ITERATIVE RE-WEIGHTED INSTANCE TRANSFER FOR DOMAIN ADAPTATION

A. Paul*, F. Rottensteiner, C. Heipke

Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany
(paul, rottensteiner, heipke)@ipi.uni-hannover.de

Commission III, WG III/4

KEY WORDS: Transfer Learning, Domain Adaptation, Logistic Regression, Machine Learning, Knowledge Transfer, Remote Sensing

ABSTRACT:

Domain adaptation techniques in transfer learning try to reduce the amount of training data required for classification by adapting a classifier trained on samples from a source domain to a new data set (target domain) where the features may have different distributions. In this paper, we propose a new technique for domain adaptation based on logistic regression. Starting with a classifier trained on training data from the source domain, we iteratively include target domain samples for which class labels have been obtained from the current state of the classifier, while at the same time removing source domain samples. In each iteration the classifier is re-trained, so that the decision boundaries are slowly transferred to the distribution of the target features. To make the transfer procedure more robust we introduce weights as a function of distance from the decision boundary and a new way of regularisation. Our methodology is evaluated using a benchmark data set consisting of aerial images and digital surface models. The experimental results show that in the majority of cases our domain adaptation approach can lead to an improvement of the classification accuracy without additional training data, but also indicate remaining problems if the difference in the feature distributions becomes too large.

1. INTRODUCTION

Supervised classification of images and derived data for automatic information retrieval is an important topic in photogrammetry and and remote sensing. One problem related to the machine learning techniques used in this context is the necessity to provide a sufficient amount of representative training data. Whereas the use of training data allows such methods to adapt to the specific distributions of features in varying scenes, the acquisition of training data, in particular the generation of the class labels of the training samples, is a tedious and time-consuming manual task. Applying a trained classifier to another image than the one from which the training data were generated reduces the amount of manual labour, but this strategy is also very likely to give sub-optimal results. This is due to the fact that in the new image the features may follow a different distribution than in the original one, so that the assumption of the training data being representative for the data to be classified is no longer fulfilled. The question of how a classifier trained on one data set can be of help in another learning task is dealt with in approaches for *Transfer Learning* (*TL*) (Thrun and Pratt, 1998; Pan and Yang, 2010). In TL, one tries to adapt a classifier trained on samples from a *source domain* to data from a *target domain*. These domains may be different, but they have to be related for this type of transfer to be possible. There are different settings for the TL problem; in the context of the classification of remote sensing images we are mostly interested in the case where labelled training data are only available for the source domain, which is related to the *transductive transfer learning* paradigm.

In this paper, we address one specific setting of transductive transfer learning called *domain adaptation* (*DA*) in which the source and the target domains are supposed to differ by the marginal distributions of the features used in the classification process, e.g. (Bruzzone and Marconcini, 2009). The particular application we are interested in is the pixel-based classification of images and Digital Surface Models (DSM). We use multiclass logistic regression (Bishop, 2006) for classification. The classifier is trained on an image for which training data are available and which corresponds to the source domain in the TL framework. When a new image is to be classified, the classifier is iteratively adapted to the distribution of the features in that image, which, thus, corresponds to the target domain. This DA process is based on an iterative replacement of training samples from the source domain by samples from the target domain which receive their class labels (*semi-labels*) from the current version of the classifier. Our method is inspired by (Bruzzone and Marconcini, 2009), but it uses logistic regression rather than Support Vector Machines (SVM) as a base classifier, which is supposed to make it faster in training and classification. An initial version of our approach was found to have considerable problems in case of strong overlaps between the feature distributions of different classes (Paul et al., 2015). In this paper we expand this method so that it becomes more robust with respect to overlapping feature distributions, and we evaluate the new method using a subset of the ISPRS 2D semantic labelling challenge (Wegner et al., 2016).

This paper is organized as follows. Section 2 gives an overview on related work in transfer learning in the framework of DA, with a focus on applications in remote sensing. In Section 3 we describe our new methodology for DA, whereas Section 4 presents our experimental evaluation. We conclude the article with a summary and an outlook on future work in Section 5.

2. RELATED WORK

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 $X \in \mathcal{X}$. In TL, we consider two domains, the source domain $\mathcal{D}_S$ and the target domain $\mathcal{D}_T$. Given a specific domain $\mathcal{D}$, a task $\mathcal{T} = \{\mathcal{C}, f(\cdot)\}$ consists of a label space $\mathcal{C}$, representing object *classes*, and a predictive function $f(\cdot)$. This function can be learned from the training data $\{\mathbf{x}_i, C_i\}$, where $\mathbf{x}_i \in X$ and $C_i \in \mathcal{C}$. Here again, a distinction is made between a source task $\mathcal{T}_S$ and a target task $\mathcal{T}_T$. In (Pan and Yang, 2010), TL

*Corresponding author

doi:10.5194/isprsannals-III-3-339-2016

is defined as a procedure that helps to learn the predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_T$ and $\mathcal{D}_S$, where either the domains or the tasks, or both, are different but related. There are three settings of TL (Pan and Yang, 2010). In *inductive TL*, the domains are assumed to be identical, but the tasks to be solved in these domains are different ($\mathcal{D}_S = \mathcal{D}_T, \mathcal{T}_S \neq \mathcal{T}_T$). In contrast, in the *transductive TL* setting, the tasks are assumed to be identical, but the domains may be different ($\mathcal{D}_S \neq \mathcal{D}_T, \mathcal{T}_S = \mathcal{T}_T$). In the case of *unsupervised TL*, both, the tasks and the domains may be different ($\mathcal{D}_S \neq \mathcal{D}_T, \mathcal{T}_S \neq \mathcal{T}_T$). For a thorough review of TL techniques, refer to (Pan and Yang, 2010). We are concerned primarily with the transductive setting and more specifically with DA techniques for remote sensing not requiring training data in the target domain.

According to Bruzzone and Marconcini (2009), one can distinguish two TL scenarios in which the distributions of the features used for training (source domain) and for testing (target domain) do not match. In the first scenario, the two domains and, thus, the distributions are considered to be identical, but the training data are not representative and do not allow a sufficiently good estimation of the joint distribution of the data and the classes. Depending on the nature of the differences between the estimated distributions and the true ones, this problem is referred to as *sample selection bias* (Zadrozny, 2004) or *covariate shift* (Sugiyama et al., 2007). It is the second scenario called *domain adaptation* we are interested in. Here, the source and target data are drawn from different domains, and the two domains differ in the marginal distributions of the features and posterior class distributions, thus $P(X_S) \neq P(X_T)$ and $P(C_S|X_S) \neq P(C_T|X_T)$ (Bruzzone and Marconcini, 2009). Contsequently, techniques for importance estimation such as (Sugiyama et al., 2007) can no longer be applied. Note that this definition of DA, which is adopted in this paper, is different from (Pan and Yang, 2010), where in the DA scenario the posteriors are assumed to be identical. From the point of view of our application, DA corresponds to a problem where the training (source domain) data are extracted from another image than the test (target domain) data in which the distribution of the features and the class posteriors are different, e.g. due to different lighting conditions or seasonal effects. Finding a solution to the TL problem in this scenario implies that one can transfer a classifier trained on one image data set to a set of similar images (i.e. to a related domain in the context of DA) without having to define training data in the new images.

There are two groups of DA methods which can be differentiated according to what is actually transferred (Pan and Yang, 2010). The first group of methods is based on *feature representation transfer*. Such methods try to find feature representations that allow a simple transfer from the source to the target domain, e.g. (Gopalan et al., 2011). The purpose is to obtain a set of shared and invariant features, for which the differences in the marginal and joint distributions between the two domains are minimized. Once this mapping has been established, the feature samples from both domains can be transferred to the joint representation, thus allowing the application of the classifier trained on source data in the transformed domain without any adaptation. An unsupervised feature transfer method based on feature space clustering and graph matching is proposed in (Tuia et al., 2013). Experiments based on synthetic and real data show good results. However, graph matching relies on an initial cross-domain graph containing all possible matches between cluster centroids in the two domains, and the authors conclude that their method might not work if the correct matches are not contained in that graph. In (Tuia et al., 2014), graph matching is expanded for so-called semi-supervised manifold alignment, which leads to an improved classification performance. However, this method requires labelled samples from all domains to provide some supervision for the graph matching process. In (Tuia, 2014), this requirement is relaxed under the assumption that the images have a certain spatial overlap, in which case one can identify corresponding points (*semantic tie points*) which provide the required labels across domains. However, spatial overlap is a relatively strong prerequisite that is not met in our application. Another approach for feature representation transfer based on graph matching is proposed in (Banerjee et al., 2015). The method can also deal with different class structures in the two images. However, experiments are only presented for multitemporal data sets of the same image region; the authors' claim that their method can also be applied in other settings is not supported by an empirical evaluation. The semi-supervised method for DA developed in (Cheng and Pan, 2014) uses linear transformations characterised by a set of rotation matrices for feature representation transformation. However, it also requires a small amount of representative labels from the target domain.

The second group of methods for DA is based on *instance transfer*. These methods successively replace training data from the source domain by data from the target domain. The classifier is adapted to the distribution of the data in the target domain, e.g. by weighing training samples with a probability ratio of data from the source and target domains (Zhang et al., 2010). However, the approach of Zhang et al. (2010) only deals with binary problems and other applications than image classification. In the context of remote sensing, an unsupervised retraining technique for a Gaussian maximum likelihood classifier is presented in (Bruzzone and Prieto, 2001). This method was evaluated on two images of the same area from different epochs. The distribution of the data of the target domain (the second epoch) is assumed to be a mixture of Gaussians whose components correspond to the individual classes. These components are initialized using parameters learned by training on source domain data, and their optimal parameters are determined using expectation maximisation (EM), which, thus, is the basis of transfer. Such a generative model is supposed to require more training data than discriminative classifiers (Bishop, 2006). In (Acharya et al., 2011), a discriminative classifier is trained on the basis of the source domain. The result is combined with the results of several clustering algorithms in order to obtain improved posterior probabilities for the target domain data based on the assumption that the data points of a cluster in feature space probably belong to the same class. A DA method using instance transfer based on a SVM has been presented in (Bruzzone and Marconcini, 2009). After training the SVM using source domain data, feature vectors from the target domain are iteratively added to the set of training samples, their semi-labels being based on the current state of the classifier, while other feature vectors are deleted from the source domain. The SVM is retrained after each iteration. The method shows good adaptation behaviour and it is shown to be superior to the one of Bruzzone and Prieto (2001). 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. However, although incremental training methods exist (Cauwenberghs and Poggio, 2001), SVM training is considered to be relatively slow (Abe, 2006), in particular in a multi-class setting, and requires a careful tuning of hyperparameters. In (Kuznetsova et al., 2015), a method for incremental learning for object detection in videos is presented. In this context, the classifier is adapted incrementally based on unlabelled data as the videos are processed, using the assumption of temporal consistency to find more semi-labelled samples. However, this method is not designed for the classification of an entire image. In (Amini and Gallinari, 2002), logistic regression is used in a semi-supervised setting, in which classification is combined with clustering of unlabelled data. Training is based on EM, and the semi-labels of the unlabelled data are determined according to the cluster membership of EM. Unlike our approach, this method assumes the labelled and the unlabelled data to follow the same distribution, so that no labelled training data are excluded in the training process.

In our previous work (Paul et al., 2015) we proposed a method for instance transfer that was inspired by Bruzzone and Marconcini (2009). We used a discriminative probabilistic classifier of lower computational complexity, which should also require fewer training samples than a generative approach. We followed the same strategy of gradually replacing source training samples by target samples, but using logistic regression as the base classifier. This required a new strategy for deciding which source training samples are to be eliminated from the training data and which samples from the target domain are to be added into the current training data set. We proposed two different strategies for that purpose. However, both had problems with overlapping feature distributions. One of the main reasons for failure was the binary nature of the inclusion of target samples into or the removal of source features from the training data set in the DA process. In particular, the inclusion of a training sample having a high impact on the decision boundaries (a *leverage point*) could lead to a sudden change in the decision boundary, which made the DA procedure unstable. In order to overcome this problem, we expanded our methodology so that it can consider individual weights in the DA process, modulating the impact on the basis of a sample's distance from the decision boundary. This allows the exclusion of uncertain feature samples by setting their weights to lower values and thus to avoid drifting of our model parameters. Additional stability is achieved by using the current state of the classifier for regularisation rather than a generic prior that assumes the expected value of the parameters to be zero. This regularisation method is inspired by Kuznetsova et al. (2015), where, however, it is used in a different context. Unlike in (Paul et al., 2015), we use real data including real changes in the feature distributions for the evaluation of our new method in order to assess its potential, but also its limitations.

3. METHODOLOGY

In this section we describe our new method for TL based on multiclass logistic regression. We start with the theory of logistic regression based on (Bishop, 2006), before presenting our approach for domain adaptation in section 3.2.

3.1 Logistic Regression

Logistic regression is a discriminative probabilistic classifier that directly models the posterior probability $P(C \mid \mathbf{x})$ of the class labels C given the data $\mathbf{x}$. In the multiclass case we distinguish K classes, i.e. $C \in \mathcal{C} = \{C^1, \ldots, C^K\}$. A feature transformation into a higher-dimensional space is applied to achieve non-linear decision boundaries. That is, logistic regression is applied to a vector $\Phi(\mathbf{x})$ whose components are functions of $\mathbf{x}$ and whose dimension is typically higher than the dimension of $\mathbf{x}$. The first element of $\Phi(\mathbf{x})$ is assumed to be a constant with value 1 for simpler notation of the subsequent equations. In the multiclass case, the model of the posterior is based on the softmax function:

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

where $\mathbf{w}_k$ is a vector of weight coefficients for a particular class C^k. As the sum of the posterior over all classes has to be 1, these weight vectors are not independent. This is considered by setting the first weight vector $\mathbf{w}_1$ to $\mathbf{0}$.

The parameters to be determined in training are the weights $\mathbf{w}_k$ for all classes except C^1, which can be collected in a parameter vector $\mathbf{w} = (\mathbf{w}_2^T, \ldots, \mathbf{w}_K^T)^T$. For that purpose, a training data set, denoted as $\overline{TD}$, is assumed to be available. It consists of N training samples $(\mathbf{x}_n, C_n)$ with $n \in \{1, \ldots, N\}$, where $\mathbf{x}_n$ is a feature vector and C_n its corresponding class label. In addition, we define a weight g_n for each training sample. In the standard setting, we use $g_n = 1 \; \forall n$, but in the DA process, the training samples will receive individual weights (cf. Section 3.2). Training is based on a Bayesian estimation of these parameters, determining the optimal values of $\mathbf{w}$ given $\overline{TD}$, by optimising the posterior (Vishwanathan et al., 2006; Bishop, 2006):

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

where $p\left(C = C^k|\mathbf{x}_n, \mathbf{w}\right)$ is defined in equation 1. The indicator variable t_{nk} takes the value 1 if the class label C_n of training sample n is C^k and zero otherwise. Compared to the standard model for multiclass logistic regression, the only difference is the use of the weights g_n in the exponent, which can be motivated in an intuitive way by the interpretation of the weights as an indicator for multiple instances of the same training sample, although we do not use integer values for these weights. Maximising the posterior in equation 2 is equivalent to minimising the negative logarithm $E(\mathbf{w})$ of the posterior:

$$E(\mathbf{w}) = -\sum_{n,\, k} g_n \cdot t_{nk} \cdot ln(y_{nk}) + \frac{(\mathbf{w} - \bar{\mathbf{w}})^T \cdot (\mathbf{w} - \bar{\mathbf{w}})}{2 \cdot \sigma^2}, \qquad (3)$$

where we use the short-hand $y_{nk} = p\left(C = C^k|\mathbf{x}_n, \mathbf{w}\right)$. The second term models the prior $p(\mathbf{w})$ as a Gaussian with mean $\bar{\mathbf{w}}$ and standard deviation σ, used to avoid overfitting (Bishop, 2006).

We use the Newton-Raphson method for finding the minimum of $E(\mathbf{w})$. Starting from initial values $\mathbf{w}^0 = \bar{\mathbf{w}}$, the updated parameters $\mathbf{w}^\tau$ in iteration τ are estimated according to:

$$\mathbf{w}^\tau = \mathbf{w}^{\tau-1} + \mathbf{H}^{-1} \nabla E(\mathbf{w}^{\tau-\mathbf{1}}), \qquad (4)$$

where $\nabla E(\mathbf{w}^{\tau-\mathbf{1}})$ and $\mathbf{H}$ are the gradient and the Hessian matrix of $E(\mathbf{w})$, respectively, both evaluated at the values from the previous iteration, $\mathbf{w}^{\tau-\mathbf{1}}$. The gradient is the concatenation of all derivatives with respect to the class-specific parameter vectors $\mathbf{w}_k$ (Vishwanathan et al., 2006; Bishop, 2006), i.e. $\nabla E(\mathbf{w}) = \left[\nabla_{\mathbf{w}_2} E(\mathbf{w})^T, \ldots, \nabla_{\mathbf{w}_K} E(\mathbf{w})^T\right]^T$, with

$$\nabla_{\mathbf{w}_k} E(\mathbf{w}) = \sum_{n=1}^{N} g_n \cdot (y_{nk} - t_{nk}) \cdot \Phi(\mathbf{x}_n) + \frac{1}{\sigma^2} \cdot (\mathbf{w}_k - \bar{\mathbf{w}}_k)\,. \qquad (5)$$

The Hessian matrix $\mathbf{H} = \nabla_{\mathbf{w}_k} \nabla_{\mathbf{w}_j} E(\mathbf{w})$ consists of $(K-1) \times (K-1)$ blocks $\mathbf{H}_{jk}$ (Vishwanathan et al., 2006; Bishop, 2006):

$$\mathbf{H}_{jk} = -\sum_{n=1}^{N} \left[g_n \cdot y_{nk} \cdot (I_{kj} - y_{nj}) \cdot \Phi(\mathbf{x}_n) \cdot \Phi(\mathbf{x}_n)^T\right] + \qquad (6)$$
$$+ \frac{\delta(j = k)}{\sigma^2} \cdot \mathbf{I},$$

where $\mathbf{I}$ is a unit matrix with elements I_{kj} and $\delta(\cdot)$ is the Kronecker function delivering a value of 1 if its argument is true and 0 otherwise. The iterative scheme according to equation 4 is repeated until the norm of $\nabla E(\mathbf{w})$ is numerically equal to zero.

3.2 Transfer Learning

To formally state our problem, we need to define our source domain data set $DS_S = \{(\mathbf{x}_{S_n}, C_{S_n}, g_{S_n})\}_{n=1}^{N_S}$, which contains N_S labelled samples $\mathbf{x}_{S_n}$ and the target domain data set $DS_T = \{(\mathbf{x}_{T_n}, g_{T_n})\}_{n=1}^{N_T}$ containing N_T unlabelled samples $\mathbf{x}_{T_n}$. This definition includes the weights g for both source and target data. Our aim is to transfer the initial classifier trained on source domain data to the target domain in an iterative procedure. For that purpose we have to adapt the current training data set $\overline{TD}$ with $\overline{TD} = \{(\mathbf{x}_{\overline{TD}_n}, C_{\overline{TD}_n}, g_{\overline{TD}_n})\}_{n=1}^{N_{\overline{TD}}}$, which is used to train a logistic regression, in each iteration.

We start with the initial training set $\overline{TD}^0 = DS_S$ to train our initial classifier. For that purpose we set the expected value of the model parameters for regularisation to zero, thus $\bar{\mathbf{w}} = \mathbf{0}$, and use a relatively loose setting σ_0 for the standard deviation σ of the Gaussian prior in equation 3. In this initial training stage, all weights are set to 1, thus $g_{\overline{TD}_n} = 1 \; \forall \; n$.

After the initial training on the source data in each further iteration i of domain adaptation, a predefined number ρ_E of source samples is removed from and a number ρ_A of target domain samples is included into the current training data set. Thus, in iteration i, the current training data set $\overline{TD}^i$ consists of a mixture of N_R^i source samples and N_L^i target samples:

$$\overline{TD}^i = \left\{ \{(\mathbf{x}_{S_r}, C_{S_r}, g_{S_r})\}_{r=1}^{N_R^i} \cup \{(\mathbf{x}_{T_l}, \widetilde{C}_{T_l}, g_{T_l})\}_{l=1}^{N_L^i} \right\}. \tag{7}$$

In equation 7, the symbol $\widetilde{C}_{T_l}$ denotes the *semi-labels* of the target samples, which are determined automatically (Section 3.2.2). As i is increased, N_R^i becomes smaller and N_L^i increases, until finally, only target samples with semi-labels are used for training, thus $\overline{TD}^{i_{end}} = \{(\mathbf{x}_{T_l}, \widetilde{C}_{T_l}, g_{T_l})\}_{n=1}^{N_T^{'}}$ with $N_T^{'} \leq N_T$. The criteria for selecting the source samples to be removed and the target samples to be included in iteration i are described in Sections 3.2.1 and 3.2.2, respectively.

Once the set of samples in $\overline{TD}^i$ has been defined, the weights $g_{\overline{TD}_n}$ have to be determined. The weights are defined in a way to avoid sudden changes in the decision boundaries that could cause the DA procedure to diverge, a problem that we encountered in our previous work (Paul et al., 2015). Our strategy for defining these weights is described in Section 3.2.3. Another measure to avoid too sudden changes in the decision boundaries between two subsequent iterations $i - 1$ and i of the DA process is to use the final weight vector $\mathbf{w}^{\tau_{end}, i-1}$ of the previous iteration for regularisation, that is $\bar{\mathbf{w}}^i = \mathbf{w}^{\tau_{end}, i-1}$. As the weight vector of the previous iteration is considered to provide better prior information than the generic model $\bar{\mathbf{w}} = \mathbf{0}$ used in the initial training in the source domain, the standard deviation σ of the Gaussian prior in equation 3 is set to a value $\sigma_{DA} < \sigma_0$. This type of regularisation is adapted from Kuznetsova et al. (2015), and it leads to a smoother adaptation of the parameters.

Having defined the current training data set $\overline{TD}^i$, the weights and the expected value of the prior, we use these data to retrain the logistic regression classifier. This leads to an updated weight vector $\mathbf{w}^{\tau_{end}, i}$ 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. In this manner, we gradually adapt the classifier to the distribution of the target domain data.

3.2.1 Criterion for source sample selection:

We use a criterion related to the distance of a sample from the decision boundary in the transformed feature space $\Phi(\mathbf{x})$ in order to select source samples to be removed from the set $\overline{TD}^i$. As the criterion is only used for ranking, we can use the posterior according to equation 1, increasing with the distance from the decision boundary. Basically, we would like first to eliminate source samples that are as far away as possible from the current decision boundary. The rationale behind this choice is that these samples have relatively low influence on decision boundaries between the classes, which supports a smooth shift of the transition boundaries. However, we have to consider the case that some source samples may be situated on the wrong side of the decision boundary, i.e., there may be source samples whose class labels C_{S_r} are inconsistent with the class labels $C_{S_r}^{LR}$ obtained by applying the current version of logistic regression classifier to that sample. Thus, our criterion d_B for ranking is defined as follows:

$$d_B = \begin{cases} p(C = C_{S_r} | \mathbf{x}_{S_r}) & \text{if } C_{S_r} = C_{S_r}^{LR} \\ 2 - p(C = C_{S_r} | \mathbf{x}_{S_r}) & \text{if } C_{S_r} \neq C_{S_r}^{LR} \end{cases}. \tag{8}$$

We rank all source samples remaining in $\overline{TD}^i$ by d_B separately for all classes and eliminate ρ_E samples having the highest values of d_B. Using the definition according to equation 8 eliminates inconsistent source samples first, starting with the samples being most distant from the decision boundary. As soon as all inconsistent samples in an iteration i have been eliminated, the procedure continues with the consistent samples that are most distant from the decision boundary.

3.2.2 Criterion for target sample selection:

Here we adapt a criterion based on the distance to other training samples which was found to work best for overlapping feature distributions in our previous work (Paul et al., 2015). For each candidate sample for inclusion into $\overline{TD}^i$ from the target domain we select its k nearest neighbours in the transformed feature space $\Phi(\mathbf{x})$ among the training samples in $\overline{TD}^i$ independently from their class labels. For efficient nearest neighbour search we apply a kd-tree as a spatial index. We determine the average distance $d_{a_{knn}}$ of the candidate sample from its k nearest neighbours and we determine a class label C^{max_k} corresponding to the class label occurring most frequently among its neighbours. We also predict the most likely class label C^{LR} using the current state of the logistic regression classifier and do not consider samples for which $C^{LR} \neq C^{max_k}$ in the current transfer step, because this indicates a high uncertainty of the predicted class label (which could lead to the inclusion of candidate samples with wrong labels into $\overline{TD}^i$). Thus, the score function used to rank all target samples not yet contained in $\overline{TD}^i$ with $C^{max_k} = C^{LR}$ becomes:

$$D = d_{a_{knn}}. \tag{9}$$

We sort all samples in ascending order according to D, and we select the ρ_A samples having the best (i.e., smallest) score for inclusion into the set $\overline{TD}^i$, ignoring samples with $C^{LR} \neq C^{max_k}$, as stated previously. These target samples included in $\overline{TD}^i$ are removed from the list of available target domain samples. Note that in this step we also check all the semi-labels of target samples in $\overline{TD}^{i-1}$. If the semi-label is found to be inconsistent with the output of the current state of the classifier, it will be changed. This allows the target samples to change their semi-labels depending on the current position of the decision boundary. The score function according to equation 9 again prevents the decision boundary from changing too abruptly in the DA process, relating the selection criterion to the density of samples in feature space.

3.2.3 Weights:

In each iteration i, we have to define a weight $g_{\overline{TD}_n} \in [0, 1]$ for all training samples in $\overline{TD}^i$. The weight indicates the algorithm's trust in the correctness of the label of a training sample and, thus, first of all depends on whether the sample is from the source or target domain. For source domain samples having consistent class labels (i.e., $C_{S_r} = C_{S_r}^{LR}$), the weight is set to 1, indicating maximum trust into that sample. For all other samples, we need a weight function that can reduce the influence of potential outliers. In this context, the samples most likely to be affected by errors (i.e., wrong training labels) are source samples with inconsistent class labels and target samples that are close to the current decision boundary. In both cases, the posterior according to equation 1 is a good indicator for such a situation: for inconsistent source samples, it is smaller the further away the sample is from the decision boundary. Given our method for defining the semi-labels (cf. Section 3.2.2) there are no inconsistent target samples, but the posterior monotonically increases with the sample's distance from the current decision boundary. Denoting the posterior by the shorthand $p_n = p(C = C_{S_n}|\mathbf{x}_{\overline{TD}_n})$ for source samples and $p_n = p(C = \widetilde{C}_{T_l}|\mathbf{x}_{\overline{TD}_n})$ for target samples, we need to find a monotonically increasing function for $g(p_n)$ to define the weights. We use a variant of the weight function proposed in (Klein and Förstner, 1984) for modulating weights in the context of robust least squares adjustment (cf. figure 1):

$$g(p_n, h) = 1 - \frac{1}{1 + \left(\frac{p_n}{h}\right)^4}, \qquad (10)$$

where the parameter h is related to the steepness of the weight function. Using this function, we define the weight for a sample $\mathbf{x}_{\overline{TD}_n}$ in $\overline{TD}^i$ as follows:

$$g_{\overline{TD}_n}(\mathbf{x}_{\overline{TD}_n}) = \begin{cases} 1 & \text{if } \mathbf{x}_{\overline{TD}_n} \in DS_S \wedge C_{S_n} = C_{S_n}^{LR} \\ g(p_n, h) & \text{if } \mathbf{x}_{\overline{TD}_n} \in DS_S \wedge C_{S_n} \neq C_{S_n}^{LR} \\ g(p_n, h) & \text{if } \mathbf{x}_{\overline{TD}_n} \in DS_T \end{cases}. \qquad (11)$$

In equation 11, C_{S_n} and $C_{S_n}^{LR}$ denote the source labels from the training data set and the output of the current version of the classifier for that sample, respectively. This definition of the weight function puts more emphasis on target samples whose semi-labels are certain (as indicated by a large value of the posterior) while mitigating the impact of wrong semi-labels in the vicinity of the decision boundary. Thus, a new target sample close to the decision boundary becomes less likely to cause a sudden change in the decision boundary, and it becomes more likely to change its semi-label in subsequent iterations if the inclusion of other target samples indicates a different position of that boundary. We expect this weight function to increase the robustness of the entire DA process.

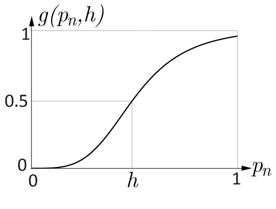


Figure 1: The weight function $g(p_n, h)$; note that $g(h, h) = 0.5$.

4. EXPERIMENTS

4.1 Test Data and Test Setup

For the evaluation of our method we use the Vaihingen data set from the ISPRS 2D semantic labelling contest (Wegner et al., 2016), acquired on 24 July 2008. The data set contains 33 patches of different size, each consisting of an 8 bit colour infrared true orthophoto (TOP) and a digital surface model (DSM) generated by dense matching, both with a ground sampling distance of 9 cm. For our tests, we use the 16 patches for which labels are made available by the organisers of the benchmark. As it is the goal of this experiment to highlight the principle of TL rather than to achieve optimal results, we only use two features, namely the normalized vegetation index (NDVI) and the normalized DSM (nDSM), the latter corresponding to the height above ground; the terrain height required for determining the nDSM was generated by morphologic opening of the DSM. Both features are scaled linearly into the interval $[0 \dots 1]$. The test data show a suburban scene with six object classes, namely *impervious surface*, *building*, *low vegetation*, *tree*, *car* and *clutter/background*. As there are very few samples for some of these classes and because for a proof-of-concept of our new method do not want to investigate more complex feature spaces, we merge the classes *impervious surface*, *clutter/background*, *low vegetation*, and *car* to a joint class *ground*.

In our experiments, each of the 16 image patches is considered to correspond to an individual domain. Consequently, there are 240 possible pairs of domains which we could use to test our DA approach. One patch of each pair constitutes the source domain, whereas the other one corresponds to the target domain. First, we used source domain samples to train a classifier and classified the pixels of the target domain without applying DA. This experiment is referred to as variant V_{ST}. Afterwards, we used target domain data for training and again classified the target domain (variant V_{TT}). This variant represents the best possible performance using logistic regression. For both variants, we compared the predicted labels of the target samples to the reference, determined the confusion matrices and derived quality metrics such as completeness, correctness and the overall accuracy OA as a compound quality measure, e.g. (Rutzinger et al., 2009). The difference in OA between variants V_{ST} and V_{TT} indicates the degree to which the classification accuracy deteriorates if a classifier trained on the source domain is applied to the target domain without adaptation. In the following experiments we concentrated on the 36 pairs of patches showing a loss in OA larger than 5%. For these 36 pairs, we additionally used our TL procedure and applied the transferred classifier to the target domain data (variant V_{TL}), again deriving quality metrics for comparison.

In these experiments, in each domain (each image patch) the source and target samples were selected in a regular grid with a spacing of 20 pixels, thus using 0.25% of the data for training and transfer (variants V_{ST}, V_{TT} and V_{TL}). This turned out to be a good compromise to extract meaningful information with a relatively small number of samples. We used a polynomial expansion of degree 2 for feature space mapping, as a tradeoff between simplification and overfitting avoidance. The weights for the Gaussian prior for regularisation were set to: $\sigma_0 = 50$ for training the initial classifier based on source domain data and $\sigma_{DA} = 25$ for the modified prior in the DA process. The number of samples per class for transfer and elimination were set to $\rho_E = \rho_A = 30$, which corresponds to $0.15\% - 0.40\%$ of the samples used for training or DA, depending on the patch size. We use $k = 19$ neigbours in the knn analysis for deciding which target samples to include for training. The parameter h of the weight function (equation 10) is set to $h = 0.7$. In Section 4.2 we report the results achieved for these parameter settings. A sensitivity analysis showing the impact of these parameters on the DA performance is presented in Section 4.3.

4.2 Evaluation

In this section we evaluate the results of our DA procedure based on the 36 pairs of patches selected in the way described in Sec-

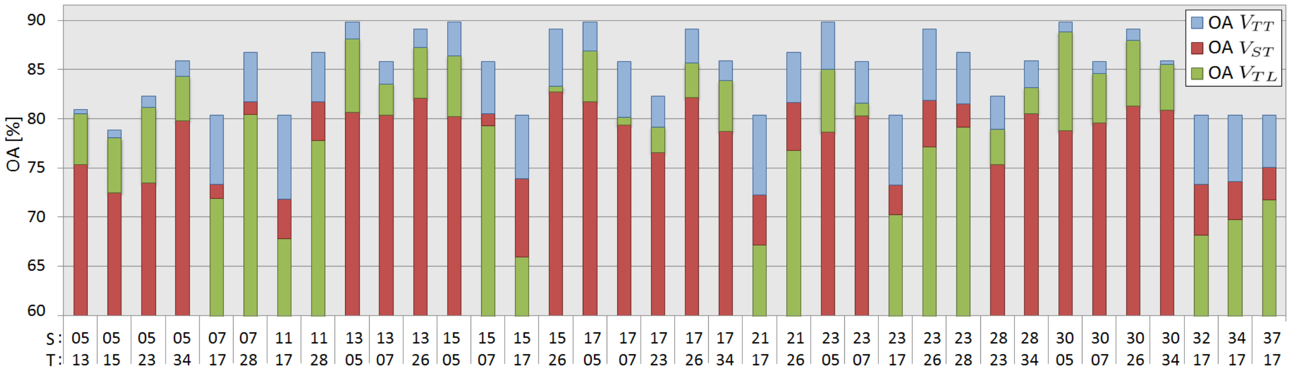
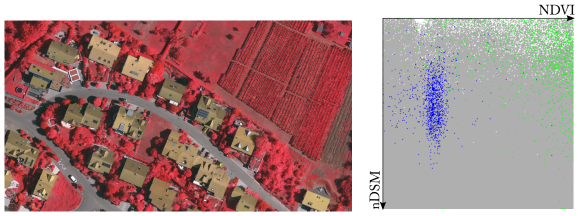


Figure 2: Overall accuracy [%] for the 36 test pairs, obtained for three classification variants (red: V_{ST}; blue: V_{TT}; green: V_{TL}). S: number of the patch corresponding to the source domain; T: number of the patch corresponding to the target domain.

tion 4.1. The OA achieved for all three variants is presented in figure 2. The red bars indicate the OA for variant V_{ST}, i.e. the application of a classifier trained on source domain data to the target domain without adaptation, whereas the blue bars correspond to the OA for variant V_{TT}, i.e. the case when the classifier was trained using target domain data. The green bars correspond to variant V_{TL}, i.e. to the results of DA.

Figure 2 shows that a positive transfer, indicated by a larger overall accuracy of variant V_{TL} compared to V_{ST}, could be achieved for 22 out of 36 patch pairs, i.e. in about 61% of the cases. For eight pairs (22% of the test set), more than 80% of the loss in classification accuracy between variants V_{ST} and V_{TT} could be compensated by our DA technique, for another nine pairs (25%) between 50% and 80% of the loss was compensated, and for two more pairs the compensation was at least larger than 30%. The average improvement in OA achieved by our DA method is 4.7% in the 22 cases where DA was successful. This is contrasted by 14 pairs (39% of the test cases) where a negative transfer occurred, indicated by a smaller overall accuracy of variant V_{TL} compared to V_{ST}. On average, for these 14 pairs the OA is decreased by 3.7% by our DA method. It is interesting to observe that eight of these 14 cases of negative transfer occur when image patch 17 corresponds to the target domain. The patch and the distribution of target features are shown in figure 3. We can observe a strong overlap of the distributions for the classes $ground$ and $tree$. This is caused by the fact that a large part of the scene is covered by a vineyard, which corresponds to class $ground$ in our classification scheme. Even in the variant V_{TT}, the separation of $ground$ and $tree$ is very uncertain, with a completeness for $tree$ of only 45%. After transfer learning, there tends to be even more confusion between $ground$ and $tree$ whereas the classification accuracy for $buildings$ is hardly affected. This is the only scene containing a vineyard. Thus, it would seem that the feature distribution of this area is too different from those of the other areas, so that the prerequisites for DA are not fulfilled here. Nevertheless, with 79.7% the average OA over all 36 test for the variant V_{TL} is 1.4% above the average OA for variant V_{ST} (78.3%). Thus, we also achieve a positive transfer over the whole data set. In order to fully exploit the benefits of DA it would be desirable to identify situations of negative transfer; this could be achieved by a circular validation strategy as pursued in (Bruzzone and Marconcini, 2009).

A general evaluation of completeness and correctness per class cannot be given here for lack of space. We present exemplary results for test pair $30/34$ (S/T) in figure 4 and give the corresponding quality measures per class in table 1. Both, the figure and the table show that the results after TL (Variant V_{TL}) correspond closely to what can be achieved if training samples from the target domain are used (Variant V_{TT}), while showing a considerable improvement over the variant without transfer (V_{ST}).



(a) Orthophoto (b) Feature space

Figure 3: Orthophoto and feature space for patch 17. Colours: *ground* (white), *building* (blue) and *tree* (green).

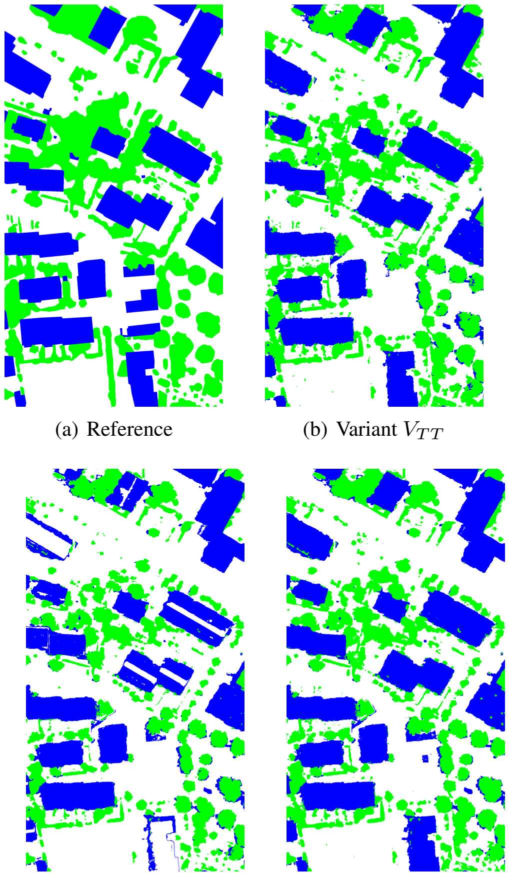
Variant		Class			OA
		$ground$	$build.$	$tree$	[%]
V_{TT}	$Comp.$ [%]	91.0	90.3	73.6	85.9
	$Corr.$ [%]	83.9	89.9	85.9	
	$Quality$ [%]	77.4	82.0	65.7	
V_{ST}	$Comp.$ [%]	94.4	78.3	60.7	80.9
	$Corr.$ [%]	74.8	87.4	92.0	
	$Quality$ [%]	71.7	70.3	57.7	
V_{TL}	$Comp.$ [%]	92.7	91.2	68.7	85.6
	$Corr.$ [%]	82.4	89.5	88.7	
	$Quality$ [%]	77.3	82.4	63.1	

Table 1: Overall accuracy (OA), completeness ($Comp.$), correctness ($Corr.$) and quality values for the classes $ground$, $building$ ($build.$) and $tree$, obtained for the three variants of the test (V_{TT}, V_{ST}, V_{TL}) in pair 30/34 (S/T).

4.3 Parameter sensitivity analysis

The influence of the three most important parameters in terms of TL quality and our proposed instance weighting approach are studied here to show the stability of the proposed methodology. In these experiments, one parameter is varied, whereas all other settings remain constant as described in Section 4.1. We evaluate the average OA over the 36 tests for the three variants described in the previous sections and additionally include the average OA achieved by DA in the cases of positive transfer (V_{TL}^{+}). The results are presented in figure 5.

The first parameter is the number of samples per class to transfer (figure 5(a)). Here we use $\rho_E = \rho_A = \rho$, i.e., the number of source samples to be removed from $\overline{TD}^i$ is identical to the number of target samples to be added in each iteration. The best average OA of V_{TL} was achieved for our standard setting with $\rho = 30$. Larger as well as lower values for ρ led to poorer average transfer performance. The value $\rho = 10$ even led to negative transfer measured in average OA of V_{TL} compared to V_{ST}. The good result for V_{TL}^{+} in case of $\rho = 10$ is not representative because the number of tests with positive transfer is very small. Thus, the value $\rho = 30$ is a good compromise between too slow



(a) Reference (b) Variant V_{TT}

(c) Variant V_{ST} (d) Variant V_{TL}

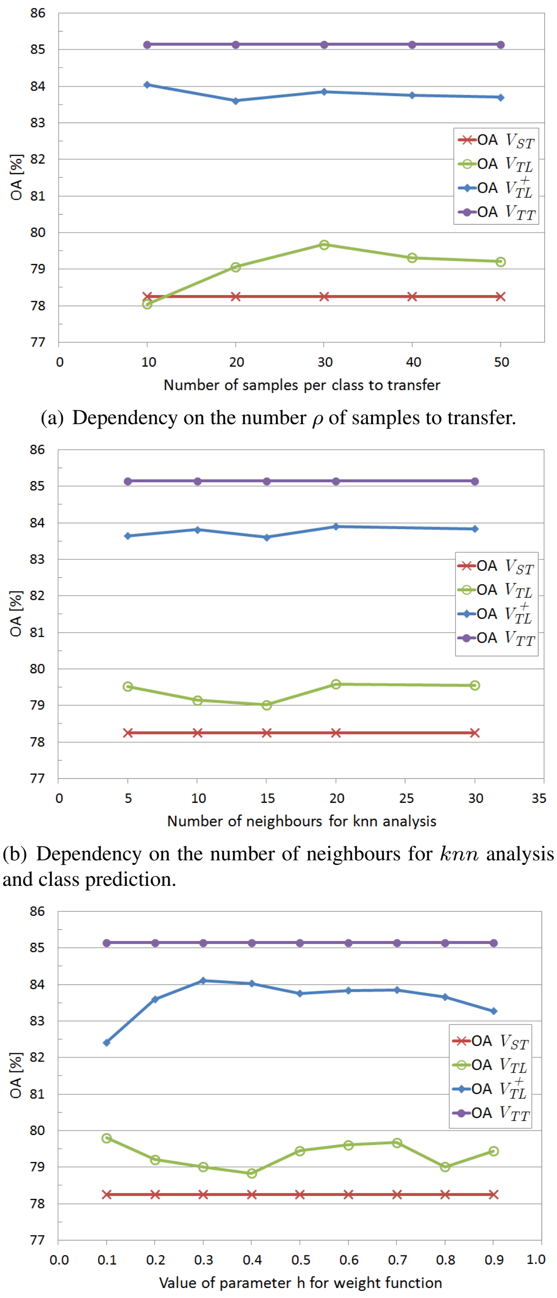
Figure 4: Reference data and results of classification of the target area for test pair $30/34$ for the three classification variants. Colours: *ground* (white), *building* (blue) and *tree* (green).

and too fast changes caused by removal of certain source samples from and adding semi-labeled target samples to the set $\overline{TD}^i$.

The second parameter is the number of neighbours for knn analysis and class prediction. The value of neighbours for knn analysis and class prediction for our standard case $k = 19$ seems to be a good choice, this led to slightly worse result in average OA of V_{TL} and V_{TL}^+ than the best case in this study with value $k = 20$. The changes of this parameter causes only small differences in average OA of V_{TL}. It would seem that the choice of this parameter is not crtitical for the quality of transfer.

The last parameter in our sensitivity analysis is the parameter h of the weight function in equation 10. The best results of average OA of V_{TL} are achieved by h values between 0.50 and 0.70, the best of which corresponds to our standard case $h = 0.70$. Somewhat surprisingly, using $h = 0.10$ also leads to a relatively good result in variant V_{TL}, though the average OA for the cases of positive transfer (V_{TL}^+) is worst in this case. This indicates that the relatively high total OA is due to a lower decrease in OA for the cases of negative transfer rather than to an improvement in the cases of positive transfer. The values $g(p_n, h = 0.1) \in [0.01..0.99]$ correspond to $p_n \in [0.03..0.33]$. As we have three classes, the maximum of the posterior will be larger than 1/3, so that for all target samples and for all consistent source samples, such low values for p_n will only occur with inconsistent source samples in the TL process (cf. Section 3.2.3 and equation 11). This shows that the proper handling of these source samples may have a relatively high influence on the quality of transfer.

All parameter changes during the tests result in quite stable changes of the OA for V_{TL} and V_{TL}^+, which confirms the stability and robustness of our methodology.



(a) Dependency on the number ρ of samples to transfer.

(b) Dependency on the number of neighbours for knn analysis and class prediction.

(c) Dependency on the parameter h of the weight function.

Figure 5: Dependency of the average overall accuracy (over 36 tests) on different parameter settings.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a methodology for domain adaptation based on logistic regression. Compared to our previous work, we consider individual weights in the DA process, use adapted strategies for selecting the samples for instance transfer, and apply a new regularisation technique. These changes lead to a higher robustness of the transfer process in the case of overlapping feature distributions. Experiments based on real data have shown that in the majority of the cases, the overall accuracy in the classification of the target domain can be increased considerably without additional training data. We have also identified critical cases in which the assumptions about the feature distributions were violated, leading to a negative transfer.

In the future, we will investigate alternative strategies for sample selection and weight definition that might further increase the transfer performance, and we want to investigate the impact of a higher-dimensional feature space. Furthermore, strategies for detecting cases of negative transfer automatically would allow us to fully exploit the benefits of DA. We will test our method on different data sets using a feature space of higher dimension and also differentiating more classes. We will also compare our method to other techniques, e.g. to semi-supervised classification as proposed in (Amini and Gallinari, 2002). Finally, we want to compare our methodology to a setting in which a small amount of labelled samples is available in the target domain.

ACKNOWLEDGEMENTS

This work was supported by the German Science Foundation (DFG) under grant HE 1822/30-1. The Vaihingen data were 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

Abe, S., 2006. Support Vector Machines for Pattern Classification. 2^{nd} edn, Springer, New York (NY), USA.

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 15^{th} 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.

Bishop, C. M., 2006. Pattern Recognition and Machine Learning. 1^{st} 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.

Bruzzone, L. and Prieto, D., 2001. Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 39(2), pp. 456–460.

Cauwenberghs, G. and Poggio, T., 2001. Incremental and decremental support vector machine learning. In: Advances in Neural Information Processing Systems 13, pp. 409–415.

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.

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.

Klein, H. and Förstner, W., 1984. Realization of automatic error detection in the block adjustment program PAT-M43 using robust estimators. International Archives of Photogrammetry and Remote Sensing XXV-3a, pp. 234–245.

Kuznetsova, A., Hwang, S. J., Rosenhahn, B. and Sigal, L., 2015. Expanding object detector's horizon: Incremental learning framework for object detection in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 28–36.

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., 2015. Transfer learning based on logistic regression. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-3/W3, pp. 145–152.

Rutzinger, M., Rottensteiner, F. and Pfeifer, N., 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2(1), pp. 11–20.

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.

Tuia, D., 2014. Weakly supervised alignment of image manifolds with semantic ties. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 3546–3549.

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.

Tuia, D., Volpi, M., Trolliet, M. and Camps-Valls, G., 2014. Semisupervised manifold alignment of multimodal remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 52(12), pp. 7708–7720.

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.

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.