Unsupervised Source Selection for Domain Adaptation

Karsten Vogt, Andreas Paul, Jörn Ostermann, Franz Rottensteiner, and Christian Heipke

Abstract

The creation of training sets for supervised machine learning often incurs unsustainable manual costs. Transfer learning (TL) techniques have been proposed as a way to solve this issue by adapting training data from different, but related (source) datasets to the test (target) dataset. A problem in TL is how to quantify the relatedness of a source quickly and robustly. In this work, we present a fast domain similarity measure that captures the relatedness between datasets purely based on unlabeled data. Our method transfers knowledge from multiple sources by generating a weighted combination of domains. We show for multiple datasets that learning on such sources achieves an average overall accuracy closer than 2.5 percent to the results of the target classifier for semantic segmentation tasks. We further apply our method to the task of choosing informative patches from unlabeled datasets. Only labeling these patches enables a reduction in manual work of up to 85 percent.

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 unlabeled data are abundant and are already used successfully in unsupervised and semi-supervised learning methods, they cannot completely replace the dependence on labeled data. On the other hand, the acquisition of high quality, densely sampled and representative labeled samples 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 of domains.

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 unlabeled data are abundant, our proposed method is only based on similarity measurements between the marginal distributions of the features in the source and target domains. We apply our source selection method to two different data acquisition settings: domain selection and domain ranking. In *domain selection*, given a target domain and a list of candidate source domains, we assign weights to these sources based on the *Maximum Mean Discrepancy* (MMD) metric to the target. For these candidate source domains, we assume that some labeled training data is available from earlier surveys. We then apply *multi-source selection* by transferring knowledge from multiple weighted source domains simultaneously. Additionally, we extend the approach for DA presented in (Paul *et al.*, 2016) so that it can benefit from multi-source selection. For the *domain ranking* setting, we have to process many initially unlabeled target domains while no training data is available. Using our multi-source selection algorithm, our goal is to rank these domains in terms of their informativeness. This information helps us to select the most important domains for manual labeling, which leads to a reduced effort for the generation of training data while keeping classification error at an acceptable level. Finally, we propose an improvement of the MMD metric for the application in source selection with many candidate sources. This Asymmetric Maximum Mean Discrepancy is able to significantly reduce the memory footprint for each source while featuring a linear runtime complexity by exploiting the asymmetric relationship between target and source domains. We evaluate our methods on the Vaihingen and Potsdam datasets from the ISPRS 2D semantic labeing challenge (Wegner *et al.*, 2016) and on a third, even more challenging, dataset based on aerial imagery of three German cities.

Related Work

In our work, we use notation according to Pan and Yang (2010). A domain $\mathcal{D}=\{\mathcal{X}, \mathrm{P}(X)\}$ 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 a 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 a 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 $\mathcal{D}_T$ for the *domain selection* setting, and more than one target domain for the *domain ranking* setting. There are different settings of TL. Our focus is on DA, which is a special sub-category of the

Karsten Vogt and Jörn Ostermann are with the Institute für Informationsverarbeitung, Leibniz Universität Hannover (vogt@tnt.uni-hannover.de).

Andreas Paul, Franz Rottensteiner, and Christian Heipke are with the Institute of Photogrammetry and Geoinformation, Leibniz Universität, Hannover.

Photogrammetric Engineering & Remote Sensing
Vol. 84, No. 5, May 2018, pp. 249–261.
0099-1112/18/249–261

doi: 10.14358/PERS.84.5.249

transductive TL setting (Pan and Yang, 2010). There are slightly different definitions of the DA problem; we follow the definition of Bruzzone and Marconcini (2009) according to which different domains only 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 $P(C_S \mid X_S) \neq P(C_T \mid X_T)$. 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. However, 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 applications other than image classification.

Pan and Yang (2010) subdivide DA into two groups according to what is actually transferred: *feature representation transfer* and *instance transfer*. 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 group employ 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 labeled 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*. These methods try to directly refuse 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 those 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), this idea was adapted to logistic regression, which has a lower computational complexity in training for multiclass problems. 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 unlabeled data has been presented in (Amini and Gallinari, 2002). Training is based on expectation maximization (EM), and the semi-labels of the unlabeled data are determined according to the cluster membership of EM. In contrast to our DA technique, that method assumes the labeled and the unlabeled data to follow the same distribution.

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, *source selection*, would try to detect a relevant source prior to applying TL, 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 focused on extending these metrics to multivariate data by using 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 *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.

Besides TL, *active learning* has also been an active research topic with the aim to reduce manual labeling costs (Settles, 2010). *Active learning* methods select the most informative samples from an initially unlabeled training set which are presented to a human operator for labeling. Further samples may then be selected while taking the user feedback and the peculiarities of the classifier into account. Some ideas were proposed to utilize active learning for DA (Tuia *et al.*, 2011). While our *domain ranking* setting bears some similarity to *active learning*, our approach works at a coarser level and does not incorporate a user feedback loop, resulting in a much simpler user workflow and faster computation times.

In this paper, we present an unsupervised and a supervised method for source selection based on different distance metrics for domains. The work 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 reduce a distance between the distributions of the synthetic source and the target domains. We also propose variations of our distance metrics that are able to exploit the asymmetrical relationship between target and source domains in TL. Furthermore, we extend the algorithm in (Paul *et al.*, 2016) so that it can deal with multiple sources. Finally, we apply our proposed methods to rank a set of target domains in order of their informativeness. Only the most informative domains need to be labeled manually in order to generate high quality semantic segmentation for all targets.

Domain Adaptation

We start this section with a short description of the work from (Paul *et al.*, 2016) before presenting our improvements in the next section.

DA Approach

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

$$P\left(C = C^{k} \mid \mathbf{x}\right) = \frac{exp\left(\mathbf{w}_k^T \cdot \Phi(\mathbf{x})\right)}{\sum_j exp\left(\mathbf{w}_j^T \cdot \Phi(\mathbf{x})\right)} \tag{1}$$

where $\mathbf{w}_k$ is a parameter vector for a particular class label C^k to be determined in the training process for the class $k \in K$. 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 $g_n = 1$ for each sample $n \in \{1, \ldots, N\}$, 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 the parameter vector $\mathbf{w}$ (collecting the parameter vectors $\mathbf{w}_k$ for all classes k) given $\overline{TD}$ are determined by optimizing the posterior (Vishwanathan *et al.*, 2006):

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

where q_{nk} is 1 if $C_n = C^k$ and 0 otherwise, $p(C = C^k \mid \mathbf{x}_n, \mathbf{w})$ 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 $-log(p(\mathbf{w} \mid \overline{TD}))$ (Bishop, 2006).

Our aim is to transfer the classifier trained on labeled source domain data to the target domain in an iterative procedure. Our initial classifier is trained on the 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-labeled 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:

$$\overline{TD}^i = \left\{\left(\mathbf{x}_{S,r}; C_{S,r}; g_{S,r}\right)\right\}_{r=1}^{N_S^i} \bigcup \left\{\left(\mathbf{x}_{T,l}; \tilde{C}_{T,l}; g_{T,l}\right)\right\}_{l=1}^{N_T^i}.$$

The symbol $\tilde{C}_{T,l}$ denotes the *semi-labels* of the target samples, which are determined by applying a criterion based on the class labels of the k nearest neighbors (*knn*) of a sample in feature space. If the most frequent class label among the *knn* of an unlabeled 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 neighbors 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. As i is increased, N_S^i becomes smaller and N_T^i increases, until finally, only target samples with semi-labels are used for training.

At each iteration i, we have to define sample weights $g_{\overline{TD}}^i \in [0,1]$ for all training samples in $\overline{TD}^i$, where $\left\{g_{\overline{TD}}^i\right\} = \left\{\left\{g_{S,r}^i\right\}_{r=1}^{N_S^i} \bigcup \left\{g_{T,l}^i\right\}_{l=1}^{N_T^i}\right\}$. For simplicity, we refer to the weight of a sample as $g_{(\overline{TD},n)}^i$, $n \in \{1, .., N^i\}$ with $N^i = |\overline{TD}^i|$ the number of elements in that training set 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 $g_{(\overline{TD},n)}^i$ 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 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.

Multi-Source Logistic Regression DA

In this section, the method previously described is adapted for using data from multiple source domains for training. To formally state our problem, we define our current training data set as follows:

$$\overline{TD}^i = \bigcup_{s=1}^{|\mathbb{S}|} \left\{\left(\mathbf{x}_{S^s,r}; C_{S^s,r}; g_{S^s,r}\right)\right\}_{r=1}^{N_{S^s}^i} \bigcup \bigcup_{t=1}^{|\mathbb{T}|} \left\{\left(\mathbf{x}_{T^t,l}; C_{T^t,l}; g_{T^t,l}\right)\right\}_{l=1}^{N_{T^t}^i}, \tag{3}$$

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 of sample weights should decrease the weight of uncertain samples; the other one is required to deal with prior weights assigned to the individual source domains (See the next Section).

Sample Weights

The individual weights for the training samples should indicate the algorithm's trust in the correctness of the semi-labels, but the 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; Matasci *et al.*, 2012) 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 (Figure 1):

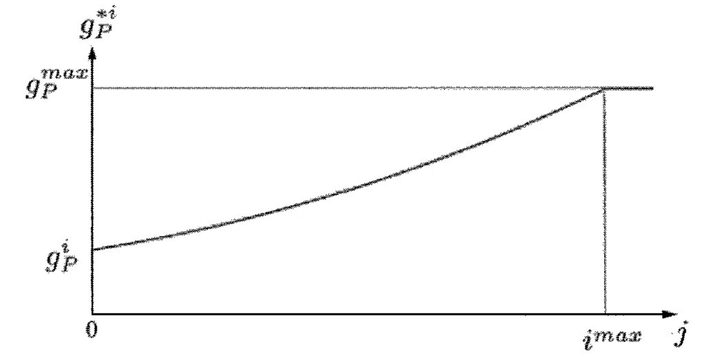


Figure 1. Sample weight function according to Eq. (4), assuming constant g_n^i during the adaptation.

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

In Equation 4, g_n^i is the weight of sample n in the current adaptation step i according to the 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 only one source domain were considered, the weight for each training sample n in $\overline{TD}^i$ would be g_n^{*i}, i.e., the algorithm outlined in the previous Section would be applied using the new definition of weights.

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 $\overline{S}$ (See the next Section) from multiple sources that 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 is $g_{\overline{TD},n}^{i} = g_n^{*i} \cdot \pi_{S^s}$, where g_n^{*i} is defined in Equation 4, whereas a sample n with a semi-label taken from the target domain has only 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 come from more similar source domains.

Multi-Source Selection

The goal of source selection is to improve the prospects of DA by choosing a source $\overline{S}$ that is, in some sense, most similar to the target domain. Naturally, one should prefer sources that produce similar decision boundaries as the target task. Therefore, the selection criterion should be based on $\varepsilon(h_S,\overline{TD}_T)$, i.e., the relative classification error ($\in [0,1]$) on the target data, given the predictive function h_S of the source task:

$$\overline{S} = \text{argmin}_{S\in\mathbb{S}}\,\varepsilon\left(h_S, \overline{TD}_T\right) \qquad (5)$$

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. We first design two complementary domain distance functions, which we 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} does not require any class labels at all. We refer to d_{DA} in places where either of these functions could be used. Equation 5 can then be approximated by $\overline{S}$ =argmin$_{S\in\mathbb{S}}d_{\text{DA}}(.)$. 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 developed variants of these domain distances that are able to capture the often asymmetric relationship between the target and source domains in TL. Finally, we also show how all critical hyperparameters can be tuned automatically in an efficient manner.

Similarity of Domains

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

$$\varepsilon\left(h_S.\overline{TD}_T\right) \le \varepsilon\left(h_S,\overline{TD}_S\right) + d_{\mathcal{A}}\left(\overline{TD}_T,\overline{TD}_S\right) + \gamma \qquad (6)$$

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 holds. 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 $d_{\text{SDA}} = \varepsilon(h_S,\overline{TD}_S)+d_{\text{A}}(\overline{TD}_T,\overline{TD}_S)$. In the following, we define $d_{\mathcal{A}}$ and derive a more computationally friendly way to estimate this distribution distance. In (Ben-David *et al.*, 2007), the A-distance is defined as:

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

The term $\varepsilon(h_{T\perp S},\overline{TD}_{T\perp S})$ describes the classification error for a classifier discriminating between feature vectors from the source and target domains. In the referenced 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 source and target domains is explicitly assumed. It is therefore desirable to find an approximation to the $\mathcal{A}$-distance that displays more favorable properties. Gretton *et al.* (2012) independently proposed the Maximum Mean Discrepancy (MMD) as a general distance function between probability distributions:

$$\begin{aligned} d_{MMD}^2\left(\overline{TD}_T,\overline{TD}_S\right) &= E\left[(\phi(\mathbf{x}_T) - \phi(\mathbf{x}_S))^2\right] \\ &= E\left[k(\mathbf{x}_T,\mathbf{x'}_T)\right] - 2E\left[k\left(\mathbf{x}_T,\mathbf{x}_S\right)\right] + E\left[k\left(\mathbf{x}_S,\mathbf{x'}_S\right)\right] \end{aligned} \qquad (8)$$

where $\mathbf{x}$ and $\mathbf{x}'$ are statistically independent samples from the same distribution. 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

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

holds for positive bounded kernels such as the Gaussian kernel:

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

Evaluation of the MMD can be done by replacing the expectations in Equation 8 with their empirical estimates. A naive estimator would have a run-time complexity of $\mathcal{O}(N_T{\cdot}N_S)$, where N_T and N_S are the numbers of features available in the target and source domains, respectively, 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:

$$\begin{aligned} & d_{LMMD}^2\left(\overline{TD}_T,\overline{TD}_S\right) \\ &= \frac{2}{M}\left[\sum_{r=1}^{M/2} k(\mathbf{x}_{T,2r},\mathbf{x}_{T,2r-1}) - \sum_{r=1}^{M} k(\mathbf{x}_{T,r},\mathbf{x}_{S,r}) + \sum_{r=1}^{M/2} k(\mathbf{x}_{S,2r},\mathbf{x}_{S,2r-1})\right] \end{aligned} \qquad (11)$$

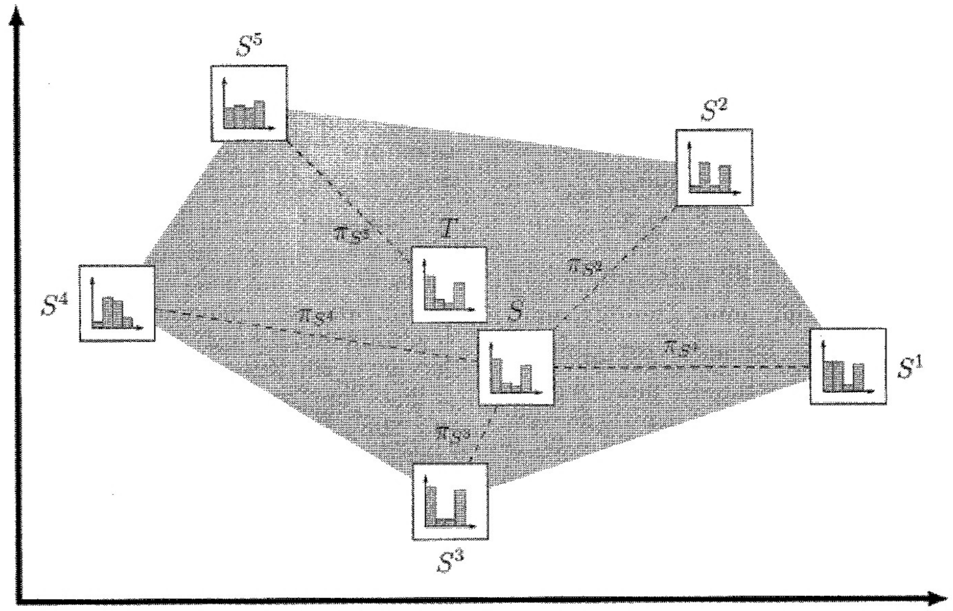


Figure 2. A synthesized source $\bar{\boldsymbol{S}}$ is formed as the convex combination of candidate sources $\boldsymbol{S}^s$.

Finally, replacing d_A by d_{LMMD} using Equation 9 leads to the definition of our supervised domain distance:

$$d_{SDA}(\overline{TD}_T,\overline{TD}_S)=\varepsilon\left(h_S,\overline{TD}_S\right)+2d_{LMMD}(\overline{TD}_T,\overline{TD}_S) \qquad (12)$$

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

$$d_{UDA}\left(\overline{TD}_T,\overline{TD}_S\right)=2d_{LMMD}\left(\overline{TD}_T,\overline{TD}_S\right) \qquad (13)$$

Asymmetric Domain Distance

The described domain distances based on the MMD, see Equations 12 and 13, are theoretically motivated and assume a symmetric relationship between the source and target domains, e.g., if a classifier learned from $\overline{TD}_S$ performs well on $\overline{TD}_T$, then the reverse must also be true. In reality, this assumption may not always hold. For instance if $\overline{TD}_T \subset \overline{TD}_S$ we should expect that a classifier learned on S will perform well on T as all classes are well represented by the training data. Yet, the distributions are measurably different, which can be observed by a high MMD measure. We therefore propose a modification of the MMD which is aimed at directly measuring whether all regions in T are represented in S, while being invariant to those regions in S that are conversely not represented in T. First, let us re-examine the MMD from Equation 8:

$$\begin{aligned}d^2_{MMD}\left(\overline{TD}_T,\overline{TD}_S\right)&=E\left[k(\mathbf{x}_T,\mathbf{x'}_T)\right]-2E\left[k(\mathbf{x}_T,\mathbf{x}_S)\right]+E\left[k\left(\mathbf{x}_S,\mathbf{x'}_S\right)\right]\\&=\underbrace{\left(E\left[k(\mathbf{x}_T,\mathbf{x'}_T)\right]-E\left[k(\mathbf{x}_T,\mathbf{x}_S)\right]\right)}_{Target}\\&+\underbrace{\left(E\left[k(\mathbf{x}_S,\mathbf{x'}_S)\right]-E\left[k(\mathbf{x}_S,\mathbf{x}_T)\right]\right)}_{Source}\end{aligned} \qquad (14)$$

It should be noted that this is valid for the Gaussian kernel since $k_{\text{RBF}}(x,y) = k_{\text{RBF}}(y,x)$. The term $E[k(\mathbf{x}_S, \mathbf{x'}_S]$ describes the compactness of the source domain, which is mostly irrelevant for measuring the relatedness of a source. We therefore drop the entire second part of this equation. The remaining terms describe the average intra domain similarity ($T \Leftrightarrow T$) and inter domain similarity ($T \Leftrightarrow S$), respectively. We argue that for each sample in the target domain to be well represented, a related source should contain at least one sample that is not significantly more dissimilar than its next most similar target sample. Therefore, we propose to replace the simple average in the MMD with a maximum operator over similarity scores:

$$\begin{aligned}d^2_{AMMD}\left(\overline{TD}_T,\overline{TD}_S\right)&=\frac{1}{N_T}\sum_{i=1}^{N_T}\max_{j\in[1..N_T]}k\left(\mathbf{x}_{T,i},\mathbf{x'}_{T,j}\right)\\&-\frac{1}{N_T}\sum_{i=1}^{N_T}\max_{j\in[1..N_S]}k\left(\mathbf{x}_{T,i},\mathbf{x}_{S,j}\right)\end{aligned} \qquad (15)$$

A disadvantage of this formulation is that in order to find the most similar sample we always have to look at the entire training set. Therefore, Equation 15 has a quadratic run-time complexity. Yet, suppose we are content with finding only the q% most similar samples and we further also allow a failure probability p for locating such a sample. Then, it can be shown that it is sufficient to only look at a random subset of size $N_{\max} \geq \log_{(100\%-q)}(p)$, irrespective of the underlying data distribution or sample size (See Appendix A for the proof). Consequently, Equation 15 can be approximated without significant loss of accuracy using a procedure having linear runtime complexity. This result also has the benefit that a source selection system based on our AMMD never has to store the full source training sets to perform queries. In fact, for each source only $N_{\max}$ samples have to be held in memory, where $N_{\max}$ is typically less than 100. Using these results, one can use the metric d_{AMMD} to obtain modified (asymmetric) versions of the domain distances d_{SDA} and d_{UDA}:

$$d_{A-SDA}(\overline{TD}_T,\overline{TD}_S)=\varepsilon\left(h_S,\overline{TD}_S\right)+2d_{AMMD}(\overline{TD}_T,\overline{TD}_S) \qquad (16)$$

$$d_{A-UDA}(\overline{TD}_T,\overline{TD}_S)=2d_{AMMD}(\overline{TD}_T,\overline{TD}_S) \qquad (17)$$

Convex Combination of Domains

In general, we have to expect 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 (Figure 2). Any point in this subspace represents a valid marginal distribution and can be parametrized as:

$$p_{S_\pi}(\mathbf{x}) = \sum_{s=1}^{|\mathbb{S}|} \pi_{S^s} p_{S^s}(\mathbf{x}) \tag{18}$$

given a source weight vector $\pi = \left[\pi_{S^1}, \ldots, \pi_{S^{|\mathbb{S}|}}\right]^T$ satisfying the constraints $\pi_{S^s} \geq 0, \sum_{s=1}^{|\mathbb{S}|} \pi_{S^s} = 1$. By definition (Equation 18), the distribution $p_{S_\pi}(\mathbf{x})$ is a mixture of the source marginal distributions. The weighted training set $\overline{TD}_{S_\pi} = \bigcup_{s=1}^{|\mathbb{S}|} \left\{\mathbf{x}_{S^s}; C_{S^s}; \pi_{S^s}\right\}_{r=1}^{N_{S^s}}$ is therefore a representative sample of 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 extensions of the linear-time MMD estimator (Equation 11) and our asymmetric MMD (Equation 15) to a weighted union of source training sets:

$$\begin{aligned} d^2_{LMMD}\left(\overline{TD}_T, \overline{TD}_{S_\pi}\right) \\ = \frac{2}{M}\Bigg[& \sum_{r=1}^{M/2} k\left(\mathbf{x}_{T,2r}, \mathbf{x}_{T,2r-1}\right) - \sum_{u=1}^{|\mathbb{S}|} \pi_{S^u} \sum_{r=1}^{M} k\left(\mathbf{x}_{T,r}, \mathbf{x}_{S^u,r}\right) \\ & + \sum_{u=1}^{|\mathbb{S}|} \sum_{v=u+1}^{|\mathbb{S}|} \pi_{S^u} \pi_{S^v} \sum_{r=1}^{M} k\left(\mathbf{x}_{S^u,r}, \mathbf{x}_{S^v,r}\right) \\ & + \sum_{u=1}^{|\mathbb{S}|} \pi^2_{S^u} \sum_{r=1}^{M/2} k\left(\mathbf{x}_{S^u,2r}, \mathbf{x}_{S^u,2r-1}\right) \Bigg] \end{aligned} \tag{19}$$

$$\begin{aligned} d^2_{AMMD}\left(\overline{TD}_T, \overline{TD}_{S_\pi}\right) = & \frac{1}{N_T} \sum_{i=1}^{N_T} \max_{j \in [1..N_T]} k\left(\mathbf{x}_{T,i}, \mathbf{x}'_{T,j}\right) \\ & - \frac{1}{N_T} \sum_{u=1}^{|\mathbb{S}|} \pi_{S^u} \sum_{i=1}^{N_T} \max_{j \in \left[1..N_{S^u}\right]} k\left(\mathbf{x}_{T,i}, \mathbf{x}_{S^u,j}\right) \end{aligned} \tag{20}$$

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

Fast Synthesis of Source Domains by Boosting

Convex representation problems, like the one in Equation 18, are related to dictionary learning. The Iterative Nearest Neighbor (INN) algorithm (Timofte and Van Gool, 2012) is a recent method that approximately solves such problems in a greedy fashion. The solution at iteration L is given as:

$$p_S^L(\mathbf{x}) = \sum_{l=1}^{L} w^l p_{S_l}(\mathbf{x}), \tag{21}$$

where the iteration weights are computed as:

$$w^l = \frac{\lambda}{(1+\lambda)^l} \tag{22}$$

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:

$$S_{L+1} = \underset{S \in \mathbb{S}}{\operatorname{argmin}}\, d_{DA}\left(\overline{TD}_T, \left\{\mathbf{x}_{S,r}; C_{S,r}; w^{L+1}\right\}_{r=1}^{N_S} + \bigcup_{l=1}^{L} \left\{\mathbf{x}_{S_l,r}; C_{S_l,r}; w^l\right\}_{r=1}^{N_{S_l}}\right) \tag{23}$$

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

$$\pi_{S^s} = \sum_{l=1}^{L} w^l \cdot \mathbf{1}_{\left\{S_l = S^s\right\}}. \tag{24}$$

Originally, the INN algorithm was designed to work on vectors in Euclidean spaces. When interpreted in the space o 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. In addi tion, the update step in Equation 23 has the effect to steer the optimization successively to prioritize parts of the distribution that are not yet well represented while also attenuating overrepresented parts.

The sum $\sum_{l=1}^{\infty} w^l$ approaches 1 while the iteration weights w^l 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 Equation 22 follows

$$L = -\frac{\log(1-\beta)}{\log(1+\lambda)} \tag{25}$$

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

Algorithm 1 Kernel Bandwidth Estimation

```
$\phi \leftarrow 1.61803398875$
$(L,R) \leftarrow (0, \pi/2)$
$(A,B) \leftarrow (R - (R-L)/\phi,\ L + (R-L)/\phi)$
for $i = 1..\mathrm{MaxIter}$ do
    $f_A \leftarrow d^2_{MMD}(\overline{TD}_T, \overline{TD}_S)$ with $\sigma = \tan(A)$
    $f_B \leftarrow d^2_{MMD}(\overline{TD}_T, \overline{TD}_S)$ 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
    $(A,B) \leftarrow (R - (R-L)/\phi,\ L + (R-L)/\phi)$
end for
return $\sigma_{\max} = \tan((L+R)/2)$
```

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_{MMD} with respect to σ:

$$d^2_{\overline{MMD}}\left(\overline{TD}_T, \overline{TD}_S\right) = \max_{\sigma \in (0,\infty)} d^2_{MMD}\left(\overline{TD}_T, \overline{TD}_S\right) \tag{26}$$

Using the results by Shestopaloff (2010), we can show that this optimization problem has exactly one maximum at $\sigma_{\max}$ and at most one minimum at $\sigma_{\min}$. Furthermore, if $\sigma_{\min}$ exists then $\sigma_{\max} < \sigma_{\min}$ holds. Finally, d_{MMD} will tend towards zero for both $\sigma \to 0$ and $\sigma \to \infty$. We can therefore conclude that $d_{\mathrm{MMD}}(\sigma_{\min})<0$ if a minimum exists. Whereas theoretically, the MMD only can take positive values, this case can still occur for very similar domains due to errors in the empirical estimates of the expectations. The general shape of the function $d_{\overline{MMD}}$ is shown in Figure 3.

We solve this optimization problem using a *Golden-Section-Search* (GSS) (Press, 2007) (see Algorithm 1). The GSS searches the maximum of a strictly unimodal function. We modified the GSS to handle the case where a minimum $\sigma_{\min}$ exists. 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. Our empirical evaluation in the Experiments Section shows that the same approach is also valid for our asymmetrical MMD.

Improving Robustness by Bootstrap Aggregation

As all empirical estimators, our MMD estimators have a non-zero estimation variance which may result in a suboptimal solution $\boldsymbol{\pi}$. We propose to reduce this variance by averaging $\boldsymbol{\pi}$ 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 using independent draws with replacement from an input sample. The statistical properties of bootstrap sampling are described in detail in (Hesterberg *et al.*, 2003).

Ranking of Source Domains

The *domain ranking* setting might resemble a more relevant workflow for the supervised classification of remote sensing imagery than source selection as previously presented. We assume that we have to process a batch of E images for which initially no training data is available. In order to create some training data we have to label some of these images manually. Obviously, we do not intend to label all of them. In this setting all images can be considered as target domains $T_e \in \mathbb{T}$, while only some of them will also be used as source domains S_s for our source-selection algorithm. Our goal is to find a small subset S_s that will be sufficient to achieve acceptable classification results. A reasonable workflow could be to label source domains sequentially, training a classifier whenever a new source domain is added and applying that classifier to all target domains; a visual inspection of the results could guide the decision when to stop labeling new domains. A *domain ranking* algorithm must therefore be able to compute an ordering of the domains of the batch in which the most informative domains are placed early. For computational reasons we have chosen to restrict our research to greedy algorithms. We use a variant of the *kernel herding* algorithm by Chen *et al.* (2012). *Kernel herding* greedily selects a small representative *super sample* from a larger sample of an unknown distribution. At each step, we select a new sample that is similar to many other samples while also being dissimilar to already selected ones. Due to its formulation as a kernel method, *kernel herding* is flexible enough to be adapted to the *domain ranking* problem. Only a kernel matrix K needs to be defined which encapsulates a pairwise similarity measure between domains. A simple kernel matrix could be directly constructed from the MMD as $k_{i,j}^{MMD} = 1 - d_{MMD}\left(\overline{TD}_{T_i}, \overline{TD}_{T_j}\right)$. While this simple approach typically produces good results, we have determined empirically that it can be far from the optimum if less than five sources are to be selected. Therefore, we propose a more elaborate method to supersede the simple $k_{i,j}^{MMD}$ domain kernel. We first note that the source weights $\boldsymbol{\pi}$ from our multi-source selection algorithm also describe a domain similarity, as more related sources are associated with larger weights. To construct K we first apply multi-source selection to each T_e using any of our unsupervised domain distances $(d_{[\mathrm{UDA,A\text{-}UDA}]})$ while using all other domains as candidate sources. We define the e^{th} column vector of K as the source weight vector $\boldsymbol{\pi}_S$ for the e^{th} domain. We also have to consider self-similarity of domains by setting the main diagonal of K to 1. The *kernel herding* algorithm then starts with an empty set S_s of selected source domains. At each iteration the next most informative source domain S^{select} is chosen as:

$$S^{select} = \underset{T_u \in \mathbb{T} \setminus S_s}{\operatorname{argmax}} \left[\frac{1}{|\mathbb{T}|} \sum_{T_e \in \mathbb{T}} k_{u,e} - \frac{1}{|S_s|+1} \sum_{S_v \in S_s} k_{u,v} \right] \tag{27}$$

and added to S_s. The main result of the algorithm is the order in which datasets should be selected for labeling so that they can serve as source domains.

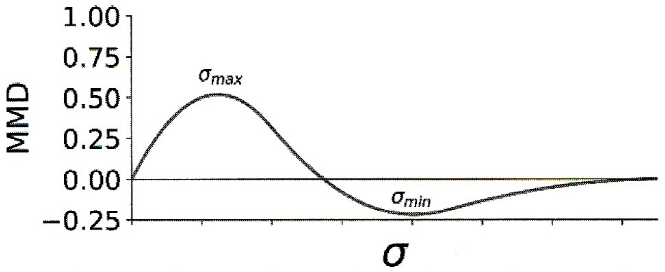
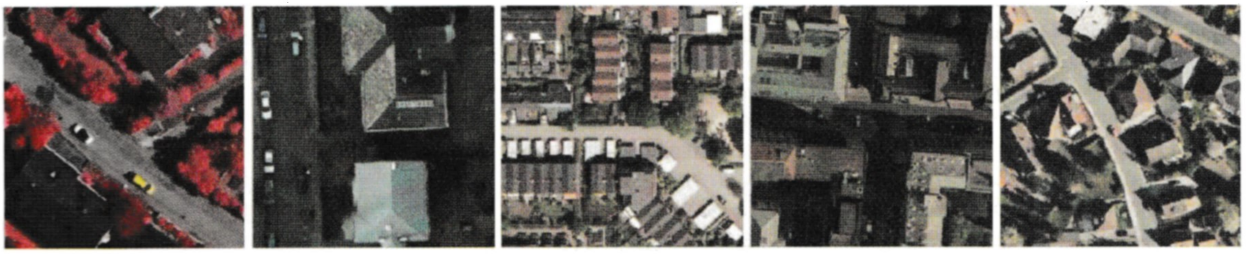


Figure 3. General shape of the optimization problem for hyperparameter estimation. The optimal $\boldsymbol{\sigma}$ can be found at $\boldsymbol{\sigma}_{\max}$.



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

Figure 4. Details of orthophotos of all datasets used in the experiments.

Experiments

Test Data and Test Setup

Our experimental evaluation is based on three datasets (see Figure 4). Two of them are the Vaihingen and Potsdam datasets from the ISPRS 2D semantic labeling 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. Each region covers an area of 2×2 km^2 but is evenly split up into nine patches. The reference data for the 3CITYDS dataset was generated manually based on the image data. For all datasets, both orthophotos and digital surface models (DSM) generated by image matching are available. The properties of all datasets are given in Table 1. Finally, we also consider a fourth dataset, *Combined*, which is the union of the Vaihingen, Potsdam, and 3CITYDS datasets.

All experiments are based on a pixel-wise classification of the input data into the four object classes *building, tree, low vegetation*, and *impervious surface*. The *impervious surface* class also includes clutter and cars. Furthermore, we used the same feature space for all datasets. Under this constraint, we selected the five most discriminative features using a *Random Forest*-based feature selection method (Breiman, 2001) from a pool of spectral, structural, and texture features. We settled on the *normalized difference vegetation index* (NDVI), *normalized digital surface model* (NDSM) and the pixelwise red, green and near infrared spectral components.

Table 1. Dataset properties. *GSD*: ground sampling distance. R/G/B/I: red / green / blue / near infrared band; patches: number of patches per data set; features / classes: numbers of features used / classes discerned in classification

Dataset	GSD	Channels	Patches	Features	Classes
Vaihingen	8 cm	RGI	15	5	4
Potsdam	8 cm	RGBI	23	5	4
3CITYDS	20 cm	RGBI	27	5	4

In this section, we present an experimental evaluation for two different data acquisition settings. The first, *domain selection*, corresponds to a setting in which only one new target image needs to be classified while large quantities of labeled images are already available from earlier surveys. For the *domain ranking* setting, we assume that a large amount of target images has to be classified and that initially no training data is available, so that domain ranking is applied to determine which images should be labeled to serve as source domains. In all experiments, the evaluation is based on metrics derived from the overall accuracy (OA), i.e., the percentage of correctly classified pixels when comparing the classification results to a reference.

Domain Selection

A successful source selection should be able to find related sources and reduce the expected classification error. The evaluation consists of two parts. First, we analyze our proposed multi-source selection method. Our method is applied to each patch (=T) to synthesize a source $\overline{S}$ using all remaining patches of the dataset as candidate sources. For the *domain selection* setting, we assume that these candidate sources are fully labeled. We examine several source selection strategies. *Single source selection* selects only one source domain that has the lowest domain distance to the target domain while *multi-source selection* utilizes labeled samples from all source domains using source weights as previously described. We examine both strategies in combination with both domain distances $d_{\{SDA,UDA\}}$ and their asymmetric variants $d_{\{A\text{-}SDA,A\text{-}UDA\}}$.

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, uses all sources and assigns them 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.

In our second experiment, we enable the DA extension for our classifier, applying it to a synthesized source $\overline{S}$ generated by our unsupervised asymmetric multi-source selection algorithm using only the 1 to 3 sources featuring the largest source weights.

Source selection and DA are applied using pixels on a regular grid of size 10 px to 30 px to reduce spatial dependency; the grid size was adapted to the GSD and the patch size of the individual datasets, thus using only about 0.25 percent of the data in these processes (while using all data for evaluation). For the source selection, we selected about 80 percent of these pixels per patch for each bootstrap run. For the logistic regression classifier, we applied a polynomial expansion of degree 2. The entire set of parameters used for DA is given in Table 2, whereas Table 3 shows the parameters used for source selection. The DA parameters were tuned empirically on a small random subset of patches across all datasets. The same parameter values were used for all datasets without further tuning. The source selection parameters are non-critical and were set to achieve a good tradeoff between speed and performance. As source selection has some random components, each experiment is repeated ten times, and we report average quality indices.

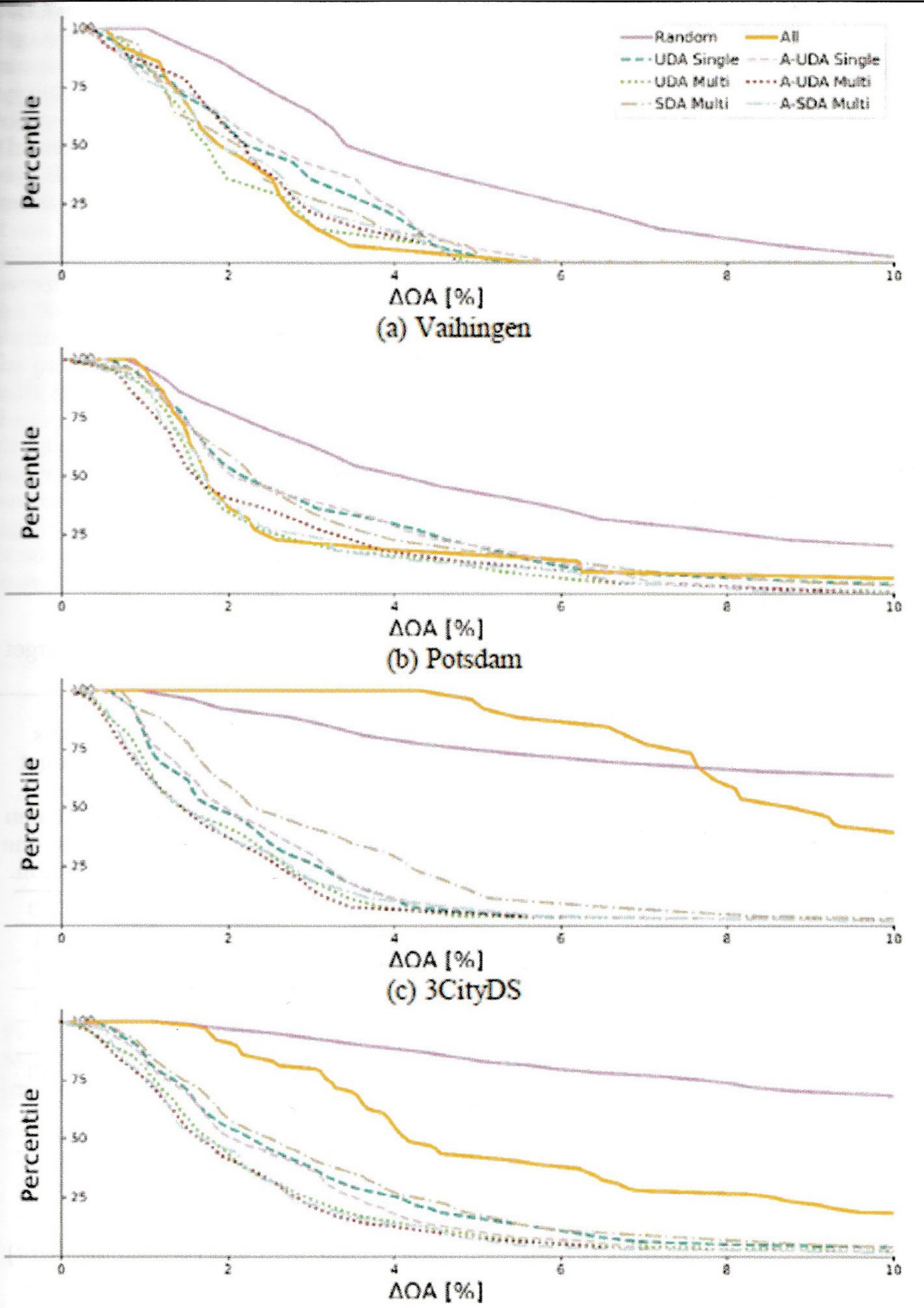
Table 2. Parameters used for the DA method previously described. σ_0, σ_{DA}: Weights for the gaussian priors for regularization used for training the initial classifier and in the DA process, respectively. ρ_E, ρ_A: number of samples per class for transfer and elimination. KNN: number of neighbors in the KNN analysis for deciding which target samples to include for training. *h*: parameter of the weight function $g_{\overline{TD},n}^{i}$ (Paul *et al.*, 2016). $i^{\max}$, $g_{P,S}^{max}$, $g_{P,T}^{max}$: parameters of the weight function in Equation 4, in case of $g^{\max}$ for source and target domain, respectively.

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

Table 3. Parameters of multi-source selection. MaxIter GSS: Maximum number of iterations of Golden-Section-Search. INN λ: parameter of the weight function in equation 22. INN β: threshold for the sum of weights for generating a synthetic source domain. Bootstrap runs / size: number of bootstrap runs for synthetic source generation and number of samples used in each run, respectively. N_{max}: number of samples used to determine the asymmetric domain distance.

MaxIter GSS	INN λ	INN β	Bootstrap Runs	Bootstrap Size	N_{max}
10	0.5	0.9	10	5000	60

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



(a) Vaihingen

(b) Potsdam

(c) 3CityDS

Figure 5. Source selection results. ΔOA: difference in overall accuracy compared to a classifier based on target training data. Percentile: the percentage of patches in the data set for which ΔOA is smaller than the value on the abscissa. Example (Vaihingen, *All Sources*): for 25% of the target patches the loss in OA is larger than 3% (ΔOA > 3%).

Table 4. Source selection results for different variants of the algorithm as previously explained. Mean ΔOA: Average loss in overall accuracy when compared to a classifier based on target training data in 10 test runs (lower is better). STDEV: standard deviation of ΔOA over 10 test runs.

				UDA	A-UDA	UDA	A-UDA	SDA	A-SDA
		Random	All	Single	Single	Multi	Multi	Multi	Multi
Vaihingen	Mean ΔOA	4.4	2.2	2.5	2.7	2.1	2.3	2.5	2.3
	Stdev	2.9	1.3	1.5	1.6	1.3	1.3	1.4	1.3
Potsdam	Mean ΔOA	6.2	3.1	3.4	3.1	2.5	2.6	3.3	2.5
	Stdev	5.9	3.5	3.3	2.4	2.3	2.3	3.0	2.1
3CITYDS	Mean ΔOA	26.6	10.6	2.6	2.7	2.3	2.2	3.4	2.3
	Stdev	22.5	5.0	2.8	2.8	2.8	2.7	3.3	2.6
Combined	Mean ΔOA	20.5	7.5	3.2	2.8	2.5	2.3	3.3	2.4
	Stdev	15.6	7.4	2.9	2.6	2.5	2.3	2.8	2.2

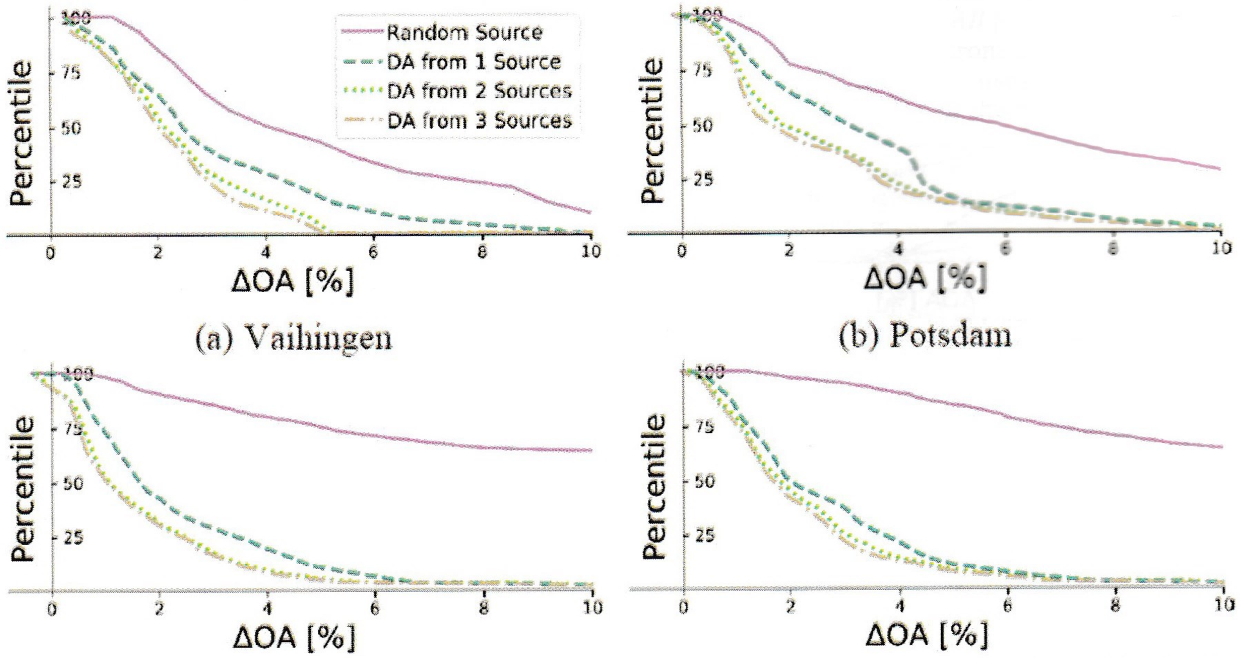
Domain Ranking

For this experiment, we evaluate our proposed domain ranking algorithm. The goal is to achieve a high overall accuracy while only using sources from a small set of candidates, thus reducing the work related to manually labeling these sources. Therefore, we only use the most informative domains as source candidates as defined by the domain ranking produced by the *kernel herding* algorithm. Previous experiments have shown that single source selection with our new asymmetric domain distance ($d_{\text{A-UDA}}$) is competitive with our best multi-source method while also being much faster to compute. For this reason, we ran the *domain ranking* experiments using this source selection method only. To give a context to our results, we also provide an upper and lower bound of the average overall accuracy for the datasets. When only a single labeled source was used, the upper bound was determined by testing all patches as sources, selecting the source that maximized the average overall accuracy over the entire dataset. The bound for larger sets of sources was estimated in a greedy manner by iteratively adding source candidates using the same criterion. The lower bound was generated similarly by minimizing the average overall accuracy.

Results and Discussion

Domain Selection

Figure 5 and Table 4 show the evaluation of source selection without using DA. The evaluation is based on ΔOA = $OA_{TT} - OA_{ST}$, where OA_{TT} is the overall accuracy achieved on the target dataset when training the classifier on a labeled target dataset and OA_{ST} is the overall accuracy on the target dataset when training on a synthesized source. Thus, ΔOA directly shows how much performance is lost by not having access to class labels in the target domain, and it should be as small as possible. We present percentile plots and the average ΔOA as well as the standard deviation (STDEV) of ΔOA over 10 test runs for each dataset separately. The percentile plots show the cumulative distribution of ΔOA over all patches in a dataset. Generally, we strive to achieve large losses (right side on the percentile plots) for only a small number of patches in a dataset (bottom of the percentile plots). The results do not exhibit too many surprises. With all datasets, random selection is clearly inferior to all other tested methods. Furthermore, using multiple weighted sources usually outperforms single source selection. Our asymmetric MMD generally performs similarly to their symmetric versions. Yet, while the MMD



(a) Vaihingen

(b) Potsdam

Figure 6. Multi-source domain adaptation results. ΔOA: difference in overall accuracy compared to a classifier based on target training data. For the interpretation of the figures, cf. Figure 5.

is evaluated on the entire training sets, the AMMD only ever has access to N_{max} samples of each source training set. In contrast to the experiments in (Vogt *et al.*, 2017), the supervised domain distances generally perform worse than their unsupervised variants. The core idea is that the d_{SDA} adds a bias to prefer sources that have a larger margin, and therefore also a simpler decision boundary. It appears that this bias might not always be desirable and its application should depend both on the feature space and on the classification method. Surprisingly the AMMD seems to be less affected. We currently do not have an explanation for this observation. For the Vaihingen dataset, the tested methods result in very similar results, which is a behavior different from the one for the other datasets. As most patches in the Vaihingen dataset have a very similar appearance and class distribution, the gains from using TL methods should be expected to be small. The 3CITYDS and Combined datasets, on the other hand, present a more difficult challenge due to the pronounced inhomogeneity between patches. While both naive source selection strategies, *Random Source* and *All Sources*, perform particularly bad here, our proposed multi-source selection methods manage to achieve stable performance (≲2.5%) across all datasets.

Table 5. Multi-source domain adaptation results. Mean ΔOA: the average loss in overall accuracy (OA) after DA when compared to a classifier based on target training data (lower is better); the average of 10 test runs is reported. Stdev: standard deviaton of ΔOA over 10 test runs. ΔDA: the improvement in OA when enabling DA (higher is better). DA1-3 applies domain adaptation to the best one to three sources based on our unsupervised asymmetric multi-source selection.

	Random	DA1	DA2	DA3
Mean ΔOA	5.3	3.2	2.4	2.2
Stdev	3.9	2.6	1.5	1.4
ΔDA	-0.9	-0.5	0.0	0.1

(a) Vaihingen

	Random	DA1	DA2	DA3
Mean ΔOA	8.0	3.5	2.8	2.6
Stdev	7.7	2.7	2.5	2.4
ΔDA	-1.8	-0.4	-0.1	-0.1

(b) Potsdam

	Random	DA1	DA2	DA3
Mean ΔOA	26.3	2.6	2.0	1.9
Stdev	22.1	2.8	2.8	2.8
ΔDA	0.4	-0.1	0.2	0.3

(c) 3CityDS

	Random	DA1	DA2	DA3
Mean ΔOA	19.6	2.8	2.4	2.3
Stdev	15.4	2.6	2.4	2.3
ΔDA	0.8	-0.3	0.0	0.1

(d) Combined

Figure 6 and Table 5 show the DA results using a random source and the 1 to 3 best sources according to unsupervised multi-source selection using our asymmetric MMD metric. Again, the evaluation is based on ΔOA as described earlier in this section. In addition, we compared the OA on the target data with and without enabling the DA extension in logistic regression.

We report ΔDA= $OA_{ST}^{DA} - OA_{ST}$, where OA_{ST}^{DA} is the overall accuracy on the target dataset when training on a synthesized source after domain adaptation. ΔDA can be understood as the mean difference in OA due to enabling DA over all patches of a dataset, where positive ΔDA represents a positive transfer. The test shows that using multiple sources always improves the prospects of DA (indicated by ΔDA > 0), but this effect also seems to diminish quickly when a larger number of sources is used. Compared to the results in (Vogt *et al.*, 2017) the gains of using DA seem to be reduced for more complex feature spaces. It can be observed that DA still shows the greatest benefits for complex and inhomogeneous datasets, like the 3CITYDS or *Combined* datasets. Our initial working hypothesis was that applying instance-transfer based DA to a related source should improve the expected gains, with the goal to achieve positive transfer in most target domains. Our experiments have shown that while selecting a related source is a necessary condition to this end, it does not appear to be sufficient alone.

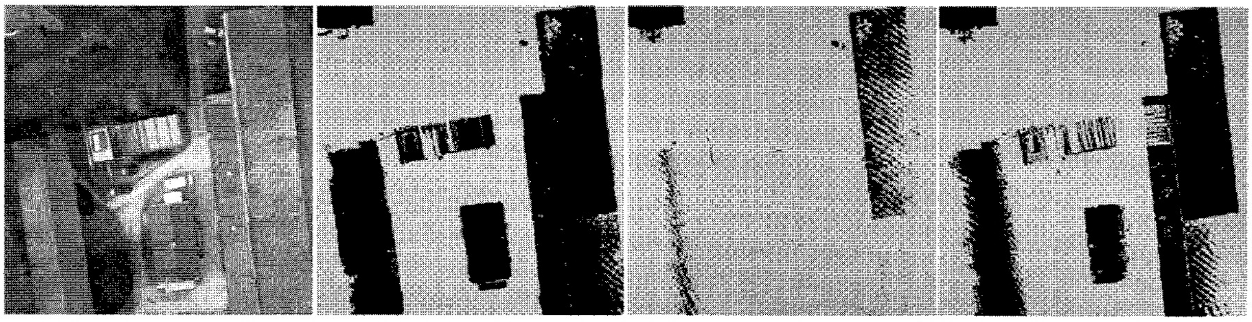
Despite the modest improvements in overall accuracy, DA may still be worthwhile for some applications. Figure 7 shows an example for the class *building* from our DA experiments using the *Combined* dataset. The figure shows that the synthesized source sometimes failed to reproduce low buildings with flat roofs; obviously, even in the synthesized source the DSM heights were not representative for such buildings in these cases. These buildings may be recovered using DA, as seen in Figure 7d. The overall pixel count covered by such objects remains small compared the patch size, which explains their low impact on the measured ΔDA values.

Domain Ranking

Figure 8 shows the results of our *domain ranking* experiments. The diagrams plot the average OA for a dataset when applying source selection as a function of the number N_l of source domains that are assumed to provide training data. The order in which the domains are considered for labeling and, thus, to be included in the set of available source domains, is the one predicted by our *domain ranking* procedure. It can be easily seen that our proposed method is capable of selecting the most important sources with a high degree of certainty. The results of our kernel herding approach follow the theoretical optimum very closely on all datasets. For the *Vaihingen, Potsdam* and 3CITYDS datasets, less than five of the patches would have to be labeled manually to achieve results closer than 2 percent in OA to a fully labeled dataset. For the *Combined* dataset, this figure can be stated as less than 10 patches. Considering the evaluated datasets, our proposed algorithm would be able to save more than 66 percent to 85 percent in manual labeling cost while only incurring a negligible amount of loss in OA. While the performance for few candidate sources is already quite satisfactory, the plots also show a very slow convergence to the optimum afterwards. It appears that while our proposed kernel matrix does contain enough information to confidently rank the most important domains, it cannot do so for the more uninformative domains. We tested this hypothesis by repeating kernel herding with small random perturbations to K. We notice that the absolute domain ranking quickly becomes unstable after the first few ranks. Yet, for practical applications, we do not expect this to become a significant problem.

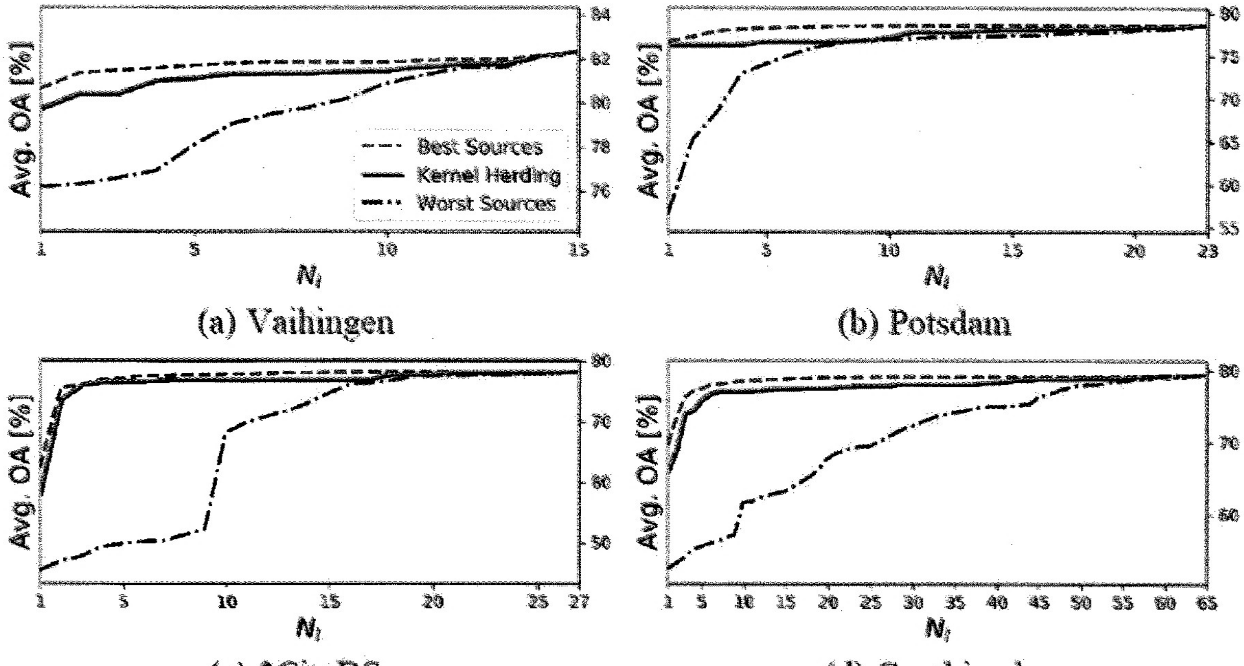
We also provide runtime measurements for our single source selection based on the $d_{A\text{-}UDA}$ domain distance. For instance, in the experiments based on the *combined* dataset, computing the source weights for a single target takes 6.6 sec using our GPGPU implementation[2] on a single NVIDIA GTX 1060. Applying *domain ranking* on this dataset therefore takes only about seven minutes. It should be noted that this performance scales linearly with the size of the target training set, the number of source domains and the number of features, yet remains constant with reference to the sizes of the source training sets.

[2] Can be made available by the corresponding author on request



(a) Visual (b) Target Classifier (c) Source Selection (d) Domain Adaptation

Figure 7. Example for the classification results for class *building* from the *Combined* dataset. Buildings are printed black. (a) Image (b) Results of a classifier trained on target data (c) Results after multi-source selection without DA using three sources (d) Results with DA.



(a) Vaihingen (b) Potsdam

(c) 3CityDS (d) Combined

Figure 8. Domain ranking results. Avg. *OA*: average overall accuracy over the dataset for different numbers of candidate source domains (higher is better). N_l: number of source domains that provide labeled training data.

Conclusions

In this work, we presented two domain distance measures based on the MMD and their variants that are able to capture the asymmetric relationship between target and source domains in a supervised learning setting. The supervised domain distances require labeled samples in the source domain, while the unsupervised distances operate without using any labels. 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 target training set. More importantly, our proposed asymmetric MMD metric has a small memory footprint since it requires less than 100 samples from each source domain and is thus applicable to very large datasets. We also expanded an existing DA method to cope with multiple sources being assigned different weights.

Our experiments show that multi-source selection is consistently able find related sources from a large set of candidate sources. The average loss in classification performance very predictably remains below 2.5 *percent* when compared to a classifier that has full access to labeled samples in the target domain over a variety of datasets. Additionally applying DA achieved a small positive transfer when using the weighted combination of two or more sources selected by our unsupervised procedure. Yet, this gain is quite small and could not be achieved for all datasets. Finally, we examined a scenario where only unlabeled data is available. We applied our source selection method to find the most informative domains. We have shown these informative domains to be good candidates for manual labeling and that an acceptable classification accuracy can be achieved while reducing manual work by up to 85 percent. For our experiments, we have assumed a shared feature space for all domains. In the future, we plan to integrate our source selection method with feature selection and feature extraction approaches, such as deep neural networks (Long *et al.*, 2015). By adaptively finding an optimized feature space in which the target and source domains maximize their similarity, the usage of more complex features should become feasible.

Acknowledgements

This work was supported by the German Science Foundation (DFG) under grants OS 295/4-1 and HE 1822/30-1. 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.

Appendix: Proof for the Relation to Determine N_{max}

Theorem 1: Given a statistically independent sample $X=(x_i)_{i=1}^{N}$ *from a distribution defined by its cumulative distribution function* Pr($x<s$). *Let* q=Pr($x\geq s$) *be the probability that x is at least as large as a given value s. Also, let* p=1–Pr($max_{\in X}x\geq s$) *be the probability that the largest element in a set X is smaller than s. Then, for a fixed p and q the relationship* $N\geq log_{1-q}p$ *holds.*

Proof. Given

$$q = \Pr(x \geq s) \tag{28}$$

$$p = 1 - \Pr\left(\max_{x\in X} x \geq s\right) = \Pr\left(x < s \forall x \in X\right) = \left(1-q\right)^{N} \tag{29}$$

It follows

$$1 - \Pr\left(\max_{x\in X} x \geq s\right) \leq p' \Leftrightarrow \left(1-q\right)^{N} \leq p' \Leftrightarrow N \geq \log_{1-q}\left(p'\right) \tag{30}$$

References

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

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

Banerjee, B., F. Bovolo, A. Bhattacharya, L. Bruzzone, S. Chaudhuri, and K. Buddhiraju, 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): 4045–4062.

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

Bishop, C.M., 2006. Pattern Recognition and Machine Learning, First edition, Springer, New York.

Breiman, L., 2001. Random forests, Machine Learning, 45(1):5–32.

Bruzzone, L., and M. Marconcini, 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):1108–1122.

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

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

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

Chen, Y., M. Welling, and A. Smola, 2012. Super-samples from kernel herding, arXiv preprint arXiv:1203.3472.

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

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

Eaton, E., M. desJardins, and Y. Lane, 2008. Modeling transfer relationships between learning tasks for improved inductive transfer, Proceedings of the European Conference on Machine Learning (ECML), Springer, pp. 317–332.

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

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

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

Long, M., Y. Cao, J. Wang, and M.I. Jordan, 2015. Learning transferable features with deep adaptation networks, Proceedings of the 32nd International Conference on Machine Learning (ICML), pp. 97–105.

Matasci, G., D. Tuia, and M. Kanevski, 2012. SVM-based boosting of active learning strategies for efficient domain adaptation, IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing 5(5):1335–1343.

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

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

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

Press, W.H., 2007. Numerical Recipes: The Art of Scientific Computing, Third edition, Cambridge University Press, Cambridge, UK.

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

Settles, B., 2010. Active Learning Literature Survey, University of Wisconsin, Madison, Computer Sciences Technical Report 1648.

Shestopaloff, Y.K., 2010. Sums of Exponential Functions and Their New Fundamental Properties, AKVY Press, Toronto, Canada.

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

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

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

Thrun, S. and L. Pratt, 1998. Learning to Learn: Introduction and Overview, (S. Yhrun and L. Pratt, editors), Kluwer Academic Publishers, Boston, Massachusetts, pp. 3–17.

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

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

Tuia, D., E. Pasolli, and W.J. Emery, 2011. Using active learning to adapt remote sensing image classifiers, Remote Sensing of Environment, 115:2232–2242.

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

Vogt, K., A. Paul, J. Ostermann, F. Rottensteiner, and C. Heipke, 2017. Boosted unsupervised multisource selection for domain adaptation, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1:229–236.

Wegner, J.D., F. Rottensteiner, M. Gerke, and G. Sohn, 2016. The ISPRS 2D Labeling Challenge, URL: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (last date accessed: 15 January 2018).

Zadrozny, B., 2004. Learning and evaluating classifiers under sample selection bias, Proceedings of the 21st International Conference on Machine Learning (ICML), pp. 114–121.

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

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