Safety Multiclass Incremental Transfer Learning

Shuang Ao
Department of Computer Science
Western University
London, Ontario
Email: sao@uwo.ca

Charles X. Ling
Department of Computer Science
Western University
London, Ontario
Email: cling@csd.uwo.ca

***Abstract*—abs**

I. Introduction

Our human beings can learn new things progressively in time. At a very young age, we can distinguish thousands of different objects and this ability can increasingly develop as we grow up. Moreover, we actually don't treat a new concept in isolation, but try to connect the new concept to the knowledge we already learned, which is referred to as transfer learning. For example, we recognize the animal zebra by referring it to the normal horse with distinctive black and white striped coats. Given a classification task of the target problem, transfer learning works on the scenario that knowledge learned from one or several prior (source) tasks can help the target learning task. Based on this, how to utilize the knowledge from multiple sources leads to the research of Multiple Source Transfer Learning (MSTL). Transferring the knowledge from multiple priors can make the learning procedure extremely efficient by mining the recurrent patterns as well as inferring inductively on the target task [1]. Taking advantage of this, the first implementation proposed by [2] using Bayesian approach shows that even with a single example, transfer learning can still get impressive results. Some methods using discriminative approach are proposed in recent years [1] [3] [4]. Previous s-tudy shows that the more prior knowledge the system acquired, the easier a new concept can be learned [5].

Transfer learning between different image databases is a very popular topic in recent years due to the fast growing vision-based applications. Multiclass incremental transfer learning (MITL) aims to add a new category model to the known source models by leveraging over them, while at the same time preserving their classification abilities [3]. Based on this, we extend it to transfer the known source models to the identical target categories with different data distribution and add a new target category model as well. It is worthy to notice that when the data of the source and target tasks are drawn from the same distribution, our problem is decayed to the MITL problem. In MITL, it assumes that we can only access to the models of the source task rather than the original images. Following this setting, in our case, we have to face another challenge, negative transfer. Transferring knowledge can consistently boost the classification performance (positive transfer) is based on the fact that the learning procedure can benefit from sufficient related prior knowledge. In some situation, where the source domain and target one are not related, transferring the knowledge between them could even degrade the performance of the classifier of target task, which is referred to as negative transfer (see Figure 1). Negative transfer is a general issue even if we transfer knowledge between identical categories in different tasks due to the mismatched data distribution. In our case, the new added category could change the data distribution totally and transfer learning could more easily suffer from negative transfer (see Figure 2). How to avoid negative transfer is still an open question in transfer learning [6]. Specifically, how to the measure the transferability of different prior knowledge and obtain a comprehensive and accurate measurement to prevent negative transfer should be studied profoundly. Previous works suggest that, to better utilizing the prior knowledge to reduce negative transfer, decision of the algorithm should be made by combining the prior and empirical knowledge (knowledge obtained from specific target task) instead of aggressively ignoring or utilizing all the prior knowledge [1] [3] [7] [8].

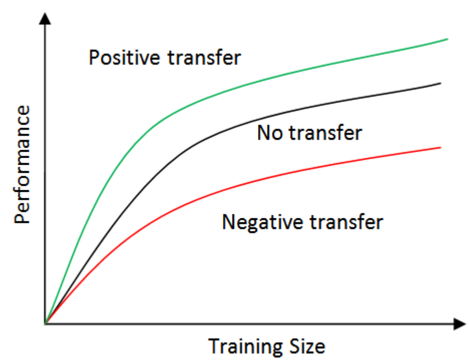


Fig. 1: Positive transfer VS Negative transfer. Relying on unrelated priors could lead to negative transfer.

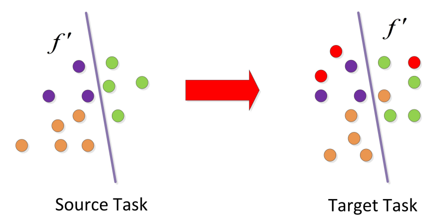


Fig. 2: Negative transfer may happen when we transfer prior knowledge f' to target one. Points with different color represent different categories. The data distribution would change even for identical categories in different task. The red points denote the new category in target task.

In MITL, previous works focus on how to utilize the prior knowledge to preserve the performance for positive transfer rather than avoiding negative transfer. In our case, we focus on preventing negative transfer as well as boosting the performance for positive transfer and propose our method Safety Multiclass Incremental Transfer Learning (SMITLe). We use Least Square Support Vector Machine (LS-SVM) [9] as our basic model. The decision of each binary LS-SVM is the linear combination of the prior knowledge and empirical knowledge controlled by some transfer parameters. To measure the transferability of each prior knowledge, we estimate our transfer parameters using closed-form leave-one-out (LOO) error. Previous works theoretically suggest that closed-from LOO error can be an efficient way for parameter estimation with a small training set [10] [11]. Then we propose our objective function that can balance the weight between the prior knowledge and empirical knowledge from target task. We also provide the theoretical proof that the transfer parameters optimized by our objective function can prevent negative transfer. Extensive empirical experiments show that other transfer learning baselines suffer from negative transfer while SMITLe can autonomously ignore the unrelated prior knowledge to prevent negative transfer. Then, we also show that when the prior knowledge is highly related to the target task (positive transfer), our method can outperform the transfer learning baselines by aggressively exploiting the prior knowledge.

The rest of this paper is organized as follow:

II. Related Works

The motivation of transfer knowledge between different domains is to apply the previous information from the source domain to the target one, assuming that there exists certain relationship, explicit or implicit, between the feature space of these two domains [12]. Technically, previous work can be concluded into solving the following three issues: what, how and when to transfer [1].

What to transfer. Previous work tried to answer this question from three different aspects: selecting transferable instances, learning transferable feature representations and transferable model parameters. Instance-based transfer learning assume that part of the instances in the source domain could be re-used to benefit the learning for the target domain. Lim et al. proposed a method of augmenting the training data by borrowing data from other classes for object detection [13]. Learning transferable features means to learn common feature that can alleviate the bias of data distribution in target domain. Recently, Long et al. proposed a method that can learn transferable features with deep neural network and showed some impressive results on the benchmarks [14]. Parameter transfer approach assumes that the parameters of the model for the source task can be transfered to the target task. Yang et al. proposed Adaptive SVMs by transferring parameters by incorporating the auxiliary classifier trained from source domain [7]. On top of Yang's work, Ayatar et al. proposed PMT-SVM that can determine the transfer regularizer according to the target data automatically [8]. Tommasi et al. proposed Multi-KT that can utilize the parameters from multiple source models for the target classes [1]. Kuzborskij et al. proposed a similar method to learn new categories by leveraging over the known source [3].

When and how to transfer. The question *when to transfer* arises when we want to know if the information acquired from previous task is relevant to the new one (i.e. in what situation, knowledge should not be transfered). *How to transfer* the prior knowledge effectively should be carefully designed to prevent inefficient and negative transfer. Some previous work consists in using generative probabilistic method [15] [16] [17]. Bayesian learning methods can predict the target domain by combining the prior source distribution to generate a posterior distribution. Alternatively, some previous max margin methods show that it is possible to learn from a few examples by minimizing the Leave-One-Out (LOO) error for the training model [3] [18]. Previous work shows that there is a closed-form implementation of LOO cross-validation that can generate unbiased model estimation for LS-SVM [11].

Our work correspond to the context above. In this paper, we propose a method based on parameter transfer approach with LS-SVM. We address our work on how to prevent negative transfer when the source data is not accessible. By optimizing the convex objective funtion, our method can autonomously adjust the transfer parameters for different prior knowledge. We theoretically and empirically show that, without any data distrbution assumption, the superior bound of the training loss for our transfer method is the loss of a method learning directly (i.e. without using any prior knowledge). This indecates that when the prior knowledge hurts the transfer procedure, our method can avoid negative transfer. Extensive experiments also show that when the prior knowledge is very related (positive transfer), our method can outperform other methods by relying on the decision of prior knowledge greatly.

III. Problem Statement

SMITLe works in the following scenario. There is a image dataset (source data) containing N categories and a classifier trained from this dataset to distinguish these N categories. This (source) classifier and the features used to learn it is publicly accessible while the dataset itself is private (unknown distribution). Now we collect our own image dataset (target data) coming from $N+1$ categories. This target dataset consists of N identical categories to the source data and one new category related to the previous N categories. In order to train a new classifier for our new task, we would expect our classifier to get improved performance with respect to

- Maximize positive transfer. If these two tasks are highly related, our algorithm should transfer the prior knowledge aggressively. In some cases where the prior knowledge is very informative, the final decision of the classifier should be mainly rely on prior knowledge.
- Minimize negative transfer. If the knowledge between these two task is unrelated, the algorithm should be able to dispose the unrelated knowledge autonomously. In the worst case, none of the prior knowledge is related and the classifier should be as good as classifier trained merely from target data.

In this paper, we focus our work on transferring the knowledge with LS-SVM as the classifier for multi-class transfer problem. In the following we briefly introduce the mathermatical setting of our problem and show

A. LS-SVM Setting and Definition

Here we introduce the notations used in the rest of the paper. We use any letter with apostrophe to denote the information from the source data, e.g. if $f(x)$ denotes the model for the target task, $f'(x)$ denotes the model for the source one.

TABLE I: useful notations in this paper

$f'(x)$	binary function for source task
$f(x)$	binary function for target task
$\phi(x)$	function mapping the input sample into a high dimensional feature space.
$K(x,x)$	kernel matrix with $\phi(x_i)\cdot\phi(x_j)$ corresponding to its element (i,j)
X	instance matrix with each row representing one instance
W	(N+1)-column hyperplane matrix for target task. Each column represents one hyperplane of a binary model
W'	hyperplane matrix for the source task
a'	the Lagrangian multiplier matrix for source problem. Each column represents a set of
a	the Lagrangian multiplier matrix for target problem
b', b	the bias vector for source and target task
a_i, w_i	i_{th} column of matrix a and w
d_γ	diagonal matrix with$[\gamma_1, ..., \gamma_N]$ in its main diagonal
β	row vector $[\beta_1, ..., \beta_N]$ to control the prior knowledge for the new category
ε_{ny_i}	loss parameter. $\varepsilon_{ny_i} = 1$ if $n = y_i$ and 0 otherwise

Assume that, for our $(N+1)$-category target task, $x \in \mathcal{X}$ and $y \in \mathcal{Y} = \{1, 2, ..., N+1\}$ are the input vector and output for the learning task respectively. Meanwhile, we have a set of binary linear classifiers $f'_n(x) = \phi(x)w'_n + b'_n$, for $n = 1, ..., N$ trained from an unknown distribution with One-Versus-All (OVA) strategy. Now we want to learn a set of new classifier $f_n(x) = \phi(x)w_n + b_n, n = 1, ..., N+1$, so that example x is assigned to the category j if $j \equiv \arg\max_{n=1,...,N+1}\{f_n(x)\}$. In LS-SVM, the solution of the model parameters (w_n, b_n) can be found by solving the following optimization problem:

$$\textbf{min} \quad R(w_n) + \frac{C}{2}\sum_{i}^{l}(Y_{i,n} - \phi(x_i)w_n - b_n)^2$$

Where $R(w_n)$ is the regularization term to guarantee good generalization performance and avoid overfitting. $\mathbf{Y}$ is a encoded label matrix so that $Y_{in} = 1$ if $y_i = n$ and -1 otherwise.

In classic LS-SVM setting, the regularization term is set to $\frac{1}{2}\|w_n\|^2$ and the optimal $w_n = \phi(X)^T\alpha_n$ while the parameters $(\aleph_n, b_n)$ can be found by solving

$$\begin{bmatrix} K(X,X) + \frac{1}{C}\mathrm{I} & \mathbf{1} \\ \mathbf{1^T} & 0 \end{bmatrix}\begin{pmatrix} \alpha_n \\ b_n \end{pmatrix} = \begin{pmatrix} Y_n \\ 0 \end{pmatrix} \tag{1}$$

Here I is the identity matrix and $\mathbf{1}$ is a column vector with all its elements equal to 1.

Now our task can be divided into two separate part: learning the N overlapped categories and the new category. We know that the source and target share N categories. From previous work [7], the regularization term can be written as $\frac{1}{2}\|w_n - \gamma_n w'_n\|^2$. Here, γ_n is the regularization parameter controlling the amount of transfer. For the task for new category, we can use multi-source kernel learning strategy in [1]

So the multi-class transfer problem can be solved by optimizing the following objective function:

$$\begin{aligned} \textbf{min} \quad & \frac{1}{2}\sum_{n=1}^{N}\|w_n - \gamma_n w'_n\|^2 + \frac{1}{2}\left\|w_{N+1} - \sum_{k=1}^{N} w'_k\beta_k\right\|^2 \\ & \frac{C}{2}\sum_{n=1}^{N+1}\sum_{i=1}^{l} e_{i,n}^2 \\ \textbf{s.t.} \quad & e_{i,n} = Y_{i,n} - \phi(x_i)w_n - b_n \end{aligned} \tag{2}$$

The closed-form of the optimal solution to Eq. (2) is:

$$\begin{aligned} w_n &= \gamma_n w'_n + \sum_{i}^{l}\alpha_{in}\phi(x_i) \qquad n = 1, ..., N \\ w_{N+1} &= \sum_{k}^{N}\beta_k w'_k + \sum_{i}^{l}\alpha_{i(N+1)}\phi(x_i) \end{aligned}$$

Here α_{ij} is the element (i,j) in $\boldsymbol{\alpha}$. The intuitive interpretation of the results above is that the hyperplane of the target problem is the linear combination of the prior knowledge (first part of the right side) and empirical knowledge from target task (second part of the right side).

Let ψ denotes the first term of left-hand side in Eq. (1) and let:

$$\begin{aligned} \psi\begin{bmatrix} \boldsymbol{\alpha'} \\ \boldsymbol{b'} \end{bmatrix} &= \begin{bmatrix} Y \\ 0 \end{bmatrix} \\ \psi\begin{bmatrix} \boldsymbol{\alpha''} \\ \boldsymbol{b''} \end{bmatrix} &= \begin{bmatrix} X(W')^T \\ 0 \end{bmatrix} \end{aligned} \tag{3}$$

We have:

$$\boldsymbol{\alpha} = \boldsymbol{\alpha'} - \begin{bmatrix} \boldsymbol{\alpha''}d_r & \boldsymbol{\alpha''\beta^T} \end{bmatrix} \tag{4}$$

From Eq. (4) we can see that, the solution of Eq. (1) is completed once $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_N]$ and $\boldsymbol{\beta}$ are set.

B. Optimize γ and β

introduce LOO error estimation. In this part, we introduce our method to estimate proper $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ that can prevent negative transfer. From above, we can see that the hyperplane for the target problem is determined by $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$. Negative transfer happens when the model aggressively leverage over irrelevant prior knowledge, i.e. set a large value to $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$. However, aggressive leverage over informative priors can improve the performance of the transfer model greatly. Inspired by some previous works [1] [3], we proposed our method that can minimize the affect of negative transfer from unrelated priors.

As we mentioned above, another important advantage of LS-SVM over the other model is that we can get unbiased LOO error in closed form [11]. The unbiased LOO estimation for sample x_i can be written as:

$$\hat{Y}_{i,n} = Y_{i,n} - \frac{\alpha_{in}}{\psi_{ii}^{-1}} \text{ for } n = 1, ..., N+1 \tag{5}$$

Here ψ^{-1} is the inverse of matrix ψ and ψ_{ii}^{-1} is its ith diagonal element.

Let us call ξ_i the empirical error of our multi-class prediction for example x_i, and ξ_i can be defined as [19]:

$$\xi_i(\gamma, \beta) = \max_{n\in\{1,...,N+1\}}\left[1 - \varepsilon_{ny_i} + \hat{Y}_{in}(\gamma, \beta) - \hat{Y}_{iy_i}(\gamma, \beta)\right] \tag{6}$$

Where $\varepsilon_{ny_i} = 1$ if $n = y_i$ and 0 otherwise. $\xi_i(\gamma, \beta) > 0$ if example x_i is misclassified. The intuition behind this loss function is to enforce the distance between the true class and other classes to be at least 1.

Then we define our objective function as:

$$\begin{aligned} \textbf{min} \quad & \frac{\lambda_1}{2}\sum_{n=1}^{N}\|\gamma_n\|^2 + \frac{\lambda_2}{2}\sum_{n=1}^{N}\|\beta_n\|^2 + \sum_{i=1}^{l}\xi_i \\ \textbf{s.t.} \quad & 1 - \varepsilon_{ny_i} + \hat{Y}_{in}(\gamma, \beta) - \hat{Y}_{iy_i}(\gamma, \beta) \leq \xi_i; \\ & \lambda_1, \lambda_2 \geq 0 \end{aligned} \tag{7}$$

Here λ_1 and λ_2 are two regularization parameters to prevent overfitting. From the objective function above we can see that, for certain λ_1 and λ_2, when the prior knowledge is unrelated and negative transfer happens, increasing $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ leads to larger punishment from both regularization and empirical error from target task. Decreasing the affect of prior knowledge reduces the loss of the objective function and eventually prevents negative transfer. Moreover, we also prove that this objective function can avoid negative transfer (for more details, see Theorem 1). On the other hand, if the prior knowledge is related, even though, increasing $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ leads to larger punishment, it also leads to smaller empirical error on the target problem. So the algorithm compromises between the prior and empirical knowledge. Besides, there are some other properties that make our method efficient.(see Section III-C)

By adding a dual set of variables, one for each constraint, we get the Lagrangian of the optimization problem:

$$\begin{aligned} \textbf{max} \quad & L(\gamma, \beta, \xi, \eta) = \\ & \frac{\lambda_1}{2}\sum_{n=1}^{N}\|\gamma_n\|^2 + \frac{\lambda_2}{2}\sum_{n=1}^{N}\|\beta_n\|^2 + \sum_{i=1}^{l}\xi_i \\ & + \sum_{i,n}\eta_{i,n}\left[1 - \varepsilon_{ny_i} + \hat{Y}_{in}(\gamma, \beta) - \hat{Y}_{iy_i}(\gamma, \beta) - \xi_i\right] \\ \textbf{s.t.} \quad & \forall i, n \quad \eta_{i,n} \geq 0 \end{aligned} \tag{8}$$

The problem of Eq. (8) is a non-differentiable strongly convex problem. The sub-gradient of it can be written as:

$$\Delta_\gamma = \begin{cases} \mathbf{0} & y_i = n \\ \left[0, .., \frac{\alpha''_{in}}{\psi_{ii}^{-1}}, .., -\frac{\alpha''_{iy_i}}{\psi_{ii}^{-1}}, .., 0\right] & y_i, n = 1, ...N \\ \left[0, .., \frac{\alpha''_{in}}{\psi_{ii}^{-1}}, .., 0\right] & y_i = N+1; n = 1, ...N \\ \left[0, .., -\frac{\alpha''_{iy_i}}{\psi_{ii}^{-1}}, .., 0\right] & \text{otherwise} \end{cases}$$

$$\Delta_\beta = \begin{cases} -\sum \alpha''_{ik}\beta_k & y_i = N+1; n = 1, ...N \\ \sum \alpha''_{ik}\beta_k & y_i = 1, ..N; n = N+1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

To obtain the optimal values for the problem above, we introduce our method using sub-gradient descent [20] and summarize it in Alg. 1.

C. Analysis

In this part, we mainly discuss our method in two aspects: convergence analysis and mathermatical proof of preventing negative transfer.

Algorithm 1 γ optimization

Input: $\psi, \alpha', \alpha'', T, \psi,$
Output: $\gamma = \{\gamma^1, ..., \gamma^n\}, \beta$
$\beta^0 \leftarrow 0, \gamma^0 \leftarrow 1$
for $t = 1$ to T **do**
  $\hat{Y} \leftarrow Y - (\psi \circ I)^{-1}\left(\alpha' - \left[\begin{array}{cc} \alpha'' d_\gamma & \alpha''\beta^T \end{array}\right]\right)$
  $\Delta_\gamma = 0, \Delta_\beta = 0$
  for $i = 1$ to l **do**
    $\Delta_\gamma \leftarrow \Delta_\gamma + \lambda_1\gamma$
    $\Delta_\beta \leftarrow \Delta_\beta + \lambda_2\beta$
    for $r = 1$ to $N+1$ **do**
      $l_{ir} = 1 - \varepsilon_{y_i r} + \hat{Y}_{ir} - \hat{Y}_{iy_i}$
      if $l_{ir} > 0$ **then**
        if $y_i, r \in \{1, ..., N\}$ **then**
          $\Delta_\gamma^{y_i} \leftarrow \Delta_\gamma^{y_i} - \frac{\alpha''_{iy_i}}{\psi_{ii}^{-1}}$
          $\Delta_\gamma^{r} \leftarrow \Delta_\gamma^{r} + \frac{\alpha''_{ir}}{\psi_{ii}^{-1}}$
        else if $y_i = N+1$ **then**
          $\Delta_\beta \leftarrow \Delta_\beta - \frac{\alpha''_{i}}{\psi_{ii}^{-1}}$
          $\Delta_\gamma^{r} \leftarrow \Delta_\gamma^{r} + \frac{\alpha''_{ir}}{\psi_{ii}^{-1}}$
        else
          $\Delta_\gamma^{y_i} \leftarrow \Delta_\gamma^{y_i} - \frac{\alpha''_{iy_i}}{\psi_{ii}^{-1}}$
          $\Delta_\beta \leftarrow \Delta_\beta + \frac{\alpha''_{i}}{\psi_{ii}^{-1}}$
        end if
      end if
    end for
  end for
  $\beta^t \leftarrow \beta^{(t-1)} - \frac{\Delta_\beta}{l \times t}$
  $\gamma^t \leftarrow \gamma^{(t-1)} - \frac{\Delta_\gamma}{l \times t}$
end for

Convergence analysis The primal problem (7) becomes the strongly convex problem by adding the L2 regularization terms. Optimizing the strongly convex problem can lead to the following error bound:

Let $\mu_1, ..., \mu_t$ be a sequence corresponding to $\mu_t = (\sqrt{\lambda_1}\gamma^t, \sqrt{\lambda_2}\beta^t)$. Problem (7) can be rewritten as:

$$J(\mu) = \frac{1}{2}\|\mu\|^2 + \sum_{i=1}^{l}\xi_i(\mu)$$

Let Δ_t be the sub-gradient for $J(\mu_t)$ and $\mu^* = (\sqrt{\lambda_1}\gamma^*, \sqrt{\lambda_2}\beta^*)$ be the optimal solution for it. Assume that $\|\Delta_t\| \leq G$. According to Lemma 1 in [21], we have:

$$J(\mu_t) - J(\mu^*) \leq \frac{G^2}{2t}(1 + \ln(t)) \tag{9}$$

This means our method converges at the rate of $O(\frac{\log(t)}{t})$.

Superior bound analysis

Theorem 1: Assume that $\bar{\xi}_i$ is the multi-class loss of example x_i when $\gamma = \beta = \mathbf{0}$. Let γ^*, β^* be the optimal solution for Eq. (8) and ξ_i^* be the multi-class loss with respective to example x_i. Then for every example $x_i \in \mathcal{X}$, we have:

$$\sum_i \xi_i \leq \sum_i \bar{\xi}_i$$

Proof: For simplification, let $\delta_i = 1$ if $i = N+1$ and 0 otherwise, and $\theta_{ij} = \alpha''_{ij}(1-\delta_j)/\psi_{ii}^{-1}$. Eq. (6) can be written as:

$$\xi_i(\gamma,\beta) = \max_n \left\{ \varepsilon_{ny_i} - 1 + \frac{(\alpha'_{iy_i} - \alpha'_{in})}{\psi_{ii}^{-1}} + \theta_{in}\gamma_n - \theta_{iy_i}\gamma_{y_i} + (\delta_n - \delta_{y_i}) \sum_k \frac{\alpha''_{ik}\beta_k}{\psi_{ii}^{-1}} \right\} \tag{10}$$

When $\gamma = \beta = \mathbf{0}$, from Eq. (10) we can get:

$$\bar{\xi}_i = \max_n \left[\varepsilon_{ny_i} - 1 + \frac{(\alpha'_{iy_i} - \alpha'_{in})}{\psi_{ii}^{-1}} \right]$$

To obtain the optimal value of γ and β, we have to seek the saddle point of the Lagrangian problem in (8) by finding the minimum for the prime variables $\{\gamma, \beta, \xi\}$ and the maximum for the dual variables η. To find the minimum of the primal problem, we require:

$$\frac{\partial L}{\partial \xi_i} = 1 - \sum_n \eta_{in} = 0 \rightarrow \sum_n \eta_{in} = 1$$

Similarly, for γ and β, we require:

$$\begin{aligned} \frac{\partial L}{\partial \gamma_n} &= \lambda_1\gamma_n + \sum_i \eta_{in}\theta_{in} - \sum_{i,n=y_i} \left(\sum_q \eta_{iq} \right) \theta_{in}\gamma_n \\ &=_1 \lambda_1\gamma_n + \sum_i \eta_{in}\theta_{in} - \sum_i \varepsilon_{ny_i}\theta_{in} = 0 \\ &\Rightarrow \gamma_n^* = \frac{1}{\lambda_1} \sum_i (\varepsilon_{ny_i} - \eta_{in})\,\theta_{in} \end{aligned} \tag{11}$$

In $=_1$ we use the facts that $\sum_n \eta_{in} = 1$ and use ε_{ny_i} to replace it.

$$\begin{aligned} \frac{\partial L}{\partial \beta_n} &= \lambda_2\beta_n + \left[\sum_{i,n} \frac{\eta_{in}\alpha''_{in}}{\psi_{ii}^{-1}} (\delta_n - \delta_{y_i}) \right] = 0 \\ &\Rightarrow \beta_n^* = \frac{1}{\lambda_2} \sum_{i,n} \frac{\eta_{in}\alpha''_{in}}{\psi_{ii}^{-1}} (\delta_{y_i} - \delta_n) \end{aligned} \tag{12}$$

As the strong duality holds,the primal and dual objectives coincide. Plug Eq (11) and (12) into Eq. (8), we have:

$$\sum_{i,n} \eta_{in} \left[1 - \varepsilon_{ny_i} + \hat{Y}_{in}(\gamma^*, \beta^*) - \hat{Y}_{iy_i}(\gamma^*, \beta^*) - \xi_i^* \right] = 0$$

Expand the equation above, we have:

$$\begin{aligned} &\sum_{i,n} \eta_{in} \left[\varepsilon_{n,y_i} - 1 + \frac{(\alpha'_{iy_i} - \alpha'_{in})}{\psi_{ii}^{-1}} - \xi_i \right] \\ &= \lambda_1 \sum_r \|\gamma_r^*\|^2 + \lambda_2 \sum_r \|\beta_r^*\|^2 \geq 0 \end{aligned}$$

Rearranging the above, we obtain:

$$\begin{aligned} &\sum_{i,n} \eta_{in} \left[\varepsilon_{n,y_i} - 1 + \frac{(\alpha'_{iy_i} - \alpha'_{in})}{\psi_{ii}^{-1}} \right] \\ &\geq \sum_{i,n} \eta_{in}\xi_i = \sum_i \xi_i \end{aligned} \tag{13}$$

The left-hand side of Inequation (13) can be bounded by:

$$\begin{aligned} &\sum_{i,n} \eta_{in} \left[\varepsilon_{ny_i} - 1 + \frac{(\alpha'_{iy_i} - \alpha'_{in})}{\psi_{ii}^{-1}} \right] \\ &\leq \sum_i \left(\sum_n \eta_{in} \max_r \left\{ \varepsilon_{ry_i} - 1 + \frac{(\alpha'_{iy_i} - \alpha'_{ir})}{\psi_{ii}^{-1}} \right\} \right) \\ &= \sum_i \left(\sum_n \eta_{in}\bar{\xi}_i \right) = \sum_i \bar{\xi}_i \end{aligned} \tag{14}$$

■

When setting $\gamma = \beta = \mathbf{0}$, we don't utilize any knowledge from previous task (see Eq. (2)). From Theorem 1 we can conclude our method can always outperform the method learning directly.

discuss λ

IV. Experiment

In this section, we show empirical results of our algorithm on different transferring situations on two datasets: AwA[1] [22] and Caltech-256[2] [23]. We design the following 3 sets of experiments: learning from informative prior (positive transfer), unrelated prior (negative transfer) and mixed prior (negative transfer), to show the effectiveness of our algorithm.

A. Dataset

Caltch-256 contains 30607 images from 256 categories. We select the following 10 categories: *bat, bear, dolphin, giraffe, gorilla, horse, leopard, raccoon, skunk, zebra* as our dataset.

AwA dataset consists of 50 animal categories. Its source images is not publicly accessible and we can only access the six pre-extracted feature representations for each image. This property makes it natural as the unknown distribution source dataset to train the prior knowledge. We choose the identical 10 categories as those in Caltech-256 as the source dataset.

B. Baselines and algorithmic setup

We compare our algorithm with two kinds of baselines. The first one is methods without leveraging any prior knowledge (no transfer baselines). The second consists of some methods with transfer techniques. Here are the no transfer baselines.

0+T(arget): LS-SVM trained only on target data. This baseline can be the indicator as the best performance in the bad oracle experiment.

S(ource)+T(arget): This baseline is only used in good oracle experiment. We combined the source and target data, assuming that we have fully access to all data, to train the LS-SVM. The result of this baseline might be considered as the best performance achieved in the experiment as well as an important reference for assessing the models with transfer learning methods.

[1]The features of AwA dataset is available from http://attributes.kyb.tuebingen.mpg.de/

[2]Images for Caltech-256 is available from http://www.vision.caltech.edu/Image_Datasets/Caltech256/

S(ource)+1: This method only train a new binary LS-SVM for the new category. For the rest of the classes, we use the predictions of the classifiers trained from source data directly. This is arguably the easiest way for transfer learning. In some of our experiments, it is a good indicator when negative transfer happens.

We select the following 3 methods as our transfer baselines. The general property of these 3 methods is that they all try to leverage multiple prior knowledge to benefit the transfer procedure.

MKTL [4]: This method uses the output of prior models as extra feature inputs, and automatically determine from which prior models to transfer and how much to transfer.

Multi-KT [1]: This method has similar idea with MKTL. It uses LOO error to determine how much to transfer from prior models and convert it into solving the convex optimization problem.

MULTIpLE [3]: The basic setting of this method is similar like ours. It is designed to balance the performance between learning the new category and preserving the model from prior knowledge.

For all the experiments in this section, we used kernel averaging [24], computing the average of RBF kernels over the available features on RBF hyperparameter $\{2^{-5}, 2^{-4}, ..., 2^{8}\}$. The penalty parameter C is tuned via cross-validation on $\{10^{-5}, 10^{-4}, ..., 10^{8}\}$ and the optimal value is reused for all the algorithms.

C. Positive Transfer

use 10 as example In MITL, the categories in both source and target task are drawn from the same distribution. Despite the affect of the new added category, transferring knowledge from prior knowledge can be considered as positive transfer. We perform the experiment under this scenario on both AwA and Caltech. On each dataset, we iteratively choose one category as the new category and run multiple times to get the average performance of each algorithm. The results of these two datasets are reported in Table II and Table III. From the result we can see that, in Caltech experiment, our algorithm consistently outperforms all the baselines even better than Batch.

To illustrate the detail performance of our algorithm, we select the experiment result on AwA dataset where horse is chosen as the new category for further explanation. In Figure 3(a), we show the average performances of different methods on different training size for 10 experiments. We can observe that, as the training size increases, our method can even outperforms the batch method. In Figure 33b we provide values of γ and β compared with the parameters of the runner-up transfer algorithm MULTIpLE. We can see that for transfer knowledge between identical categories, MULTIpLE fixes the transfer parameter (γ) to be 1 while our method sets greater weights for related prior knowledge. The same phenomenon is observed for the transfer parameter β. By exploiting the positive prior knowledge more aggressively, SMITLe is able to leverage the prior knowledge and can still achieve good performance when the train data is rare (5 examples per class).

TABLE II: Overall Caltech to Caltech with different size of training set in target problem. 30 examples are randomly chosen from each class to train the source classifier and 30 examples from each class are chosen for test.

size per category	5	10	15	20
No transfer	27.33	31.53	35.73	38.47
Source+1	43.33 ± 5.96	43.87 ± 6.14	44.33 ± 6.00	44.57 ± 5.99
MKTL	38.89 ± 8.54	43.27 ± 7.66	45.72 ± 6.41	47.44 ± 6.96
MULTIKT	37.96 ± 5.99	42.89 ± 5.77	45.96 ± 6.25	47.32 ± 5.73
MULTIpLE	42.63 ± 6.35	45.63 ± 5.99	47.81 ± 5.92	48.73 ± 6.01
Gama	**43.53 ± 5.53**	**46.45 ± 5.31**	**48.25 ± 5.74**	**49.15 ± 5.90**
Batch	43.77 ± 5.99	44.73 ± 6.01	46.67 ± 5.71	48.00 ± 5.35

TABLE III: Overall AwA to AwA with different size of training set in target problem. 50 examples are randomly chosen from each class to train the source classifier and 200 examples from each class are chosen for test.

size per category	5	10	15	20
No transfer	23.52	26.79	29.60	31.50
Source+1	39.00 ± 2.13	39.34 ± 1.95	39.62 ± 1.94	39.74 ± 2.02
MKTL	31.46 ± 2.51	34.76 ± 2.37	37.41 ± 2.10	38.81 ± 2.03
MULTIKT	29.86 ± 1.52	32.86 ± 1.31	35.22 ± 1.30	36.33 ± 1.26
MULTIpLE	37.80 ± 2.11	38.81 ± 2.06	39.80 ± 1.96	40.47 ± 1.96
Gama	**37.83 ± 2.38**	**39.31 ± 2.26**	**40.37 ± 2.18**	**41.09 ± 2.05**
Batch	39.62 ± 1.98	40.18 ± 2.05	40.67 ± 2.03	41.44 ± 1.93

D. from bad oracle

use 6 as example

E. mixed

V. Conclusion

Acknowledgment

References

[1] T. Tommasi, F. Orabona, and B. Caputo, "Learning categories from few examples with multi model knowledge transfer," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 5, pp. 928–941, 2014.

[2] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 4, pp. 594–611, 2006.

[3] I. Kuzborskij, F. Orabona, and B. Caputo, "From n to n+ 1: Multiclass transfer incremental learning," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3358–3365.

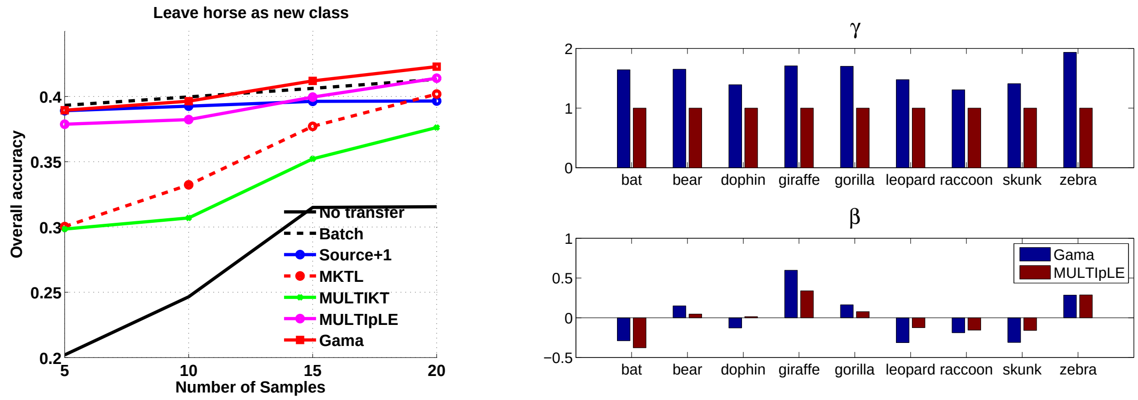
[4] L. Jie, T. Tommasi, and B. Caputo, "Multiclass transfer learning from unconstrained priors," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1863–1870.

[5] S. Thrun, "Is learning the n-th thing any easier than learning the first?" in *Advances in Neural Information Processing Systems*. The MIT Press, 1996, pp. 640–646.

[6] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14 – 23, 2015, 25th anniversary of Knowledge-Based Systems.

[7] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 188–197.

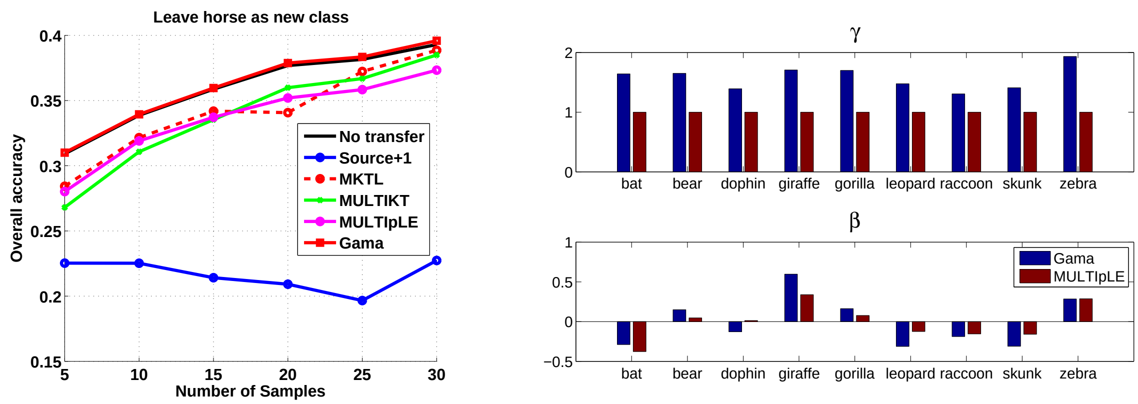
[8] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2252–2259.



(a) Overall accuracy comparision with different baselines.

(b) Comparision with MULTIpLE.

Fig. 3: Experiment results for 10 classes, AwA. Horse is used as the new category. (a)



(a) Overall accuracy comparision with different baselines.

(b) Comparision with MULTIpLE.

Fig. 4: Experiment results for 10 classes, AwA. Horse is used as the new category. (a)

TABLE IV: Overall AwA to Caltech

	5	10	15	20	25	30
No transfer	**30.99 ± 2.61**	33.97 ± 2.79	35.95 ± 1.94	37.78 ± 1.81	38.27 ± 1.94	39.39 ± 1.61
Source+1	17.89 ± 3.44	18.69 ± 2.66	18.79 ± 2.32	19.69 ± 2.16	19.39 ± 2.70	20.20 ± 1.94
MKTL	25.19 ± 4.14	30.14 ± 4.18	32.53 ± 4.00	34.30 ± 3.39	35.83 ± 3.19	36.66 ± 3.22
MULTIKT	27.60 ± 1.90	32.19 ± 2.88	34.51 ± 2.52	36.78 ± 1.68	37.79 ± 2.00	39.27 ± 2.00
MULTIpLE	29.79 ± 2.41	33.45 ± 2.20	35.49 ± 1.79	36.77 ± 1.48	37.43 ± 1.61	38.62 ± 1.61
Gama	30.93 ± 2.55	**34.13 ± 2.94**	**36.09 ± 2.09**	**38.01 ± 1.76**	**38.46 ± 1.95**	**39.59 ± 1.65**

TABLE V: AwA leave class 6 as new category. 2345 as bad class

	5	10	15	20	25	30
no transfer	23.45 ± 1.55	26.97 ± 2.08	29.64 ± 2.40	31.62 ± 2.38	32.59 ± 2.64	34.44 ± 2.40
source+1	17.35 ± 1.05	18.43 ± 1.57	17.66 ± 1.15	18.18 ± 1.47	18.59 ± 1.65	19.16 ± 1.44
MKTL	21.73 ± 2.45	25.19 ± 2.83	28.45 ± 2.13	31.46 ± 2.38	31.74 ± 4.62	32.44 ± 3.17
MultiKT	21.78 ± 2.08	25.60 ± 2.34	28.84 ± 2.68	31.29 ± 2.26	32.13 ± 2.56	33.78 ± 2.23
MULTIpLE	22.13 ± 1.83	26.19 ± 2.04	29.29 ± 2.39	31.52 ± 2.36	32.86 ± 2.37	34.37 ± 2.30
Gama	25.14 ± 1.31	27.89 ± 1.91	30.31 ± 2.47	32.14 ± 2.52	33.08 ± 2.70	34.78 ± 2.46

[9] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[10] I. Kuzborskij and F. Orabona, "Stability and hypothesis transfer learning," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 942–950.

[11] G. C. Cawley, "Leave-one-out cross-validation based model selection criteria for weighted ls-svms," in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*. IEEE, 2006, pp. 1661–1668.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[13] J. J. Lim, "Transfer learning by borrowing examples for multiclass object detection," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

[14] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France*, 2015, pp. 97–105.

[15] J. Davis and P. Domingos, "Deep transfer via second-order markov logic," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 217–224.

[16] X. Wang, T.-K. Huang, and J. Schneider, "Active transfer learning under model shift," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1305–1313.

[17] T. Zhou and D. Tao, "Multi-task copula by sparse graph regression," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 771–780.

[18] T. Tommasi, F. Orabona, and B. Caputo, "Safety in numbers: Learning categories from few examples with multi model knowledge transfer," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3081–3088.

[19] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *The Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2002.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[21] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[22] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 951–958.

[23] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[24] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 221–228.