BOOSTED UNSUPERVISED MULTI-SOURCE SELECTION FOR DOMAIN ADAPTATION

K. Vogt[a,*], A. Paul[b], J. Ostermann[a], F. Rottensteiner[b], C. Heipke[b]

[a] Institut für Informationsverarbeitung, Leibniz Universität Hannover, Germany
(vogt, ostermann)@tnt.uni-hannover.de,
[b] Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany
(paul, rottensteiner, heipke)@ipi.uni-hannover.de

KEY WORDS: Transfer Learning, Domain Adaptation, Negative Transfer, Source Selection, Machine Learning, Remote Sensing

ABSTRACT:

Supervised machine learning needs high quality, densely sampled and labelled training data. Transfer learning (TL) techniques have been devised to reduce this dependency by adapting classifiers trained on different, but related, (source) training data to new (target) data sets. A problem in TL is how to quantify the relatedness of a source quickly and robustly, because transferring knowledge from unrelated data can degrade the performance of a classifier. In this paper, we propose a method that can select a nearly optimal source from a large number of candidate sources. This operation depends only on the marginal probability distributions of the data, thus allowing the use of the often abundant unlabelled data. We extend this method to multi-source selection by optimizing a weighted combination of sources. The source weights are computed using a very fast boosting-like optimization scheme. The run-time complexity of our method scales linearly in regard to the number of candidate sources and the size of the training set and is thus applicable to very large data sets. We also propose a modification of an existing TL algorithm to handle multiple weighted training sets. Our method is evaluated on five survey regions. The experiments show that our source selection method is effective in discriminating between related and unrelated sources, almost always generating results within 3% in overall accuracy of a classifier based on fully labelled training data. We also show that using the selected source as training data for a TL method will additionally result in a performance improvement.

1. INTRODUCTION

Supervised classification plays an important role for extracting semantic information from remote sensing imagery. From statistical considerations it can be expected that the estimation of any complex model with high accuracy will require large amounts of training data. While unlabelled data are abundant and are already used successfully in unsupervised and semi-supervised learning methods, they cannot completely replace the dependence on labelled data. The acquisition of high-quality, densely sampled and representative labelled samples, on the other hand, is expensive and a time consuming task. *Transfer Learning* (TL) is a paradigm that strives to vastly reduce the amount of required training data by utilizing knowledge from related learning tasks (Thrun and Pratt, 1998; Pan and Yang, 2010). In particular, the aim of TL is to adapt a classifier trained on data from a *source domain* to a *target domain*. The only assumption to be made is that these domains are different but related. We are interested in one specific setting of TL called domain adaptation (DA). DA methods assume the source and target domains to differ only by the marginal distributions of the features and the posterior class distributions (Bruzzone and Marconcini, 2009). The performance of DA depends on how the source is related to the target (Eaton et al., 2008). From that point of view, DA can be divided into two steps: find the most similar sources and transfer knowledge from these sources to the target. In this context, the major challenge in source selection is how to measure the similarity. This is important to avoid *negative transfer*, i.e. a reduction in accuracy compared to not transferring any knowledge at all (Pan and Yang, 2010).

In this paper we will address the problems of searching for similar sources, also known as *source selection* and of integrating the results into DA. As unlabelled data are abundant, our proposed method is only based on similarity measurement between the marginal distributions of source and target domains. Given a target domain and a list of candidate source domains, we assign weights to these sources on the basis of the *Maximum Mean Discrepancy* metric to the target. We apply *multi-source selection* by transferring knowledge from multiple weighted source domains simultaneously. Additionally, we adapt our approach for DA presented in (Paul et al., 2016) so that it can benefit from multi-source selection. We evaluate our method on the *Vaihingen* and *Potsdam* datasets from the ISPRS 2D semantic labelling challenge (Wegner et al., 2016) and on a third, even more challenging, dataset based on aerial imagery of three German cities.

2. RELATED WORK

In our work we use notation according to Pan and Yang (2010). A domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$ with $X \in \mathcal{X}$. A task for a given domain is defined as $\mathcal{T} = \{\mathcal{C}, h(\cdot)\}$, consisting of label space $\mathcal{C}$ and a predictive function $h(\cdot)$. The predictive function can be learned from the training data $\{\mathbf{x}_r, C_r\}$, where $\mathbf{x}_r \in X$ and $C_r \in \mathcal{C}$. We consider a target T, for which we want to learn a predictive function $h(\mathbf{x})$, and source S, from which some knowledge can be transferred. Both T and S are fully described by their domains and their tasks. In our work we consider at least one source domain $\mathcal{D}_S$ and only one target domain $\mathcal{D}_T$. There are different settings of TL. Our focus is on DA, which is a special sub-category of the transductive TL setting (Pan and Yang, 2010). There are slightly different definitions of the DA problem (Paul et al., 2016). We follow the definition of Bruzzone and Marconcini (2009) according to which different domains only

[*] Corresponding author

doi:10.5194/isprs-annals-IV-1-W1-229-2017

differ by the marginal distributions of the features and the posterior class distributions, i.e. we assume $P(X_S) \neq P(X_T)$ and $P(C_S|X_S) \neq P(C_T|X_T)$. From that point of view, DA corresponds to a problem where the source and target domain data are different, e.g. due to different lighting conditions or seasonal effects; the domains must be related, i.e. these differences must not be so large that transfer becomes impossible. In this scenario, finding a solution to the DA problem would allow to transfer a classifier trained on one set of images where training data are available ($\mathcal{D}_S$) to other images ($\mathcal{D}_T$) without having to provide additional training data in $\mathcal{D}_T$. This is different from the problem that the training set is non-representative, e.g. due to class imbalance. Such algorithms are known as *sample selection bias* or *covariate shift* correcting methods, as in (Zadrozny, 2004; Sugiyama et al., 2007). Zhang et al. (2010) adapted the classifier to the distribution of the target data by weighing training samples with a probability ratio of data from the source and target domains. However, this approach only deals with binary problems and other applications than image classification.

Pan and Yang (2010) categorize DA in two groups according to what is actually transferred. Methods of the first group using *feature representation transfer* assume that the differences between domains can be mitigated by projecting both domains into a shared feature space in which the differences between the marginal feature distributions are minimized, e.g. by using feature selection (Gopalan et al., 2011) or feature extraction (Matasci et al., 2015). Some of the methods in this category are driven by a graph matching procedure to find correspondences between domains (Tuia et al., 2013; Banerjee et al., 2015). These methods need to contain the correct matching sequence among the possible matches or labelled samples across domains to perform well. Cheng and Pan (2014) propose a semisupervised method for DA that uses linear transformations for feature representation transfer. However, this method also requires training data from the target domain. Methods that assume that differences can be found in the marginal distributions mostly fall into the second group of DA algorithms, based on *instance transfer*. They try to directly re-use training samples from the source domain, successively replacing them by samples from the target domain that receive their class labels (*semi-labels*) from the current state of the classifier.

Methods for instance transfer have been used in the classification of remotely sensed data, e.g. in (Acharya et al., 2011). Acharya et al. (2011) train the classifier on the basis of the source domain and combine the result with the results of several clustering algorithms to obtain improved posterior probabilities for the target domain data. The approach is based on the assumption that the data points of a cluster in feature space probably belong to the same class. Bruzzone and Marconcini (2009) present a method for DA based on instance transfer for Support Vector Machines (SVM). In (Paul et al., 2016), we adopted that principle for DA based on logistic regression, thus a simpler classifier which, nevertheless, may have a lower computational complexity in training, not least because it can be applied to multiclass problems in a straight-forward way. Durbha et al. (2011) show that methods of TL for classification of remotely sensed images can produce better results than a modification of the SVM. A DA method using logistic regression in a semi-supervised setting combined with clustering of unlabelled data has been presented in (Amini and Gallinari, 2002). Training is based on expectation maximisation (EM), and the semi-labels of the unlabelled data are determined according to the cluster membership of EM. In contrast to our DA technique, that method assumes the labelled and the unlabelled data to follow the same distribution. Our previous method was shown to achieve a positive transfer for many image pairs of similar domains, but there were problematic cases with negative transfer (Paul et al., 2016). The major problem was that it could not detect cases in which the general assumption of TL, namely that the domains have to be related, was violated.

The detection of negative transfer is of vital importance for TL. In (Bruzzone and Marconcini, 2010) a circular validation scheme was proposed to detect negative transfer after adapting the classifier. An alternative approach would try to detect a relevant source prior to applying TL, which is known as *source selection*, which, of course, requires the availability of multiple source domains. Most work in this area uses a distance measure between the marginal distributions to measure the similarity between domains. Such distribution distances are well known in statistics, where the problem is mostly solved for 1D feature spaces. Most research has therefore focussed on extending these metrics to multivariate data or to non-parametric models. Examples for such measures are the *Kullback-Leibler Divergence* (Sugiyama et al., 2007), the *Total-Variation Distance* (Sriperumbudur et al., 2012) and its approximations, the *Maximum-Mean-Discrepancy* (Gretton et al., 2012; Chattopadhyay et al., 2012; Matasci et al., 2015) and *$\mathcal{A}$-Distance* (Ben-David et al., 2007). These approaches are kernel-based and usually scale well to high-dimensional data, but they may be computationally expensive. Therefore, another focus of research has been on reducing computational requirements and an improved regularization by careful kernel tuning (Zaremba et al., 2013; Sriperumbudur et al., 2009). Chattopadhyay et al. (2012) proposed a multi-source DA algorithm for the detection of muscle fatigue from *surface electromyography* (SEMG) data. The data show a high variability between individual subjects, therefore not all subject data should be considered when learning an individualized fatigue detector for a new subject. A synthesized source is generated as a weighted combination of all candidate sources using a MMD-based domain distance. The method has cubic complexity in the number of candidate sources, which may make it slow for cases with many available sources.

In this paper, we present two methods for source selection based on two different distance metrics for domains. It is inspired by (Chattopadhyay et al., 2012), but we use an approximate optimization with linear run-time complexity and propose a method for tuning the kernel hyperparameter automatically. The methods deliver a synthetic source as a weighted combination of similar sources, designed to avoid negative transfer. Furthermore, we expand the algorithm in (Paul et al., 2016) so that it can deal with multiple sources. By selecting suitable source domains, it should be possible to achieve a positive transfer for most target domains.

3. DOMAIN ADAPTATION

We start this section with a short description of our previous work (Paul et al., 2016) before presenting improvements in section 3.2.

3.1 DA approach

We use multiclass *logistic regression* (LR) as our base classifier. LR directly models the posterior probability $P(C\,|\,\mathbf{x})$ of the class labels C given the data $\mathbf{x}$. We transform features into a higher-dimensional space $\Phi(\mathbf{x})$ in order to achieve non-linear decision boundaries. In the multiclass case, the model of the posterior is based on the softmax function (Bishop, 2006):

$$p\left(C = C^k|\mathbf{x}\right) = \frac{exp\left(\mathbf{w}_k^T \cdot \Phi(\mathbf{x})\right)}{\sum\limits_j exp\left(\mathbf{w}_j^T \cdot \Phi(\mathbf{x})\right)}, \tag{1}$$

where $\mathbf{w}_k$ is a parameter vector for a particular class C^k, $k \in K$, to be determined in the training process. For that purpose, a *training data set*, denoted as $\overline{TD}$, is assumed to be available. Initially, it contains only training samples from the source domain, each consisting of a feature vector $\mathbf{x}_n$, its class label C_n and a weight g_n. In the initial training, we use $g_n = 1, \forall n \in \{1, .., N\}$, but in the DA process, the samples will receive individual weights indicating the algorithm's confidence in the labels. In training, the optimal values of $\mathbf{w}$ given $\overline{TD}$ are determined by optimizing the posterior (Vishwanathan et al., 2006):

$$p\left(\mathbf{w}|\overline{TD}\right) \propto p\left(\mathbf{w}\right) \cdot \prod_{nk} p\left(C_n = C^k|\mathbf{x}_n, \mathbf{w}\right)^{g_n \cdot q_{nk}}, \qquad (2)$$

where q_{nk} is 1 if $C_n = C^k$ and 0 otherwise, $p\left(C = C^k|\mathbf{x}_n, \mathbf{w}\right)$ is defined in Eq. (1) and $p(\mathbf{w})$ is a Gaussian prior with mean $\bar{\mathbf{w}}$ and standard deviation σ. Compared to standard multiclass LR, the only difference is the use of the weights g_n (Paul et al., 2016). We use the Newton-Raphson method for finding the optimal parameters $\mathbf{w}$ by minimizing $-log(p\left(\mathbf{w}|\overline{TD}\right))$ (Bishop, 2006).

Our aim is to transfer the classifier trained on labelled source domain data to the target domain in an iterative procedure. Our initial classifier is trained on training set $\overline{TD}^0$ containing only source data. In each further iteration i of DA a predefined number ρ_E of source samples is removed from and a number ρ_A of semi-labelled target samples is included into the current training data set $\overline{TD}^i$. Thus, in iteration i, the current training data set $\overline{TD}^i$ consists of a mixture of N_S^i source samples and N_T^i target samples: $\overline{TD}^i = \{(\mathbf{x}_{S,r}; C_{S,r}; g_{S,r})\}_{r=1}^{N_S^i} \cup \{(\mathbf{x}_{T,l}; \widetilde{C}_{T,l}; g_{T,l})\}_{l=1}^{N_T^i}$. The symbol $\widetilde{C}_{T,l}$ denotes the *semi-labels* of the target samples, which are determined by applying a criterion based on a knn analysis. If the most frequent class label among the k nearest neighbours of an unlabelled sample is consistent with the predicted label according to a current state of the LR classifier, it is considered a candidate for inclusion into $\overline{TD}^i$. The ρ_A candidate samples having the shortest average distance to their k nearest neighbours will be added to $\overline{TD}^i$. We first remove source samples that are most distant from the decision boundary starting with the samples showing inconsistent class labels and continuing with samples with consistent labels.

At each iteration i, we have to define sample weights $g^i_{\overline{TD}} \in [0, 1]$ for all training samples in $\overline{TD}^i$, where $g^i_{\overline{TD}} = \{\{g^i_{S,r}\}_{r=1}^{N_S^i} \cup \{g^i_{T,l}\}_{l=1}^{N_T^i}\}$. For simplicity we refer to the weight of a sample as $g^i_{\overline{TD},n}$, $n \in \{1, .., N^i\}$ with $N^i = |\overline{TD}^i|$ if it does not matter whether the sample is originally from the source or from the target domain. The weight indicates the algorithm's trust in the correctness of the label of a training sample. The weight function used for determining $g^i_{\overline{TD},n}$ depends on the distance to the decision boundary: the higher that distance, the higher is the weight; a parameter h models the rate of increase of the weight with the distance (Paul et al., 2016). Having defined the current training data set $\overline{TD}^i$ and the weights, we retrain the LR classifier. This leads to an updated parameter vector $\mathbf{w}$ and a change in the decision boundary. This new state of the classifier is the basis for the definition of the training data set in the next iteration. Thus, we gradually adapt the classifier to the distribution of the target data.

3.2 Multi-source logistic regression DA

The method described in section 3.1 was adapted for using data from multiple source domains for training. To formally state our problem we define our current training data set set as follows:

$$\begin{aligned}\overline{TD}^i = &\bigcup_{s=1}^{|\mathbb{S}|} \{(\mathbf{x}_{S^s,r}; C_{S^s,r}; g_{S^s,r})\}_{r=1}^{N^i_{S^s}} \cup \\ &\bigcup_{t=1}^{|\mathbb{T}|} \{(\mathbf{x}_{T^t,l}; \widetilde{C}_{T^t,l}; g_{T^t,l})\}_{l=1}^{N^i_{T^t}}, \qquad (3)\end{aligned}$$

where $\mathbb{S}$ or $\mathbb{T}$ describe a set of source or target data sets, respectively, and $|\mathbb{T}| = 1$. Again, we refer to a particular sample in $\overline{TD}^i$ by its index n in $\overline{TD}^i$ if we are not interested in the domain it comes from. We use the defined training data set $\overline{TD}^i$ in our multi-source DA approach, but we use different definitions of the sample weights. One modification should decrease the weight of uncertain samples, the other one is required to deal with prior weights assigned to the individual source domains (cf. section 4).

3.2.1 Sample weights: The individual weights for the training samples should indicate the algorithm's trust in the correctness of the semi-labels, but our definition of weights in (Paul et al., 2016) only depended on the distance of a sample from the decision boundary. It may happen that a semi-label changes in the iterative DA process, which would imply that the semi-label is uncertain; semi-labels not having changed for many iterations should be trusted more than others. Here we introduce an adapted definition of the sample weights as shown in (Chang et al., 2002; Bruzzone and Marconcini, 2009) to model the trust in a sample in $\overline{TD}$ as a function of the number of iterations j for which its semi-label has remained unchanged (Fig. 1):

$$g_n^{*i} = min\left(g_n^i + \frac{(g^{max} - g_n^i)j^2}{(i^{max} - 1)^2}, g^{max}\right). \qquad (4)$$

In Eq. 4, g_n^i is the weight of sample n in the current adaptation step i according to original distance-based weight function (Paul et al., 2016), g_n^{*i} is the new weight of that sample, i^{max} defines the number of iterations for which the weight of a samples is allowed to increase quadratically with j, and g^{max} is the maximum possible sample weight. If no different source domains were considered, the weight for each training sample n in $\overline{TD}^i$ would be g_n^{*i}, i.e. the algorithm outlined in section 3.1 would be applied using the new definition of weights.

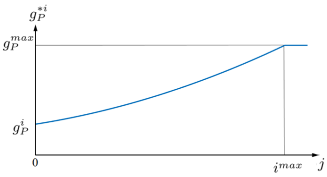


Figure 1. Sample weight function according to Eq. (4), assuming constant g_n^i during the adaptation.

3.2.2 Domain weights: In the context of multi-source selection, we introduce an individual domain weight π_{S^s} for every source domain s used in the DA process. The domain weights allow us to obtain a synthesized source $\bar{S}$ (section 4) from multiple sources, which is more similar to the target domain than any of the original ones. The domain weights remain constant during the adaptation procedure. For a sample n in the current training set $\overline{TD}^i$ taken from source domain s, the weight used in the DA process will be $g^i_{\overline{TD},n} = g_n^{*i} \cdot \pi_{S^s}$, where g_n^{*i} is defined in Eq. (4), whereas a sample n with a semi-label taken from the target domain will only have the weight g_n^{*i}. Thus, the weights of the source-domain samples are affected by the similarity of the corresponding domain to the target domain, placing a higher trust into samples that are from more similar source domains.

4. MULTI-SOURCE SELECTION

The goal of source selection is to avoid negative transfer by choosing a source which is, in some sense, similar to the target domain. Naturally, one should prefer sources which produce similar decision boundaries as the target task. The selection criterion should therefore be based on $\epsilon(h_s, \overline{TD}_T)$, i.e. the relative classification error ($\in [0,1]$) on the target data, given the predictive function h_s of the source task:

$$\bar{S} = \underset{S\in\mathbb{S}}{arg\,min}\,\epsilon(h_S, \overline{TD}_T)\,. \tag{5}$$

The main difficulty lies in the fact that estimating the classification error requires the class labels of the target domain to be known. Here, we introduce a theoretical framework and outline an algorithm that allows us to quickly find approximate solutions while requiring much less information. Our aim is the design of two complementary domain distance functions which we will call d_{SDA} and d_{UDA}. The function d_{SDA} measures a supervised domain distance in the sense that only class labels in the source domain need to be known, whereas d_{UDA} is completely unsupervised and does not require any class labels at all. We will refer to d_{DA} in places where either of these functions could be used. Equation (5) can then be approximated by $\bar{S} = arg\,min_{S\in\mathbb{S}}\, d_{\text{DA}}(\cdot)$. Our main contribution is the extension of these domain distances to the transfer from multiple sources while having a linear run-time complexity. In addition, we also show how the critical hyperparameters of our domain distances can be tuned automatically in an efficient manner.

4.1 Similarity of domains

We will derive our approximation of Eq. (5) in several steps. Using the results of Ben-David et al. (2007), an upper-bound for the classification error can be given as

$$\epsilon(h_S, \overline{TD}_T) \leq \epsilon(h_S, \overline{TD}_S) + d_{\mathcal{A}}(\overline{TD}_T, \overline{TD}_S) + \gamma\,. \tag{6}$$

The first term corresponds to the classification error on the source task. The term $d_{\mathcal{A}}(\overline{TD}_T, \overline{TD}_S)$, called $\mathcal{A}$-distance, describes a distance between the marginal feature distributions of the source and target domains. The third term, γ, encapsulates to which degree the DA assumption is held. The exact value can only be computed if class labels in the target task are available, but for related datasets this term should only take small positive values. Assuming that γ is unknown yet constant over the dataset, the upper bound gives us a definition for d_{SDA} according to $d_{\text{SDA}} = \epsilon(h_S, \overline{TD}_S) + d_{\mathcal{A}}(\overline{TD}_T, \overline{TD}_S)$. In the following we will define $d_{\mathcal{A}}$ and derive a more computationally friendly way to estimate this distribution distance. In (Ben-David et al., 2007), the $\mathcal{A}$-distance is defined as

$$d_{\mathcal{A}}(\overline{TD}_T, \overline{TD}_S) = 2(1 - 2\epsilon(h_{T\perp S}, \overline{TD}_{T\perp S}))\,. \tag{7}$$

The term $\epsilon(h_{T\perp S}, \overline{TD}_{T\perp S})$ describes the classification error for a classifier discriminating between feature vectors from the source and target domains. In that paper, only signed linear classifiers such as SVMs or logistic regression models were considered. Evaluation of the $\mathcal{A}$-distance involves the training of such a classifier for each candidate source, which has a high computational complexity. Furthermore, linear separability of the target and source domains is explicitely assumed. It is therefore desirable to find an approximation to the $\mathcal{A}$-distance that displays more favourable properties. The *Maximum Mean Discrepancy* (MMD) was independently proposed by Gretton et al. (2012) as a general distance function between probability distributions:

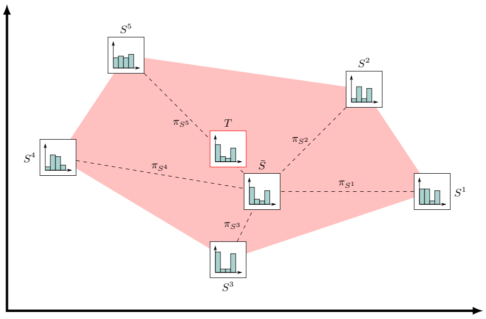


Figure 2. A synthesized source $\bar{S}$ is formed as the convex combination of candidate sources S^s.

$$\begin{aligned} d^2_{\text{MMD}}(\overline{TD}_T, \overline{TD}_S) &= E[(\phi(\mathbf{x}_T) - \phi(\mathbf{x}_S))^2] \\ &= E[k(\mathbf{x}_T, \mathbf{x}'_T)] - 2E[k(\mathbf{x}_T, \mathbf{x}_S)] + E[k(\mathbf{x}_S, \mathbf{x}'_S)]\,. \end{aligned} \tag{8}$$

The MMD computes the distance between the means of the probability distributions in a *Reproducing Hilbert Kernel Space* (RKHS). The RKHS is uniquely defined by either a feature space mapping $\phi(\mathbf{x})$ or its kernel function $k(\mathbf{x}, \mathbf{y})$. It was shown by Sriperumbudur et al. (2012) that the relation

$$d_{\mathcal{A}}(\overline{TD}_T, \overline{TD}_S) \approx 2 \cdot d_{\text{MMD}}(\overline{TD}_T, \overline{TD}_S) \tag{9}$$

holds for positive bounded kernels such as the Gaussian kernel:

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)\,. \tag{10}$$

Evaluation of the MMD can be done by replacing the expectations in Eq. (8) with their empirical estimates. A naive estimator would have a run-time complexity of $\mathcal{O}(N_T \cdot N_S)$, which becomes untenable for large training sets. A much faster linear-time estimator d_{LMMD} was proposed by Gretton et al. (2012). Assuming $M = N_T = N_S$, it can be stated as:

$$\begin{aligned} d^2_{\text{LMMD}}(\overline{TD}_T, \overline{TD}_S) = \frac{2}{M}\Bigg[\sum_{r=1}^{M/2} k(\mathbf{x}_{T,2r}, \mathbf{x}_{T,2r-1}) \\ - \sum_{r=1}^{M} k(\mathbf{x}_{T,r}, \mathbf{x}_{S,r}) + \sum_{r=1}^{M/2} k(\mathbf{x}_{S,2r}, \mathbf{x}_{S,2r-1})\Bigg]\,. \end{aligned} \tag{11}$$

Finally, replacing $d_{\mathcal{A}}$ by $2 \cdot d_{\text{LMMD}}$ using Eq. (9) leads to the definition of our supervised domain distance

$$d_{\text{SDA}}(\overline{TD}_T, \overline{TD}_S) = \epsilon(h_S, \overline{TD}_S) + 2d_{\text{LMMD}}(\overline{TD}_T, \overline{TD}_S) \tag{12}$$

Assuming the classification error to be approximately constant over all candidate sources, we obtain the unsupervised distance:

$$d_{\text{UDA}}(\overline{TD}_T, \overline{TD}_S) = 2d_{\text{LMMD}}(\overline{TD}_T, \overline{TD}_S)\,. \tag{13}$$

4.2 Convex combination of domains

In general we can assume that none of the candidate source domains $S \in \mathbb{S}$ is a perfect match for the target domain. Nonetheless, the target marginal distribution $p_T(\mathbf{x})$ might be much closer to the subspace spanned by the convex combination of the source marginal distributions (Fig. 2). Any point in this subspace represents a valid marginal distribution and can be parametrized as

$$p_{S_\pi}(\mathbf{x}) = \sum_{s=1}^{|\mathbb{S}|} \pi_{S^s} p_{S^s}(\mathbf{x}) \tag{14}$$

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.

doi:10.5194/isprs-annals-IV-1-W1-229-2017

given a source weight vector π satisfying the constraints $\pi_{S^s} \geq 0$, $\sum_s^{|\mathbb{S}|} \pi_{S^s} = 1$. By definition (14), the distribution $p_{S_\pi}(\mathbf{x})$ is a mixture of the source marginal distributions. The weighted training set $\overline{TD}_{S_\pi} = \bigcup_{s=1}^{|\mathbb{S}|} \{\mathbf{x}_{S^s,r}; C_{S^s,r}; \pi_{S^s}\}_{r=1}^{N_{S^s}}$ is therefore a representative sample from this distribution. The weights can be intuitively understood to mean that each sample from source $S^s \in \mathbb{S}$ is counted as π_{S^s} such samples. As an important intermediate result, we propose an extension of the linear-time MMD estimator (Eq. (11)) to a weighted union of source training sets:

$$\begin{aligned} d^2_{\text{LMMD}}(\overline{TD}_T, \overline{TD}_{S_\pi}) = \frac{2}{M} \Big[&\sum_{r=1}^{M/2} k(\mathbf{x}_{T,2r}, \mathbf{x}_{T,2r-1}) \\ &- \sum_{u=1}^{|\mathbb{S}|} \pi_{S^u} \sum_{r=1}^{M} k(\mathbf{x}_{T,r}, \mathbf{x}_{S^u,r}) \\ &+ \sum_{u=1}^{|\mathbb{S}|} \sum_{v=u+1}^{|\mathbb{S}|} \pi_{S^u} \pi_{S^v} \sum_{r=1}^{M} k(\mathbf{x}_{S^u,r}, \mathbf{x}_{S^v,r}) \qquad (15) \\ &+ \sum_{u=1}^{|\mathbb{S}|} \pi^2_{S^u} \sum_{r=1}^{M/2} k(\mathbf{x}_{S^u,2r}, \mathbf{x}_{S^u,2r-1}) \Big] . \end{aligned}$$

In the next section we will present a fast greedy optimization scheme that minimizes d_{DA} w.r.t. π.

4.3 Fast synthesis of source domains by boosting

Convex representation problems, like in Eq. (14), are related to *dictionary learning*. The *Iterative Nearest Neighbor* (INN) algorithm (Timofte and Van Gool, 2012) is a recent method that approximatively solves such problems in a greedy fashion. The solution at iteration L is given as

$$p_S^L(\mathbf{x}) = \sum_{l=1}^{L} w^l p_{S_l}(\mathbf{x}) \, , \tag{16}$$

where the iteration weights are computed as

$$w^l = \frac{\lambda}{(1+\lambda)^l} \tag{17}$$

for a fixed parameter λ. In order to find the next solution $p_S^{L+1}(\mathbf{x})$, we select a source which minimizes the representation error to the target domain according to our domain distance:

$$\begin{aligned} S_{L+1} = \underset{S \in \mathbb{S}}{arg\,min}\, d_{\text{DA}}\Big(&\overline{TD}_T, \{\mathbf{x}_{S,r}; C_{S,r}; w^{L+1}\}_{r=1}^{N_S} \\ &+ \bigcup_{l=1}^{L} \{\mathbf{x}_{S_l,r}; C_{S_l,r}; w^l\}_{r=1}^{N_{S_l}} \Big) . \qquad (18) \end{aligned}$$

The same source may be chosen multiple times at different iterations. The source weights can be derived from the iteration weights as follows:

$$\pi_{S^s} = \sum_{l=1}^{L} w^l \cdot 1_{\{S_l = S^s\}} \, . \tag{19}$$

Originally, the INN algorithm was designed to work on vectors in euclidean spaces. When interpreted in the space of probability distributions, the procedure has strong parallels to a non-adaptive variant of the boosting paradigm, whose most well known implementation is AdaBoost (Schapire and Singer, 1999). Similar to boosting, the synthesized source S_π is a weighted combination of weaker approximations. Also, the update step in Eq. (18) has the effect to steer the optimization successively to priorize parts of the distribution which are not yet well represented while also attenuating overrepresented parts.

Algorithm 1 Kernel Bandwidth Estimation

```
$\varphi \leftarrow 1.61803398875$
$(L, R) \leftarrow (0, \pi/2)$
$(A, B) \leftarrow (R - (R - L)/\varphi, L + (R - L)/\varphi)$
for $i = 1..\text{MaxIter}$ do
    $f_A \leftarrow d^2_{\text{LMMD}}(\overline{TD}_T, \overline{TD}_S)$ with $\sigma = \tan(A)$
    $f_B \leftarrow d^2_{\text{LMMD}}(\overline{TD}_T, \overline{TD}_S)$ with $\sigma = \tan(B)$
    if $f_A < 0$ then
        $R \leftarrow A$
    else if $f_B \leq f_A$ then
        $R \leftarrow B$
    else
        $L \leftarrow A$
    end if
    $(A, B) \leftarrow (R - (R - L)/\varphi, L + (R - L)/\varphi)$
end for
return $\sigma_{\text{max}} = \tan((L + R)/2)$
```

The sum $\sum_{l=1}^{\infty} w^l$ approaches 1 while the iteration weights w^l will become smaller and smaller. We can therefore stop the algorithm after L iterations such that $\sum_{l=1}^{L} w^l > \beta$ while avoiding large approximation errors. From Eq. (17) follows

$$L = \left\lceil -\frac{\log(1-\beta)}{\log(1+\lambda)} \right\rceil . \tag{20}$$

For typical parameter values $\beta = 0.9$, $\lambda = 0.5$ only $L = 6$ iterations are required. The run-time complexity of the entire multi-source selection algorithm using d_{UDA} can be given as $\mathcal{O}(L^3 |\mathbb{S}| M)$. The same result for our supervised variant d_{SDA} reads as $\mathcal{O}(L^3 |\mathbb{S}| M f(|\mathbb{S}| M))$ and additionally depends on the term $f(|\mathbb{S}| M)$, which describes the complexity of the classification algorithm used to estimate the first term in Eq. (12).

4.4 Kernel bandwidth estimation

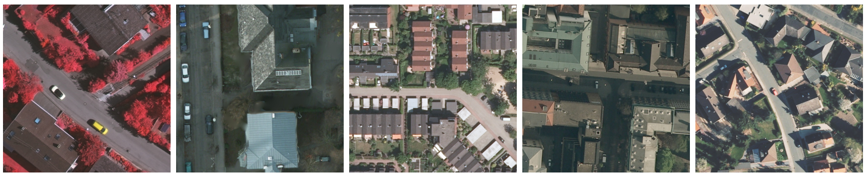
The Gaussian kernel has a single hyperparameter σ, its bandwidth. It was shown by Sriperumbudur et al. (2009) that the discriminative power of the MMD is maximized by maximizing d_{LMMD} w.r.t σ:

$$d^2_{\overline{\text{LMMD}}}(\overline{TD}_T, \overline{TD}_S) = \underset{\sigma \in (0,\infty)}{max}\, d^2_{\text{LMMD}}(\overline{TD}_T, \overline{TD}_S) \, . \tag{21}$$

Using the results by Shestopaloff (2010), we can show that this optimization problem has exactly one maximum at σ_{max} and at most one minimum at σ_{min}. Furthermore, if σ_{min} exists then $\sigma_{\text{max}} < \sigma_{\text{min}}$ holds. Finally, d_{LMMD} will tend towards zero for both $\sigma \to 0$ and $\sigma \to \infty$. We propose to solve this optimization problem using a *Golden-Section-Search* (GSS) (Press, 2007) (cf. Alg. 1). The GSS searches the maximum of a strictly unimodal function. We modified the GSS to handle cases where the MMD assumes negative values. This can occur for very similar domains due to small errors in the empirical estimates. The value range $(0, \infty)$ is mapped to $(0, \pi/2)$ using the $atan$ function. In our experiments the algorithm typically converged in less than 10 iterations.

4.5 Improving robustness by bootstrap aggregation

As all empirical estimators, our MMD estimator has a non-zero estimation variance which may result in a suboptimal solution π. We propose to reduce this variance by averaging π over multiple independent runs of our multi-source selection algorithm. Each run is performed on a bootstrap sample of the training sets $\overline{TD}_T$ and $\overline{TD}_S$. Bootstrap sampling describes a procedure where a new sample is generated via independent draws with replacement from an input sample. The statistical properties of bootstrap sampling are described in detail in (Hesterberg et al., 2003).



(a) Vaihingen (b) Potsdam (c) Buxtehude (d) Hannover (e) Nienburg

Figure 3. Example sections of orthophotos from all three datasets.

5. EXPERIMENTS

Our experimental evaluation is based on three datasets (cf. Fig. 3). Two of them are the Vaihingen and Potsdam datasets from the ISPRS 2D semantic labelling contest (Wegner et al., 2016). The Potsdam dataset was resampled from 5 cm to a ground sampling distance (GSD) of 8 cm to reduce the computational burden. Only patches for which a reference is available were used in our experiments. A third dataset, referred to as 3CITYDS, consists of three regions of German cities of varying size, degree of urbanization and architecture (Buxtehude, Hannover, Nienburg) [1]. This diversity produces much more pronounced differences between domains, thus exacerbating the effects of negative transfer when a wrong source is selected. Each region covers an area of $2 \times 2\,\text{km}^2$ but is evenly split up into 9 patches. The reference data for the 3CITYDS dataset was generated manually based on the image data. The properties of all datasets are further given in Table 1.

As it is the goal of these experiments to highlight the principles and strengths of boosted unsupervised multi-source selection for DA rather than to achieve optimal results, we only use two features: *normalized vegetation index* (NDVI) and the *normalized digital surface model* (nDSM). These features have been proven sufficient to discriminate all object classes in our datasets. All experiments are based on a pixel-wise classification of the input data into the three object classes *building*, *tree* and *ground*. The *ground* class includes all base-level surfaces and clutter objects.

A successful source selection should be able to find related sources and maximize the expected positive transfer. The evaluation will therefore consist of two parts. First we will analyze our proposed multi-source selection method. Our method is run for each patch to synthesize a source $\bar{S}$ using all remaining patches of the dataset as candidate sources. We examine several source selection strategies. Single source selection and supervised multi-source selection minimizes the domain distance d_{SDA} and utilizes labelled samples from the source domains. Unsupervised multi-source selection is based on the d_{UDA} domain distance. The multi-source methods only use the weighted combination of the three sources which received the highest source weights. We compare these methods to two simple reference methods: *Random Source* and *All Sources*. *Random Source* selects a single source randomly from all candidate sources. *All Sources*, on the other hand, assigns all candidate sources uniform source weights. In the first set of experiments, we are mainly interested in the performance of the synthesized source on the target task, so that classification is performed using multi-class logistic regression without DA, but using the source weights π_{S^s} to weight the samples (cf. Sec. 3).

In our second experiment we will enable the DA extension for our classifier, applying it to a synthesized source $\bar{S}$ generated by our unsupervised multi-source selection algorihm using only the 1-3 sources featuring the largest source weights. We will show that $\bar{S}$ is generally a better starting point for DA than a random source.

Multi-source selection and DA are applied using pixels on a regular grid of size $10\,\text{px} - 30\,\text{px}$ to reduce spatial dependency; the grid size was adapted to the GSD and the patch size of the individual datasets. For the logistic regression classifier, we applied a polynomial expansion of degree 2. For the multi-source selection we selected about 30% of the pixels per patch for each bootstrap run. The parameters used for DA (Section 3) and multi-source selection (Section 4) are given in Tab. 2. The DA parameters were tuned empirically on the three datasets. The same parameter values were used for all datasets without further tuning. The multi-source selection parameters are non-critical and were set to achieve a good tradeoff between speed and performance. As multi-source selection has some random components, each experiment is repeated ten times and we report average quality indices.

Dataset	GSD	Channels	Patches	Features	Classes
Vaihingen	8 cm	RGIR	16	2	3
Potsdam	8 cm	RGBIR	23	2	3
3CITYDS	20 cm	RGBIR	27	2	3

Table 1. Dataset properties.

Multi Source Selection

GSS MaxIter	INN λ	INN β	Bootstrap Runs	Bootstrap Size
10	0.5	0.9	10	5000

Domain Adaptation

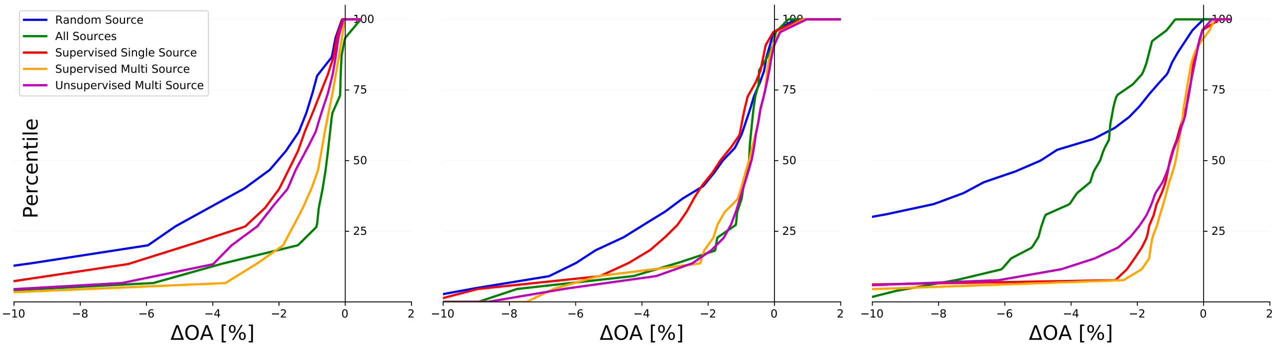
σ_0	σ_{DA}	ρ_E	ρ_A	KNN k	h	i^{max}	$g^{max}_{P,S}$	$g^{max}_{P,T}$
35	15	30	30	19	0.7	200	1.5	0.9

Table 2. Parameters.

5.1 Results and discussion

Figure 4 shows the evaluation of source selection without using DA. We present percentile plots and the average performance (Mean, Stdev) for each dataset separately. The percentile plots show the cumulative distribution of ΔOA over all patches in a dataset, where ΔOA is the difference in overall accuracy (OA) on the target task when learned on a synthesized source compared to training the classifier on a labelled target training set. The ΔOA directly shows how much performance is lost by not having access to class labels in the target domain. On all datasets, our single source method outperforms random selection, while using multiple weighted sources outperforms single source selection. Supervised source selection is generally better than unsupervised source selection, but the difference appears to be small on the tested datasets. Furthermore, training the classifier on all source patches is a competetive strategy on the Vaihingen and Potsdam dataset. These datasets only cover a single survey region and exhibit low variability between patches. For the more interesting case that most candidate sources are expected to be only weakly related to the target, this strategy falls behind our multi-source selection methods, as seen on the 3CITYDS dataset. As expected, random selection performs particularly bad under these circumstances (mean ΔOA: -10.8%), but this is compensated to a level

[1] Source: Extract from the geospatial data of the Lower Saxony survey and cadastre administration, © 2013 LGLN



	Rand	All	S	SM	UM
Mean	−4.6	−2.0	−3.7	−2.0	−2.7
Stdev	6.4	4.3	6.3	4.0	4.1

(a) Vaihingen

	Rand	All	S	SM	UM
Mean	−2.8	−1.6	−2.4	−1.5	−1.4
Stdev	3.3	2.4	2.7	2.1	2.2

(b) Potsdam

	Rand	All	S	SM	UM
Mean	−10.8	−3.9	−3.2	−2.3	−2.7
Stdev	13.1	2.4	8.4	5.7	5.3

(c) 3CityDS

Figure 4. Source selection results. ΔOA: difference in OA compared to a classifier based on target training data. Example for interpretation (Vaihingen, *All Sources*): for 25% of the target patches the loss in OA is larger than 1% (ΔOA < -1%).

that is comparable to the other datasets by multi-source selection. A classifier trained on the source synthesized by our unsupervised multi-source method only loses less than 3% in classification performance over 83% of all patches from all datasets.

Figure 5 shows the DA results using a random source and the 1-3 best sources according to unsupervised multi-source selection. We also compared the OA on the target data with and without enabling the DA extension in logistic regression; we report ΔDA, i.e. the mean difference in OA due to enabling DA over all patches of a dataset. The test shows that the performance is improved when using multi-source selection compared to random selection. When only using the optimal source (*DA1*), DA on average achieves a negative transfer, but when 2 or 3 sources are combined, a positive transfer is achieved in all experiments (ΔDA>0). Whereas a *Wilcoxon signed-rank* test (Siegel, 1956) indicates a positive transfer when using 2 or more sources at a 95% confidence level, the actual size of the improvement (< 1%) is still disappointing. However, the results show that unsupervised multi-source selection does indeed improve the prospects of DA, although it does not currently incorporate prior knowledge about specific properties of the DA method.

6. CONCLUSION

In this work, we presented two domain distances measures based on the MMD. One of these distances requires labelled samples in the source domain, the other one operates fully unsupervised. We developed a multi-source selection method that synthesizes a related source as a weighted combination of a set of candidate sources, of which only a few may be related to the target. Our fastest method has a linear run-time complexity in regard to the number of candidate sources and the size of the training set and is thus applicable to very large datasets. We also expanded an existing DA method to cope with multple sources being assigned different weights. Our experiments show that selecting the best sources, the loss in classification performance when compared to a classifier trained on target domain samples could be reduced considerably, in particular in cases with a very heterogeneous appearance of objects. Additionally applying DA could achieve a small positive transfer when using the weighted combination of two or more sources selected by our unsupervised procedure.

In future work we want to improve our DA method and its interplay with source selection. The impact of richer feature spaces, feature selection and more complex class structures still needs to be evaluated. The experiments presented in Section 5 correspond to a scenario where labelled training data are abundant from earlier projects. In such a scenario, selecting a good source domain compensates for most of the loss due to not labelling training data in a new target domain, so that perhaps the additional impact of DA must be expected to be small. We envision another application scenario in which multi-source selection is applied to a set of images when no labelled data is available. The results of source selection could be used to determine an optimal subset of patches that should be manually labelled in order to optimize classification over all patches. Such a method could become a fast alternative to active learning approaches and could greatly reduce the costs of manual labelling. With a smaller pool of source domains, the impact of DA is expected to be larger than in our experiments.

ACKNOWLEDGEMENTS

This work was supported by the German Science Foundation (DFG) under grants OS 295/4-1 and HE 1822/30-1 and in the context of the research training group GRK2159 (i.c.sens). The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) (Cramer, 2010): http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html.

References

Acharya, A., Hruschka, E. R., Ghosh, J. and Acharyya, S., 2011. Transfer learning with cluster ensembles. In: *Proceedings of the ICML Workshop on Unsupervised and Transfer Learning*, pp. 123–132.

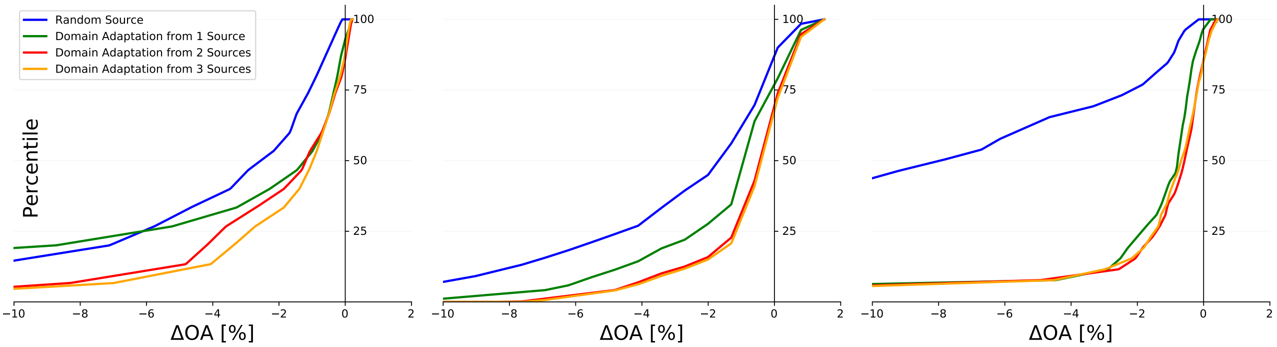
Amini, M.-R. and Gallinari, P., 2002. Semi-supervised logistic regression. In: *Proceedings of the 15th European Conference on Artificial Intelligence*, pp. 390–394.

Banerjee, B., Bovolo, F., Bhattacharya, A., Bruzzone, L., Chaudhuri, S. and Buddhiraju, K., 2015. A novel graph-matching-based approach for domain adaptation in classification of remote sensing image pair. *IEEE Transactions on Geoscience and Remote Sensing* 53(7), pp. 4045–4062.

Ben-David, S., Blitzer, J., Crammer, K. and Pereira, F., 2007. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems (NIPS)* 19, pp. 137–144.

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. 1st edn, Springer, New York (NY), USA.

Bruzzone, L. and Marconcini, M., 2009. Toward the automatic updating of land-cover maps by a domain-adaptation SVM classifier and a circular validation strategy. *IEEE Transactions on Geoscience and Remote Sensing* 47(4), pp. 1108–1122.



	Rand	DA1	DA2	DA3
Mean	−5.6	−7.2	−2.9	−2.5
Stdev	8.2	13.1	4.4	4.2
ΔDA	−1.0	−1.9	0.1	0.2

(a) Vaihingen

	Rand	DA1	DA2	DA3
Mean	−3.2	−1.8	−1.0	−0.9
Stdev	4.4	3.0	2.1	2.0
ΔDA	−0.5	−0.2	0.4	0.5

(b) Potsdam

	Rand	DA1	DA2	DA3
Mean	−12.2	−3.2	−2.6	−2.3
Stdev	12.3	7.8	6.7	5.6
ΔDA	−1.5	0.1	0.3	0.4

(c) 3CityDS

Figure 5. Domain adaptation results. ΔOA presents the difference in OA after source selection and DA when compared to a classifier based on target training data. ΔDA is the difference in OA between a classifier with and without DA being enabled.

Bruzzone, L. and Marconcini, M., 2010. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(5), pp. 770–787.

Chang, M.-W., Lin, C.-J. and Weng, R. C., 2002. Analysis of switching dynamics with competing support vector machines. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 3, pp. 2387–2392.

Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S. and Ye, J., 2012. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data* 6(4), pp. 18:1–18:26.

Cheng, L. and Pan, S. J., 2014. Semi-supervised domain adaptation on manifolds. *IEEE Transactions on Neural Networks and Learning Systems* 25(12), pp. 2240–2249.

Cramer, M., 2010. The DGPF test on digital aerial camera evaluation – overview and test design. *Photogrammetrie Fernerkundung Geoinformation* 2(2010), pp. 73–82.

Durbha, S., King, R. and Younan, N., 2011. Evaluating transfer learning approaches for image information mining applications. In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1457–1460.

Eaton, E., Lane, T. et al., 2008. Modeling transfer relationships between learning tasks for improved inductive transfer. In: *European Conference on Machine Learning (ECML)*, Springer, pp. 317–332.

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

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A., 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13(2012), pp. 723–773.

Hesterberg, T., Moore, D. S., Monaghan, S., Clipson, A. and Epstein, R., 2003. *The Practice of Business Statistics Companion Chapter 18: Bootstrap Methods and Permutation Tests*. WH Freeman & Co., New York (NY), USA.

Matasci, G., Volpi, M., Kanevski, M., Bruzzone, L. and Tuia, D., 2015. Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing* 53(7), pp. 3550–3564.

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

Paul, A., Rottensteiner, F. and Heipke, C., 2016. Iterative re-weighted instance transfer for domain adaptation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* III-3, pp. 339–346.

Press, W. H., 2007. *Numerical Recipes: The Art of Scientific Computing*. 3^{rd} edn, Cambridge university press, Cambridge, UK.

Schapire, R. E. and Singer, Y., 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37(3), pp. 297–336.

Shestopaloff, Y. K., 2010. *Sums of exponential functions and their new fundamental properties*. AKVY Press, Toronto, Canada.

Siegel, S., 1956. *Nonparametric statistics for the behavioral sciences*. McGraw-hill, New York, NY, USA.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Lanckriet, G. R. G. and Schölkopf, B., 2009. Kernel choice and classifiability for RKHS embeddings of probability distributions. In: *Advances in Neural Information Processing Systems (NIPS)*, Vol. 22, pp. 1750–1758.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., Lanckriet, G. R. et al., 2012. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics* 6, pp. 1550–1599.

Sugiyama, M., Krauledat, M. and Müller, K.-R., 2007. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research* 8, pp. 985–1005.

Thrun, S. and Pratt, L., 1998. Learning to learn: Introduction and overview. In: S. Thrun and L. Pratt (eds), *Learning to Learn*, Kluwer Academic Publishers, Boston, MA (USA), pp. 3–17.

Timofte, R. and Van Gool, L., 2012. Iterative nearest neighbors for classification and dimensionality reduction. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2456–2463.

Tuia, D., Munoz-Mari, J., Gomez-Chova, L. and Malo, J., 2013. Graph matching for adaptation in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing* 51(1), pp. 329–341.

Vishwanathan, S., Schraudolph, N., Schmidt, M. W. and Murphy, K. P., 2006. Accelerated training of conditional random fields with stochastic gradient methods. In: *Proc. 23^{rd} International Conference on Machine Learning (ICML)*, pp. 969–976.

Wegner, J. D., Rottensteiner, F., Gerke, M. and Sohn, G., 2016. The ISPRS 2D Labelling Challenge. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html. Accessed 11/03/2016.

Zadrozny, B., 2004. Learning and evaluating classifiers under sample selection bias. In: *Proceedings of the 21^{st} International Conference on Machine Learning*, pp. 114–121.

Zaremba, W., Gretton, A. and Blaschko, M., 2013. B-test: A non-parametric, low variance kernel two-sample test. In: *Advances in Neural Information Processing Systems (NIPS)*, Vol. 26, pp. 755–763.

Zhang, Y., Hu, X. and Fang, Y., 2010. Logistic regression for transductive transfer learning from multiple sources. In: L. Cao, J. Zhong and Y. Feng (eds), *Advanced Data Mining and Applications Part II*, Lecture Notes in Computer Science, Vol. 6441, Springer, pp. 175–182.