifolds. *Knowledge-Based Systems* 54(0):199–206.

Yang, J.; Yu, K.; Gong, Y.; and Huang, T. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1794–1801. IEEE.

Zheng, M.; Bu, J.; Chen, C.; Wang, C.; Zhang, L.; Qiu, G.; and Cai, D. 2011. Graph regularized sparse coding for image representation. *Image Processing, IEEE Transactions on* 20(5):1327–1336.

Zhu, Y.; Chen, Y.; Lu, Z.; Pan, S.; Xue, G.-R.; Yu, Y.; and Yang, Q. 2011. Heterogeneous transfer learning for image classification. In *AAAI Conference on Artificial Intelligence*.

Zhuang, F.; Luo, P.; Shen, Z.; He, Q.; Xiong, Y.; Shi, Z.; and Xiong, H. 2012. Mining distinction and commonality across multiple domains using generative model for text classification. *Knowledge and Data Engineering, IEEE Transactions on* 24(11):2025–2039.