CLASSIFICATION UNDER LABEL NOISE BASED ON OUTDATED MAPS

A. Maas, F. Rottensteiner, C. Heipke

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

Commission II, WG II/6

KEY WORDS: Supervised Classification, Label Noise, Logistic Regression, Map Updating

ABSTRACT:

Supervised classification of remotely sensed images is a classical method for change detection. The task requires training data in the form of image data with known class labels, whose manually generation is time-consuming. If the labels are acquired from the outdated map, the classifier must cope with errors in the training data. These errors, referred to as label noise, typically occur in clusters in object space, because they are caused by land cover changes over time. In this paper we adapt a label noise tolerant training technique for classification, so that the fact that changes affect larger clusters of pixels is considered. We also integrate the existing map into an iterative classification procedure to act as a prior in regions which are likely to contain changes. Our experiments are based on three test areas, using real images with simulated existing databases. Our results show that this method helps to distinguish between real changes over time and false detections caused by misclassification and thus improves the accuracy of the classification results.

1. INTRODUCTION

The updating of topographic databases (referred to as *maps* for brevity) is typically based on a classification of current remote sensing imagery. Comparing the results to the map, areas of change can be detected and the map can be updated accordingly. Supervised classification is commonly used for that purpose, requiring representative training data that are typically generated in a time-consuming manual process. The latter could be avoided by using the existing map to derive the class labels of the training samples. As the map may be outdated, classifiers using the class labels derived from the map for training must take into account the fact that some of these labels will be wrong. Nevertheless, changes typically only affect a relatively small part of a scene, so that one can assume the majority of the training data to be correct.

In machine learning, errors in the class labels of training data are referred to as *label noise* (Frénay and Verleysen, 2014). In remote sensing, the problem has mostly been dealt with by data cleansing, i.e. by detecting and eliminating wrong training samples, e.g. (Radoux et al., 2014). An alternative is to use probabilistic methods for training under label noise which also estimate the parameters of a noise model. An example for such an approach is the label noise tolerant logistic regression (Bootkrajang and Kabán, 2012), which has been applied successfully in the context of remote sensing in (Maas et al., 2016). However, the underlying noise model of that technique assumes wrong labels to occur at random positions in the image. This is not a very realistic model for change detection, where changes typically occur in clusters, e.g. due to the construction of a new building, and may lead to a degradation of the classification performance.

Using the existing map has another potential benefit. As change is usually a rare event, the existing class labels can be seen as providing observations for the prediction of the new class labels. This may be particularly useful in areas where the classifier cannot distinguish the class label by the given features well, e.g. at object borders. The corresponding probabilities for the classes to be correct are related to the probability of observing a wrong label and, thus, to the parameters of a probabilistic noise model (Bootkrajang and Kabán, 2012). However, such an assumption again neglects the fact that changes typically occur in compact clusters. It would typically lead to a strong bias for maintaining the class label of the map, which is desired in areas without changes, but may limit the prospects of detecting real changes.

In this paper, we propose a new supervised classification method that tries to extract as much benefit as possible from the availability of the existing map. Firstly, our method uses the class labels from the map for training. This is achieved by expanding the method by Bootkrajang and Kabán (2012) to take into account that changes typically occur in clusters, which we expect to improve the results in scenes with a large amount of change. Secondly, the class labels of the existing map are included as observations in a classification procedure based on Conditional Random Fields (CRF). We propose an iterative procedure to reduce the impact of the observed class labels in compact areas that are likely to have changed, which we expect to improve the classification results in areas of weak features without affecting the detection of real changes too much. For evaluation we use three data sets with different degrees of simulated changes.

2. RELATED WORK

Of the basic strategies for change detection identified in Jianya et al. (2008), we apply the one in which changes are inferred from differences between the independent classification of a current image and the existing map, because no sensor data are assumed to be available for the time of the acquisition of the existing map. Nevertheless, the existing map will be integrated into the classification process, which is also the focus of this paper.

For the reasons pointed out in Section 1, a training procedure taking the class labels of the training samples from an existing map must cope with label noise. Frénay and Verleysen (2014) differentiate three types of statistical models for label noise. The *noisy completely at random* (NCAR) model does not consider dependencies between label noise and other variables. In the *noisy at*

random (NAR) model, the probability of an error depends on the class label. If the dependencies between labelling errors and the observed data are considered, the model is called *noisy not at random* (NNAR). This would be an appropriate choice in our case to model that label noise typically occurs in clusters in image space. We do not build a NNAR model explicitly, but we use one implicitly by an iterative strategy for reducing the impact of training samples forming clusters of potentially changed pixels. Existing NNAR models tend to analyse the distributions of the training samples in feature space, e.g. assuming label noise to occur more likely near the classification boundaries or in low-density regions (Sarma and Palmer, 2004). Apart from being drawn from another domain than image classification, this is not a model of local dependencies between label noise at neighbouring data sites.

Frénay and Verleysen (2014) distinguish three strategies for dealing with label noise. First, classifiers that are robust to label noise by design can be used, e.g. random forests, but they still may have difficulties with large amounts of label noise (Maas et al., 2016). The second strategy tries to remove training samples affected by label noise from the training set. Such *data cleansing* methods have been criticised for eliminating too many instances (Frénay and Verleysen, 2014). The third option is to use a classifier which is tolerant to label noise. In this context, *probabilistic approaches* learn the parameters of a noise model along with the classifer in the training process; examples are (Bootkrajang and Kabán, 2012), using logistic regression as the base classifier, and (Li et al., 2007), presenting a method based on the kernel Fisher discriminant. An example for a *non-probabilistic approach* is the label noise tolerant version of a Support Vector Machine (SVM) (An and Liang, 2013). However, non-probabilistic methods typically do not estimate the parameters of a noise model, e.g. transition probabilities containing the probability for the observed label to be affected by a change (Bootkrajang and Kabán, 2012), which may be used as temporal transition matrices in our application, linking the observed class labels of the map to class labels at the second epoch (Schistad Solberg et al., 1996).

In the domain of remote sensing, classification under label noise seems to be based on data cleansing in most cases. An example is (Radoux et al., 2014), who include two techniques for eliminating outliers to derive training data from an existing map. The first technique removes training samples near the boundaries of land cover types and the other one removes outliers based on a statistical test, assuming a Gaussian distribution of spectral signatures. Designed for data of 300 m ground sampling distance (GSD), the model assumptions, e.g. Gaussian distributions, cannot be used directly for high resolution images. A similar method was used for map updating in (Radoux and Defourny, 2010), using Kernel density estimation for deriving probability densities. Another data cleansing method is reported in (Jia et al., 2014). Similarly to the method proposed in this paper, all pixels from an existing map are used for training and the resulting label image is compared to the existing map to detect changes. However, no parameters of a model for label noise are estimated in the training process. This is also true for the data cleansing method based on SVM reported in (Büschenfeld, 2013), who eliminate training samples that are assigned to another class than indicated by the given map or that show a high uncertainty. Label noise tolerant training using maps for deriving training data was done by Mnih and Hinton (2012). They propose two loss functions tolerant to label noise to train a deep neural network, but their method only deals with binary classification problems. Bruzzone and Persello (2009) include information of the pixels in the neighbourhood of the training samples in the learning process to achieve robustness to label noise in a context-sensitive semi-supervised SVM. Although the authors argue that such a strategy can be used to integrate existing maps for training, this is not shown explicitly.

In (Maas et al., 2016), we applied label noise tolerant logistic regression (Bootkrajang and Kabán, 2012) to use an existing map for training, integrating it into a CRF for context-based classification. The experiments showed that the method is tolerant to a large amount of label noise if it is randomly spread over the image, as would be expected for a method based on a NAR model. However, experiments with more realistic changes were only shown with a small percentage of wrong training labels, and the class labels from the existing map were not used in the classification process. The latter was done by Schistad Solberg et al. (1996), who applied a temporal model based on transition probabilities to include an outdated land cover map in multitemporal classification, but no local dependencies between changes were considered. In this paper we want to expand our previous work (Maas et al., 2016) by considering the fact that changes occur in clusters. Label noise logistic regression Bootkrajang and Kabán (2012) is applied in an iterative procedure in which the impact of training samples in areas of potential change is reduced, while these samples are not completely eliminated. To consider local context, the resultant classifier was integrated in a CRF, in which we also consider the original class labels as additional observations. In contrast to (Schistad Solberg et al., 1996), the influence of these observations may change in the course of an iterative process if a pixel is situated in a large cluster of potentially changed pixels, so that temporal oversmoothing (Hoberg et al., 2015) can be avoided. Our method can be seen as a combination of "soft" data cleansing (because samples are not eliminated completely) with a probabilistic noise model for including the observed labels from the map. Thus, we expect to be able to cope with a larger amount of real change than our previous method.

3. LABEL NOISE TOLERANT CHANGE DETECTION

We assume remotely sensed data and an existing but outdated raster map to be available on the same grid. The data consist of N pixels, each pixel n represented by a feature vector $\mathbf{x}_n = [x_n^1, ..., x_n^F]$ of dimension F, calculated from the imagery, and an observed class label $\widetilde{C}_n \in \mathbb{C} = \{C^1, ..., C^K\}$ from the existing map. $\mathbb{C}$ denotes the set of classes and K is the total number of classes. As the database may be outdated, the observed labels may differ from the unknown true labels $C_n \in \mathbb{C}$. Collecting the observed and the unknown class labels in two vectors $\widetilde{\mathbf{C}} = (\widetilde{C}_1, ..., \widetilde{C}_N)^T$ and $\mathbf{C} = (C_1, ..., C_N)^T$, respectively, and denoting the observed image data by $\mathbf{x}$, it is our goal to find the optimal configuration of class labels $\mathbf{C}$ by maximising the joint posterior $P(\mathbf{C}|\mathbf{x}, \widetilde{\mathbf{C}})$ of the unknowns given the observations. In this process, we use the class labels of the outdated map for deriving the class labels of the training samples. We start by outlining our modified version of the training procedure for logistic regression by Bootkrajang and Kabán (2012) (Section 3.1). In Section 3.2, we show how logistic regression is integrated into a CRF (Kumar and Hebert, 2006) together with a model for considering the existing class labels $\widetilde{\mathbf{C}}$ as observations. Section 3.3 describes the new iterative procedure for training and inference.

3.1 Label noise robust logistic regression

Classification is based on logistic regression, a discriminative probabilistic classifier that directly models the posterior probabylity $p(C_n|\mathbf{x}_n)$ of a class label C_n given the feature vector $\mathbf{x}_n$.

A feature space transformation $\mathbf{\Phi}(\mathbf{x}_n)$ may be applied to achieve non-linear decision boundaries in the original feature space. In the multiclass case the posterior is modelled by (Bishop, 2006):

$$p(C_n = C^k|\mathbf{x}_n) = \frac{exp\Big(\mathbf{w}_k^T \cdot \mathbf{\Phi}(\mathbf{x}_n)\Big)}{\sum_{j=1}^{K} exp\Big(\mathbf{w}_j^T \cdot \mathbf{\Phi}(\mathbf{x}_n)\Big)} \quad (1)$$

where $\mathbf{w}_k$ is a vector of parameters for a particular class C^k. As the sum of the posterior over all classes has to be 1, these parameter vectors are not independent, so that $\mathbf{w}_1$ is set to $\mathbf{0}$; the other vectors are collected in a joint parameter vector $\mathbf{w}$.

In our case, each training sample consists of a feature vector $\mathbf{x}_n$ and the observed label $\widetilde{C}_n$. In order to consider this fact in training, Bootkrajang and Kabán (2012) model the probability $p(\widetilde{C}_n|\mathbf{x}_n)$ as the marginal distribution of the observed labels $\widetilde{C}_n$ over all values the unknown class labels C_n may take:

$$p(\widetilde{C}_n = C^k|\mathbf{x}_n) = \sum_{a=1}^{K} p(\widetilde{C} = C^k|C = C^a)\, p(C_n = C^a|\mathbf{x}_n) \quad (2)$$

where $p(\widetilde{C} = C^k|C = C^a)$ is the probability for a specific type of label noise affecting the two classes C^a and C^k. These *transition probabilities* for all class configurations form the K x K *transition matrix* $\mathbf{\Gamma}$ with $\mathbf{\Gamma}(a,k) = \gamma_{ak} = p(\widetilde{C} = C^k|C = C^a)$. The transition matrix $\mathbf{\Gamma}$ contains the parameters of a NAR model which are estimated along with the parameters $\mathbf{w}$ in eq. 1. Because this kind of model is unrealistic to describe changes in land cover, we introduce a weight $g_n \in (0...1]$ for every sample n to control its influence in the training process. In the beginning, these weights are all set to 1; Section 3.3.3 describes how they are changed iteratively to consider the assumption that changes occur in local spatial clusters. To deterimine the unknown parameters $\mathbf{w}$ and $\mathbf{\Gamma}$, we apply maximum likelihood estimation of the unknown parameters with a Gaussian prior over $\mathbf{w}$ for regularisation. Taking the negative logarithms of the involved probabilities, this results in the minimisation of the following target function:

$$E(\mathbf{w}, \mathbf{\Gamma}) = -\sum_{n=1}^{N} g_n \cdot \sum_{k=1}^{K} t_{nk} \ln(S_{nk}) + \frac{\mathbf{w}^T\mathbf{w}}{2\sigma^2}. \quad (3)$$

In eq. 3, t_{nk} is an indicator variable taking the value 1 if $\widetilde{C}_n = C^k$ and 0 otherwise, $S_{nk} = p(\widetilde{C}_n = C^k|\mathbf{x}_n)$ as defined in eq. 2, and the rightmost term corresponds to a Gaussian prior with zero mean and covariance $\sigma \cdot \mathbf{I}$, where $\mathbf{I}$ is a unit matrix.

We use the Newton-Raphson method (Bishop, 2006) for minimising $E(\mathbf{w}, \mathbf{\Gamma})$. In each iteration τ, the parameter vector $\mathbf{w}^\tau$ is determined from $\mathbf{w}^{\tau-1}$ according to $\mathbf{w}^\tau = \mathbf{w}^{\tau-1} - \mathbf{H}^{-1}\nabla E$, where $\nabla E = [\nabla_{w_2} E^T, ..., \nabla_{w_K} E^T]^T$ is the gradient of $E(\mathbf{w}, \mathbf{\Gamma})$:

$$\nabla_{w_j} E = \sum_{n=1}^{N} g_n \cdot (f_{nj} - \bar{t}_{nj})\mathbf{\Phi}(\mathbf{x}_n) + \frac{1}{\sigma^2}\mathbf{w}. \quad (4)$$

In eq. 4 we use the shorthand $f_{na} = p(C_n = C^a|\mathbf{x}_n)$ for the posterior in eq. 1, and $\bar{t}_{nj} = f_{nj}\sum_{k=1}^{K}\Big(\gamma_{jk}\frac{t_{nk}}{S_{nk}}\Big)$. The Hessian matrix $\mathbf{H}$ consists of (K-1) x (K-1) blocks $\mathbf{H}_{ij} = \nabla_{w_i}\nabla_{w_j} E$:

$$\nabla_{w_i}\nabla_{w_j} E = \sum_{n=1}^{N} g_n\big(f_{ni}f_{nj}\xi + I_{ij}(f_{nj} - \bar{t}_{nj})\big)\mathbf{\Phi}(\mathbf{x}_n)\mathbf{\Phi}(\mathbf{x}_n)^T + \frac{\delta(i=j)}{\sigma^2}\mathbf{I}$$

where $\xi = \sum_{k=1}^{k}\Big(\gamma_{jk}\gamma_{ik}\frac{t_{nk}}{S_{nk}^2}\Big)$, I_{ij} are the elements of a unit matrix, and $\delta(\cdot)$ is the Kronecker delta function delivering a value of 1 if the argument is true and 0 otherwise.

Optimising for the unknown weights requires knowledge about the transition matrix $\mathbf{\Gamma}$, which, however, is unknown. Bootkrajang and Kabán (2012) propose an iterative procedure similar to expectation maximisation (EM). Starting from coarse initial values for $\mathbf{\Gamma}$, the parameters $\mathbf{w}$ of the classifier are updated as just described. Using these weights, the transition matrix $\mathbf{\Gamma}$ is updated afterwards, expanding the updating step presented in (Bootkrajang and Kabán, 2012) by the weights g_n:

$$\gamma^\tau = \frac{1}{c}\gamma_{jk}^{\tau-1}\sum_{n=1}^{N} g_n t_{nk}\frac{f_{nj}}{S_{nk}^{\tau-1}},$$

where $c = \sum_{l=1}^{K}\Big(\gamma_{jl}^{\tau-1}\sum_{n=1}^{N} g_n t_{nl}\frac{f_{nj}}{S_{nl}^{\tau-1}}\Big)$ and $S_{nk}^{\tau-1} = \sum_{j=1}^{K}\gamma_{jk}^{\tau-1} f_{nj}$.

This alternating update of the parameters $\mathbf{w}$ and $\mathbf{\Gamma}$ is repeated until a termination criterion is reached. The estimated parameters $\mathbf{w}$ are related to a classifier delivering the posterior for the unknown current labels C_n, not the noisy labels $\widetilde{C}_n$.

Note that this training with equal weights g_n was already used in (Maas et al., 2016). In this paper this is just the case in the beginning of the training procedure. Note that the transition matrix $\mathbf{\Gamma}$ only represents the transition between the old database and the current labels in this initial step with equal weights g_n. If the weights of training samples in large clusters of potential changes are low (cf. Section 3.3.3), the majority of the samples affected by label noise will have a low impact on the result, so that $\mathbf{\Gamma}$ only represents residual label noise of small local extents for which the NAR model is a sufficiently good approximation.

3.2 CRF considering the existing map

CRFs are graphical models consisting of nodes and edges that can be used to consider local context in a probabilistic classification framework (Kumar and Hebert, 2006). The nodes of the underlying graph represent random variables whereas the edges connect pairs of nodes and describe their statistical dependencies. Here, the unknown nodes correspond to the current labels C_n of all pixels n, and the edges are defined on the basis of a 4-neighbourhood on the image grid. As described above, the observed variables are the image data $\mathbf{x}$ and, different from (Kumar and Hebert, 2006), the observed class labels $\widetilde{\mathbf{C}}$ (cf. fig. 1 for the structure of the graphical model). The joint posterior $P(\mathbf{C}|\mathbf{x}, \widetilde{\mathbf{C}})$ of the unknowns given the observations is modelled by:

$$P(\mathbf{C}|\mathbf{x}, \widetilde{\mathbf{C}}) = \frac{1}{Z}\exp\Big[\sum_n\Big(A_x(C_n, \mathbf{x}) + A_m(C_n, \widetilde{C}_n)\Big) + \sum_{n,m\in\epsilon} I(C_n, C_m, \mathbf{x})\Big], \quad (5)$$

were Z is a normalization constant and ϵ is the set of edges in the graph. The association potential $A_x(C_n, \mathbf{x})$ connects the unknown label C_n of pixel n with the image data $\mathbf{x}$. Its dependency from the entire input image $\mathbf{x}$ is considered by using site-wise feature vectors $\mathbf{x}_n(\mathbf{x})$, which may be a function of certain image regions. Any discriminative classifier can be used to model this potential (Kumar and Hebert, 2006); here, it is based on the posterior $p(C_n|\mathbf{x}_n)$ of logistic regression according to eq. 1:

$$A_x(C_n, \mathbf{x}) = \ln p(C_n|\mathbf{x}_n). \quad (6)$$

The interaction potential $I(C_n, C_m, \mathbf{x})$ describes the statistical dependencies between a pair of neighbouring labels C_n and C_m. In this paper, the contrast-sensitive Potts model is used for that purpose, which results in a data-dependant smoothing of the resultant label image (Boykov et al., 2001):

$$I(C_n, C_m, \mathbf{x}) = \delta(C_n, C_m) \cdot \beta_0 \left(\beta_1 + \left(1 - \beta_1\right) \cdot e^{\left(-\frac{\mathbf{\Delta x}^2}{2\sigma_D^2}\right)} \right), \tag{7}$$

where the parameters β_0 and β_1 describe the overall degree of smoothing and the impact of the data-dependent term, respectively, σ_D is the average squared distance between neighbouring feature vectors, $\mathbf{\Delta x} = \| \mathbf{x}_n - \mathbf{x}_m \|$ is the distance of two feature vectors $\mathbf{x}_n$ and $\mathbf{x}_m$, and $\delta(\cdot)$ is the Kronecker delta function.

The observed labels are related to the unknown class labels by the temporal association potential $A_m(C_n, \widetilde{C}_n)$, derived from the probability of the unknown label given the observed one:

$$A_m(C_n, \widetilde{C}_n) = \theta_n \cdot \ln p(C_n = C^k | \widetilde{C}_n = C^a) \tag{8}$$

In eq. 8, there is an individual weight $\theta_n \in [0...1]$ for every pixel n. This weight models the influence of the observed label on the classification result of this pixel in inference. As we shall see in Section 3.3, these weights will be adapted in the inference process to reduce the impact of the observed labels for pixels that are very likely to belong to a larger area affected by a change.

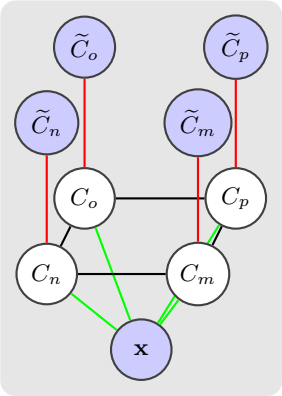


Figure 1. Graph structure of the expanded CRF: C: unknown labels, $\widetilde{C}$: observed labels, $\mathbf{x}$: image data.

3.3 Training and inference

In order to obtain the optimum configuration of the current class labels given the observations by maximising $P(\mathbf{C}|\mathbf{x}, \widetilde{\mathbf{C}})$ according to eq. 5, a joint iterative training and inference strategy is applied. After the determination of initial parameters of the association potential and the parameters of the temporal association potentials in an initial training phase, an iterative scheme of classification and re-training is applied in which the weigths of pixels in large areas of potential change according to the current classification result are modified to reduce their impact on the results. These steps are described in the subsequent sections.

3.3.1 Initial training and classification: In the initial training phase, the observed labels and the data are used for label noise robust training of the logistic regression classifier that serves as the basis for the association potentials of the CRF. For that purpose, the method described in Section 3.1 is applied, using identical weights $g_n = 1$ for all training samples. This will result in an initial set of parameters $\mathbf{w}$ for the association potentials and a transition matrix $\mathbf{\Gamma}$ that contains the transition probabilities $p(\widetilde{C}_n = C^a | C_n = C^k)$ of the NAR model (Bootkrajang and Kabán, 2012). According to the theorem of Bayes, these probabilities are related to the probabilities $p(C_n = C^k | \widetilde{C}_n = C^a)$ required for the temporal association potential (eq. 8) by:

$$p(C_n = C^k | \widetilde{C}_n = C^a) = \frac{p(\widetilde{C}_n = C^a | C_n = C^k) \cdot p(C_n = C^k)}{p(\widetilde{C}_n = C^a)}$$

As we have no access to the distribution of the unknown class labels $p(C_n)$, we assume $p(C_n = C^k) \approx p(\widetilde{C}_n = C^k)$ to derive the temporal association potential from $\mathbf{\Gamma}$. These parameters are kept constant in the subsequent iteration process for the reasons already pointed out in Section 3.1: the transition matrix corresponds to the real transition probabilities only in the first iteration (when all training samples have an identical weight $g_n = 1$). The parameters of the interaction potentials (β_0, β_1; cf. eq. 7) are set to values found empirically.

For the determination of the optimal configuration of labels $\mathbf{C} = argmax(P(\mathbf{C}|\mathbf{x}, \widetilde{\mathbf{C}}))$ loopy belief propagation is used (Frey and MacKay, 1998). In the initial classification, the weights θ_n of the temporal association potentials is set to 0 for all pixels, so that this classification is only based on the current state of the association and the interaction potentials.

3.3.2 Iterative re-training and classification: By comparing the current label image with the outdated map, areas of potential changed areas can be detected. This information is used to update the weight g_n of each training sample, and label noise robust training of the logistic regression classifier is repeated, using the updated weights. The way in which the weights are updated is explained in Section 3.3.3. Training will result in new values for the parameters $\mathbf{w}$ of the association potentials of the CRF.

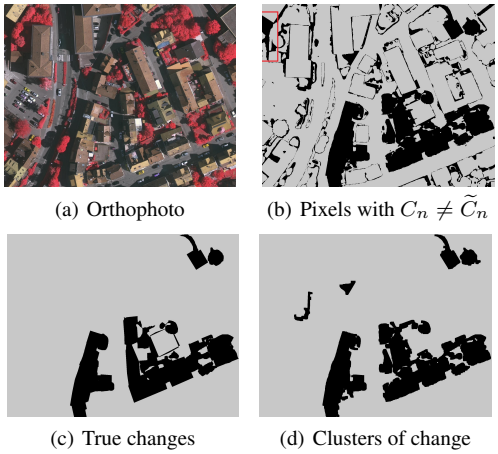
Furthermore, the information about potential areas of change is also used to change the weights θ_n of the temporal association potentials as explained in Section 3.3.4. Using the updated parameters $\mathbf{w}$ and weights θ_n, another round of inference is carried out, which will lead to an improved classification result. This procedure of updating weights on the basis of the current state of the classification results, re-training and inference is repeated until the proportion of weights that are changed in an iteration is below a threshold or a maximum number of iterations is reached. The procedure is inspired by re-weighting strategies for robust estimation in adjustment theory, e.g. (Förstner and Wrobel, 2016).

3.3.3 Weights g_n of training samples: The weight g_n of a training sample n should be high for samples which are probably not affected by a change and low for other ones. The weights are initialised by $g_n = 1$ as long as no information about changes is available. After classification, the resulting labels C_n can be compared to the map $\widetilde{C}_n$ to generate a binary map B_C of potential changes. However, as indicated in fig. 2(b) for an aerial image, this binary map will also contain classification errors.

To distinguish between real changes and classification errors three assumptions are made. First, classification errors often occur at object boundaries, e.g. because of mixed pixels or because of matching errors if digital surface models (DSM) are used in classification. Thus, a set of connected foreground pixels in B_C forming a line that is thinner than a threshold s is very likely caused by classification errors. Such sets are removed by morphological filtering using a structural element of size s. The second assumption is that changes occur in clusters having a cer-

tain minimum size. This is considered by removing all connected components of foreground pixels in B_C which cover an area smaller than a threshold u. The third assumption is that in areas affected by cast shadows, the quality of spectral information or of the DSM (if available) is poor and, thus, potential changes as indicated by B_C are very likely to correspond to classification errors. To detect shadow areas, the median and the mean of the image intensity in each cluster cl is compared to the median and the mean of the entire images. If $mean_{cl} < mean_{img}/2$ and $med_{cl} < med_{img}/2$, i.e., if the pixels in the cluster are very dark compared to the image, the pixels belonging to cluster cl are removed from the binary map of potential changes B_C. The remaining foreground pixels in B_C are likely to correspond to real changes (cf. fig. 2(d) for an example).

For pixels corresponding to the foreground in B_C, the weights g_n are decreased by a constant c, so that in iteration $t+1$, the weight of the corresponding samples is given by $g_n^{t+1} = max(g_n^t - c, \xi)$. The minimal weight is set to a small constant ξ to avoid numerical problems. The weights of pixels that belong to the background in B_C are updated according to $g_n^{t+1} = min(g_n^t + c, 1)$. As a consequence, the weights of pixels that are considered to be changes will be reduced in each iteration; however, a pixel may regain influence if in a certain iteration its most likely class label is identical to the one from the map, e.g. due to the influence of its neighbours or due to the temporal model.



(a) Orthophoto (b) Pixels with $C_n \neq \widetilde{C}_n$

(c) True changes (d) Clusters of change

Figure 2. Example for the identification of potential areas of change. Black / gray: changed / unchanged pixels. Red rectangle in (b): a cluster that corresponds to a shadow.

3.3.4 The weights θ_n of the temporal association potential: The weight θ_n of pixel n regulates the impact of the temporal association potential and, thus, the influence of the outdated map to the resulting label configuration $\mathbf{C}$ (cf. eq. 8). If a pixel is probably not affected by a change, the weight θ_n should be high, otherwise it should be low. The initial weight for each pixel is 0, because in the beginning we do not want to bias the result to reject potential changes. In the subsequent iterations, the binary map of potential changes used to adapt the weights g_n of the training samples (cf. Section 3.3.3) is also used to guide the adaptation of the weights θ_n of the temporal model, because the same assumption w.r.t. to the plausibility of a potential change indicated by the current classification results apply. In iteration $t+1$, the temporal association potentials for pixels corresponding to the foreground in B_C will the be weighted by $\theta_n^{t+1} = max(\theta_n^t - c, 0)$. The corresponding weights of pixels that belong to the background in B_C are updated by $\theta_n^{t+1} = min(\theta_n^t + c, 1)$.

4. EXPERIMENTS

4.1 Test data and test setup

We used three datasets in our experiments. The first one consists of a part of Vaihingen data of the ISPRS 2D semantic labelling contest (Wegner et al., 2015). We use ten of the training patches, each consisting of about 2000 × 2500 pixels. For each patch, a colour infrared true orthophoto (TOP) and a DSM are available with a ground samling distance (GSD) of 9 cm. The reference consists of five classes: *impervious surfaces* (*sur.*), *building* (*build.*), *low vegetation* (*veg.*), *tree*, and *car*. As cars are not a part of a topographic map, this class was merged with *sur*. For each pixel, we defined a feature vector $\mathbf{x}_n(\mathbf{x})$ consisting of the normalised difference vegetation index (NDVI), the normalised DSM (nDSM), the red band of the TOP smoothed by a Gaussian filter with $\sigma = 2$, and hue and saturation obtained from the TOP, both smoothed by a Gaussian filter with $\sigma = 10$. These features were selected from a larger pool based on the feature importance analysis of a random forest classifier (Breiman, 2001).

The other two data sets are based on satellite imagery and were also used in in (Maas et al., 2016). The first one consists of a Landsat image from 2010 of an area near Herne, Germany, with a GSD of 30 m and a size of 362 × 330 pixels. The second dataset consists of a RapidEye image of an area near Husum, Germany, from 2010. Its GSD is 5 m and its size is 3547 × 1998 pixels. In both cases only the red, green and near infrared bands are available. The reference contain four classes *residential area* (*res.*), *rural streets*, *forest* (*for.*) and *cropland* (*crop.*). As the class *rural streets* is underrepresented in both images, we merged it with *cropland*. In both datasets, 19 features were selected: four Haralick features (energy, contrast, homogenity and entropy) related to texture, the mean and variance of five spectral features (*near infrared band*, *intensity*, *hue*, *saturation* and *ndvi*) in a local neighbourhood of 6 × 6 pixels and the values of the same spectral features smoothed by a Gaussian filter with $\sigma = 5$.

For Vaihingen, we used a feature space mapping $\mathbf{\Phi}(\mathbf{x}_n)$ based on quadratic expansion, whereas for Husum and Herne no feature space mapping was used. The hyperparameter for regularisation in eq. 3 was set to $\sigma = 10$. The initial values for the transition matrix $\mathbf{\Gamma}$ (cf. Section 3.1) were $\gamma_{ij} = 0.8$ for $i = j$ and $\gamma_{ij} = 0.2/(K-1)$ for $i \neq j$, where K is the number of classes. The initial values for the parameter vector $\mathbf{w}$ of logistic regression were determined by standard logistic regression training without assuming label noise. The parameters of the contrast-sensitive Potts model were set to $\beta_0 = 1.0$ and $\beta_1 = 0.5$. The thresholds for updating the weights (Section 3.3.3) depend on the GSD. For Vaihingen, the threshold for object borders s was set to 0.5 m, assuming wrong classifications near object borders to be caused by errors in the nDSM or mixed pixels. The minimal size u of an object is set to 4 m × 4 m (i.e., smaller than a small house). For the satellite data, s is set to 2 pixels, because mistakes caused by the nDSM do not exist, and u is set to 250 m × 250 m, assuming this is the minimum size of a new residential area or field. The value c for updating the weights (Sections 3.3.3 and 3.3.4) was found empirically and set to 0.1. Except for the dataset of Herne, where all pixels are used due to the small image size, just about 20% of the data are used for training to reduce the processing time. The iteration is terminated (Section 3.3.2) if either less than 0.01% of the weights for the observed labels in classification change or if at least 40 iterations have been done.

For all experiments we manually changed the reference to simulate an outdated map. For each patch of the Vaihingen dataset

three simulated maps were created, each with a different amount of change. For Herne and Husum, the changed map from (Maas et al., 2016) were used. Based on these data, we carried out four experiments. In the first experiment (**Init**), training and classification was carried out as in (Maas et al., 2016), i.e. without iterative re-training and classification ($g_n = 1 = const, \theta_n = 0 = const$). The second experiment ($\mathbf{V}^g$) is based on our method, but without considering the outdated map ($\theta_n = 0 = const$). It shows the impact of the sample weights g_n introduced in section 3.1 in the training step. The third experiment ($\mathbf{V}_\theta$) uses constant training weights $g_n = 1$, but does apply the modified weights θ_n to include the map information. The last experiment, $\mathbf{V}^g_\theta$, uses our method with weight modification both in the training and classification steps. In each case, we compare the results to the reference on a per-pixel basis, determining the overall accuracy (OA) as well as completeness and correctness per class (Heipke et al., 1997). Comparing the simulated map with the real reference shows the amount of change in the corresponding data set, and the resultant quality indices are also reported (**map**); 100% - OA of **map** gives the amount of simulated change in each experiment. We do not distinguish a training set from a test set because an outdated map is always used, at least for training.

4.2 Results and evaluation

4.2.1 Vaihingen: Fig. 3 shows the OA of all patches achieved for three versions of the outdated map (map 1 – map 3) for Vaihingen. In most cases the variant $\mathbf{V}^g_\theta$ achieves the best OA (85%-90%), but variant $\mathbf{V}_\theta$ performs at a similar level, and both variants clearly outperform the variants without weights and without considering the outdated map (**Init**, $\mathbf{V}^g$). Obviously, the inclusion of the outdated map has a relatively high impact on the quality of the results, improving the OA by 2%-10%. This is mainly caused by an improved classification at object boundaries or at individual pixels. In fact, in some cases, the variants not considering the outdated map lead to results where a larger percentage of change than actually present is predicted, so that the corresponding OA is lower than the one indicated by **map**.

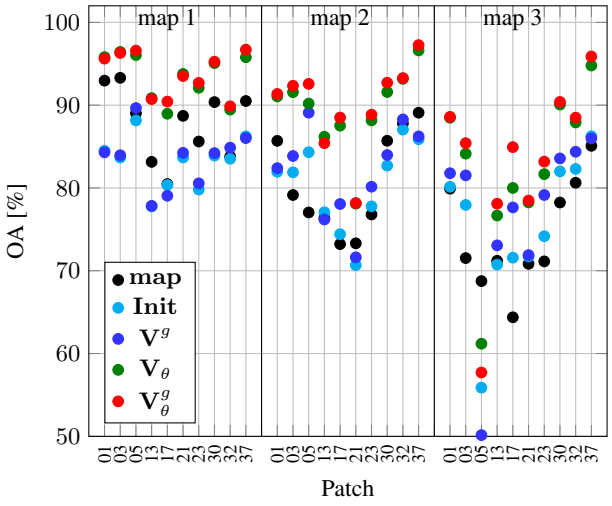
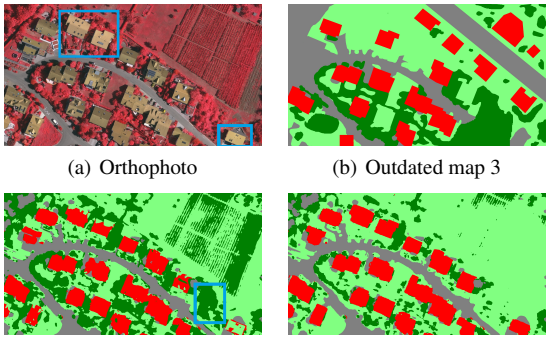


Figure 3. Overall accuracy for the four variants in Vaihingen.

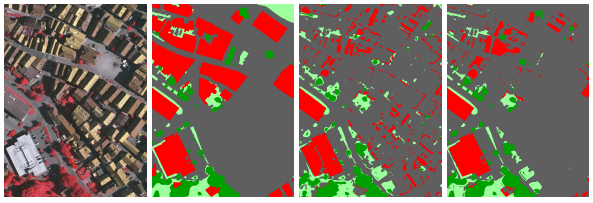
The advantage of considering the sample weights g_n in training becomes more obvious for experiments with a large amount of change. If the level of change is small, it can be compensated by the original method based on the NAR model (Maas et al., 2016). If the label noise cannot be compensated by the NAR model any more, considering the weights can improve the results. One example is patch 17 (fig. 4). It contains three buildings with a brighter appearance than the rest (blue rectangles in fig. 4(a)). Only one of them is contained in the outdated map (fig. 4(b)). Without considering the weights (variant **Init**, fig. 4(c)), one building is mostly classified as *veg*. In variant $\mathbf{V}^g$ the two changed buildings are correctly detected (fig. 4(d)). Another difference between the results of variants **Init** and $\mathbf{V}^g$ is the label of the vineyard which belongs to the class *veg.* but is often classified as *tree* in experiment **Init**. Without considering weights, the probability $p(C_n|\mathbf{x}_n)$ is low for all classes in the area of the vineyard, so that the classification results are not reliabe. By considering the sample weights in experiment $\mathbf{V}^g$, the probability $p(C_n|\mathbf{x}_n)$ for the class *veg.* is much higher than for the other classes. However, because the vineyard has a similar appearance to trees, the tree marked in fig. 4(c) is also classified as *veg.* in $\mathbf{V}^g$.



(a) Orthophoto (b) Outdated map 3 (c) **Init** (d) $\mathbf{V}^g$

Figure 4. Data and results from patch 17 (Vaihingen). Red: *build.*, dark green: *tree*, light green: *veg.*, gray: *sur.*. Blue rectangles: highlighted objects discussed in the text.

For patch 5 (fig. 5) and the otudated map 3 with more than 30% label noise, OA is always below 61% (fig. 3). In this case nearly 50% of all building pixels are labeled as *sur.* in the outdated map. This amount of label noise cannot be dealt with by the original method (**Init**). The transition probabilities γ_{ii} for no change for *build.* and *sur.* determined in the initial training step are close to 1 and, thus, not very accurate. Consequently, the iterative weight updating procedure does not converge to the correct solution.



(a) Orthophoto (b) map 3 (c) **Init** (d) $\mathbf{V}^g_\theta$

Figure 5. Data and results from patch 5 (Vaihingen). Red: *build.*, dark green: *tree*, light green: *veg.*, gray: *sur.*

Figs. 6 and 7 show the completeness and the correctness of the results. Both quality indices are higher for variants $\mathbf{V}^\theta$ and $\mathbf{V}^g_\theta$ than for the others, which again highlights the importance of using the outdated map for classification. Using the sample weights g_n in the training process does not improve the completeness in most cases, but it does have a small positive impact on the correctness.

For buildings, we also provide an evaluation on a per-object basis, counting a detected building (i.e. a connected component of pixels classified as *build.*) as a true positive if more than 70% of its area overlaps with a reference building. Because small buildings are often not included in maps, buildings smaller than 16 m^2

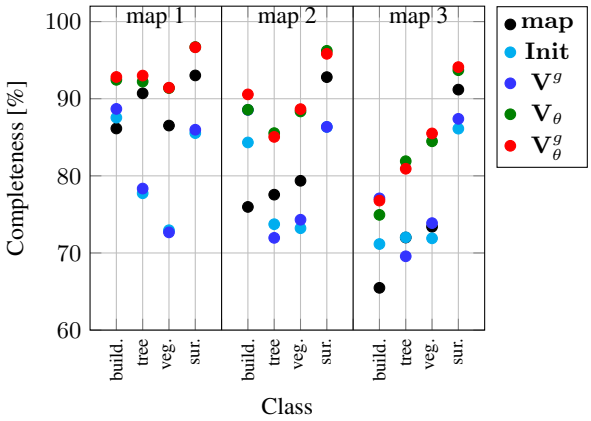


Figure 6. Average completeness over all patches in Vaihingen.

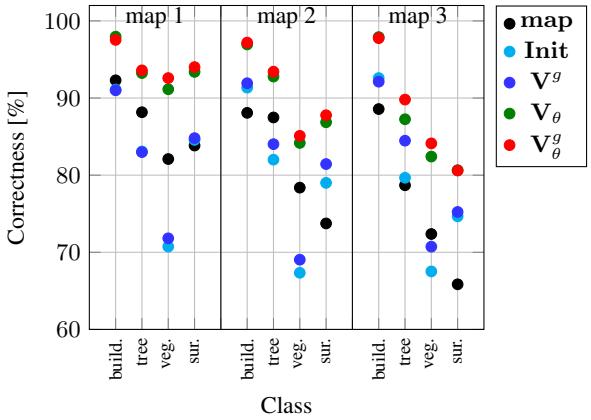


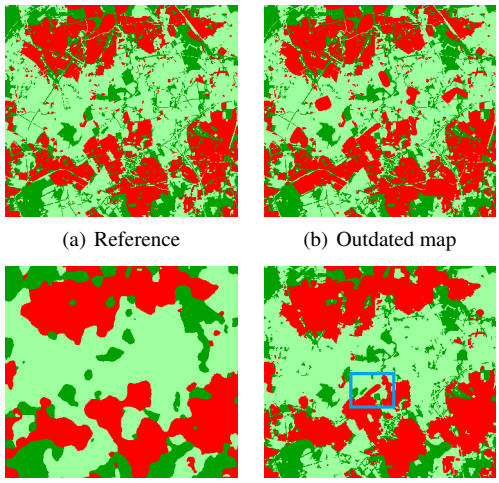
Figure 7. Average correctness over all patches in Vaihingen.

were excluded from the evaluation. The mean completeness and correctness of all areas are shown in tab. 1. Again, variant $\mathbf{V}^g_\theta$ achieves the best completeness (98.9%) and correctness (82.5%). However, variants $\mathbf{V}_\theta$ and $\mathbf{V}^g$ do not perform significantly worse considering the standard deviations of the quality indices. Nevertheless one can notice an positive impact of the new developments presented in this paper (variants $\mathbf{V}^g$, $\mathbf{V}_\theta$ and, particularly, $\mathbf{V}^g_\theta$) compared to the original algorithm (**Init**) (Maas et al., 2016).

	$\mathbf{V}^g_\theta$	$\mathbf{V}_\theta$	$\mathbf{V}^g$	**Init**	**map**
Corr.	99 [4]	99 [4]	96 [8]	93 [14]	91 [11]
Compl.	82 [19]	80 [17]	82 [18]	75 [20]	74 [13]

Table 1. Completeness and correctness on a per-object basis for buildings; mean of all areas in % [standard deviation in %]

4.2.2 Herne and Husum: As the amount of change in Husum and Herne is quite small (3% - 4%), using the sample weights g_n does not affect the results much; the OA changes by less than 0.6%. Thus, this section focuses on the impact of using the outdated map for classification. In tab. 2 OA, completeness and correctness are shown for both datasets for variants **Init** and $\mathbf{V}_\theta$. All values are larger for variant $\mathbf{V}_\theta$ by a large margin, the OA increasing by 13.9% for Herne and by 5.2% for Husum. One reason for that increase is the improvement of the delineation of object borders. As the features all depend on a local subset of pixels, borders of objects are blurred in the standard classification process. In variant $\mathbf{V}_\theta$ these areas can be correctly classified in regions without change. If regions of change are smaller than the threshold u (Section 3.3.3), considering the map has the same effect, which may lead to cases in which such small changes are not detected. An example for such a situation in Herne with variant $\mathbf{V}_\theta$ is indicated by a blue rectangle in fig. 8(d). To highlight the potential for detecting changes despite using the existing map for classification, tab. 3 shows the OA achieved for pixels in the areas affected by a change. The results show that the improved OA for the entire image caused by the inclusion of the outdated map (cf. tab. 2) comes at the cost of a reduced OA in the changed areas. In Husum, this reduction in OA is low (0.8%). In Herne it is somewhat larger (7%), though still considerably smaller than the improvement for the entire scene (14.4%).



(a) Reference (b) Outdated map

(c) **Init** (d) $\mathbf{V}_\theta$

Figure 8. Label images for Herne. Dark green: forest, light green: *crop.*, red: *res.*. Blue rectangle: an undetected change.

Dataset		Herne $\mathbf{V}_\theta$	Herne **Init**	Husum $\mathbf{V}_\theta$	Husum **Init**
OA		89.7	75.8	96.7	91.5
Compl.	res.	95.7	83.4	88.7	79.3
	for.	76.8	54.9	91.5	77.7
	crop.	92.4	81.6	98.2	94.4
Corr.	res.	81.2	74.3	87.5	72.2
	for.	96.7	69.7	92.2	75.2
	crop.	93.8	79.4	98.3	95.9

Table 2. OA, completeness, correctness for Husum and Herne.

Herne, **Init**	Herne, $\mathbf{V}_\theta$	Husum, **Init**	Husum, $\mathbf{V}_\theta$
75.9 %	68.9 %	92.6 %	91.8 %

Table 3. OA of Husum and Herne for areas affected by a change.

5. CONCLUSION AND FUTURE WORK

In this paper we presented a iterative method for supervised classification under label noise making use of the existing map both for training and in the classification process. No manual effort for the generation of training data was required. In both, the training and the classification procedure we considered the fact that changes in land cover usually appear in clusters. In training this was achieved by using a weight for each training sample in order to reduce the impact of samples in larger areas of change. By adding the labels of the map to the CRF as weighted observations, our method includes the map information for pixels which are unlikely to correspond to changes. Thus, new objects can be found without the additional map information while pixels probably not affected by label noise can take advantage of this prior information.

We tested our method using datasets with different properties and varying degrees of label noise. Due to our re-weighting scheme for training samples the method can also deal with larger amount of noise, but the improvement brought about by this strategy was smaller than expected. The inclusion of the map information to the CRF has a considerably larger positive effect, largely due to a better classification of pixels near object boundaries. The actual changes are detected nearly as good as without considering the map in classification, although very small changed objects might not be detected. These observations could be made independently from the GSD of the images. A major limitation of the method is that each cluster in feature space still must contain enough correct training samples for it to work. If the results of the base classifier in the initialization step are sufficiently good, considering the map in the classification can improve the results considerably.

In our future work we want to expand our model by images from other epochs, so that not only the map can help to improve the classification result, but also other image data. Additionally we want to expand our experiments to data with real changes to see how our method works under more realistic circumstances in terms of the extent of change, level of detail or number of classes.

ACKNOWLEDGEMENTS

This research was funded by the German Science Foundation (DFG) under grant HE-1822/35-1. 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.

REFERENCES

An, W. and Liang, M., 2013. Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises. *Neurocomputing* 110, pp. 101–110.

Bishop, C. M., 2006. *Pattern recognition and machine learning.* 1st edn, Springer. New York (NY), USA.

Bootkrajang, J. and Kabán, A., 2012. Label-noise robust logistic regression and its applications. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, pp. 143–158.

Boykov, Y., Veksler, O. and Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence* 23(11), pp. 1222–1239.

Breiman, L., 2001. Random forests. *Machine learning* 45(1), pp. 5–32.

Bruzzone, L. and Persello, C., 2009. A novel context-sensitive semisupervised svm classifier robust to mislabeled training samples. *IEEE Transactions on Geoscience and Remote Sensing* 47(7), pp. 2142–2154.

Büschenfeld, T., 2013. Klassifikation von Satellitenbildern unter Ausnutzung von Klassifikationsunsicherheiten. PhD thesis, Fortschritt-Berichte VDI, Reihe 10 Informatik / Kommunikation, Vol. 828, Institute of Information Processing, Leibniz Universität Hannover, Germany.

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

Förstner, W. and Wrobel, B. P., 2016. Robust estimation and outlier detection. In: *Photogrammetric Computer Vision*, 1^{st} edn, Springer, Cham, Switzerland, chapter 4.7, pp. 141–159.

Frénay, B. and Verleysen, M., 2014. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems* 25(5), pp. 845–869.

Frey, B. J. and MacKay, D. J., 1998. A revolution: Belief propagation in graphs with cycles. *Advances in neural information processing systems* 10, pp. 479–485.

Heipke, C., Mayer, H., Wiedemann, C. and Jamet, O., 1997. Evaluation of automatic road extraction. In: *International Archives of Photogrammetry and Remote Sensing*, Vol. XXII-3/4W2, pp. 151–160.

Hoberg, T., Rottensteiner, F., Feitosa, R. Q. and Heipke, C., 2015. Conditional random fields for multitemporal and multiscale classification of optical satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing* 53(2), pp. 659–673.

Jia, K., Liang, S., Wei, X., Zhang, L., Yao, Y. and Gao, S., 2014. Automatic land-cover update approach integrating iterative training sample selection and a markov random field model. *Remote Sensing Letters* 5(2), pp. 148–156.

Jianya, G., Haigang, S., Guorui, M. and Qiming, Z., 2008. A review of multi-temporal remote sensing data change detection algorithms. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII-B7, pp. 757–762.

Kumar, S. and Hebert, M., 2006. Discriminative random fields. *International Journal of Computer Vision* 68(2), pp. 179–201.

Li, Y., Wessels, L. F., de Ridder, D. and Reinders, M. J., 2007. Classification in the presence of class noise using a probabilistic kernel fisher method. *Pattern Recognition* 40(12), pp. 3349–3357.

Maas, A., Rottensteiner, F. and Heipke, C., 2016. Using label noise robust logistic regression for automated updating of topographic geospatial databases. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. III-7, pp. 133–140.

Mnih, V. and Hinton, G. E., 2012. Learning to label aerial images from noisy data. In: *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 567–574.

Radoux, J. and Defourny, P., 2010. Automated image-to-map discrepancy detection using iterative trimming. *Photogrammetric Engineering & Remote Sensing* 76(2), pp. 173–181.

Radoux, J., Lamarche, C., Van Bogaert, E., Bontemps, S., Brockmann, C. and Defourny, P., 2014. Automated training sample extraction for global land cover mapping. *Remote Sensing* 6(5), pp. 3965–3987.

Sarma, A. and Palmer, D. D., 2004. Context-based speech recognition error detection and correction. In: *Proceedings of HLT-NAACL 2004: Short Papers*, Association for Computational Linguistics, pp. 85–88.

Schistad Solberg, A. H., Taxt, T. and Jain, A. K., 1996. A Markov random field model for classification of multisource satellite imagery. *IEEE Transactions on geoscience and remote sensing* 34(1), pp. 100–113.

Wegner, J., Rottensteiner, F., Gerke, M. and Sohn, G., 2015. The ISPRS 2d labelling challenge. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html. Accessed 05/04/2017.