Which data do we need for training?

Domain Adaption and Learning under Label Noise

Franz Rottensteiner

Institute of Photogrammetry and GeoInformation
Leibniz Universität Hannover
rottensteiner@ipi.uni-hannover.de

Special thanks to

Prof. Christian Heipke
(IPI)

Prof. Jörn Ostermann
(tnt)

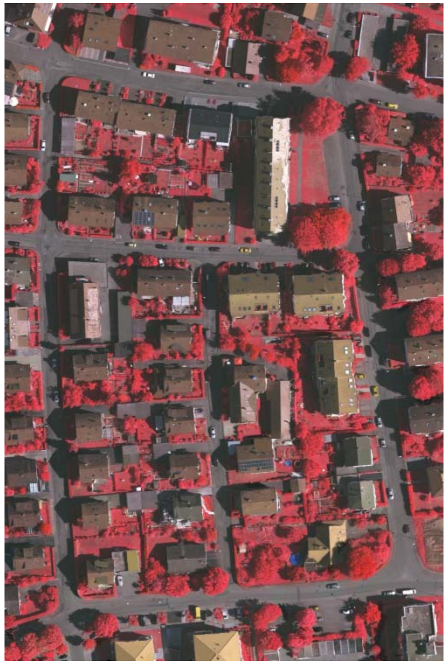
Alina Maas
(IPI)

Andreas Paul
(IPI)

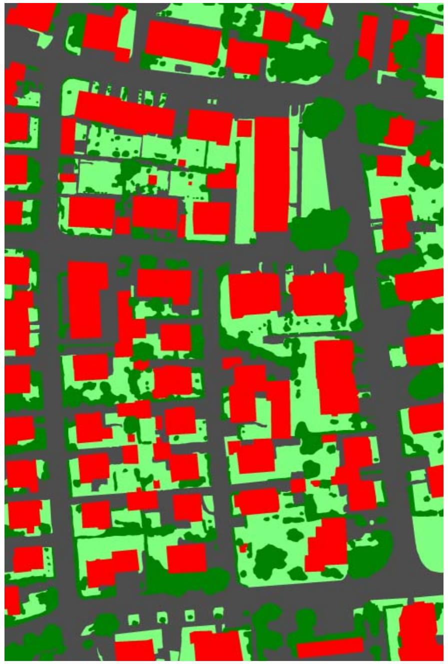
Karsten Vogt
(tnt)

Introduction

- Image analysis: make information contained in images explicit



CIR image



Semantic information

🟥 Building

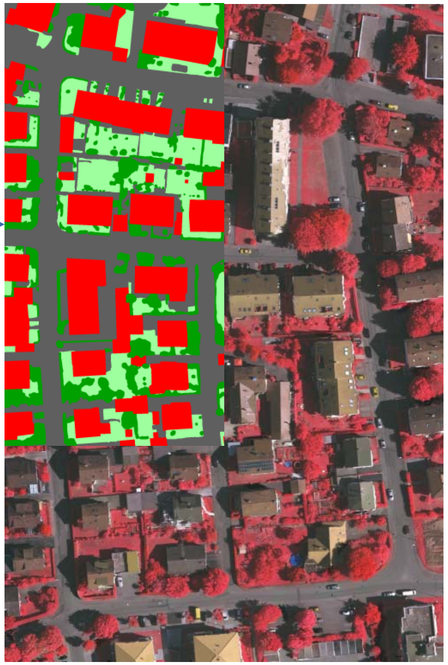
🟩 Tree

🟩 Vegetation

⬛ Street

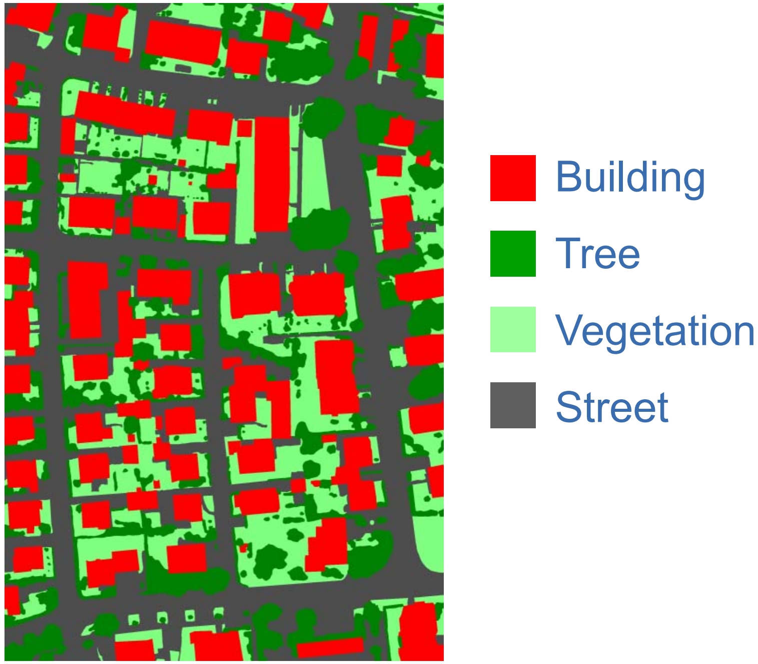
Introduction

- Image analysis: make information contained in images explicit



Training data

CIR image



Semantic information

- Supervised classification:
  - + Transferability: adapt classifier to new data via training data
  - – Training data have to be generated manually

How to Reduce the Efforts for Generating Training Data?

1) Adapt a classifier to new data with scarce or no new training data

→ **Transfer Learning** [Pan & Yang, 2010]

a) Domain adaptation: adapt classifier to new feature distribution [Bruzzone & Marconcini, 2009; Paul et al., 2015; 2016]

b) Source selection: find optimal source from a pool of training images [Vogt et al., 2017]

How to Reduce the Efforts for Generating Training Data?

1) Adapt a classifier to new data with scarce or no new training data

→ **Transfer Learning** [Pan & Yang, 2010]

   a) Domain adaptation: adapt classifier to new feature distribution [Bruzzone & Marconcini, 2009; Paul et al., 2015; 2016]

   b) Source selection: find optimal source from a pool of training images [Vogt et al., 2017]

2) Use existing map for training and classification [Maas et al., 2016; 2017]

→ **Learning under label noise** [Frénay & Verleysen, 2014]

Outline

- Introduction
- Transfer Learning:
  - Domain adaptation by instance transfer
  - Creating a synthetic domain by source selection
- Training under label noise:
  - Using existing maps for training and classification
- Conclusion

Transfer Learning

- Important definitions [Pan & Yang, 2010]:
  - Domain $\mathcal{D} = \{\underbrace{\mathcal{X}}_{\text{feature space}}, \underbrace{P(X)}_{\text{feature distribution}}\}$
  - Task $\mathcal{T} = \{\underbrace{\mathcal{C}}_{\text{label space}}, \underbrace{f(\cdot)}_{\text{predictive function (classifier)}}\}$

$\Bigg\}$ for *Source* and *Target* data

different, but related

Transfer Learning

- Important definitions [Pan & Yang, 2010]:
  - Domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$

    feature space

    feature distribution
  - Task $\mathcal{T} = \{\mathcal{C}, f(\cdot)\}$

    label space

    predictive function (classifier)

for *Source* and *Target* data

different, but related

- Assumptions:
  - Abundant amount of training samples in D_S
  - Few or no training samples in D_T
- Goal: Transfer knowledge from D_S to D_T

Domain Adaptation (DA)

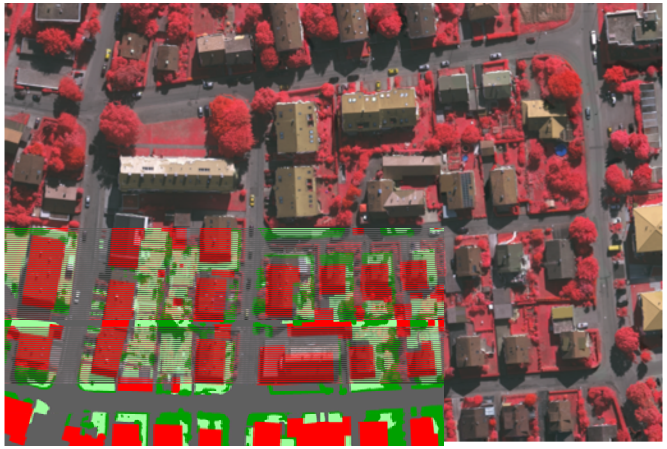
- Specific setting of transfer learning:
  - No training data in target domain
  - Tasks are identical
  - Domains are different (but related):

$$P(X_S) \neq P(X_T) \text{ and } P(C_S|X_S) \neq P(C_T|X_T)$$

- Method: Instance transfer
  - Replace source data by weighted semi-labeled target samples
  - Iterative adaptation of classifier to target domain data

DA: Scenario

- Classification of images:

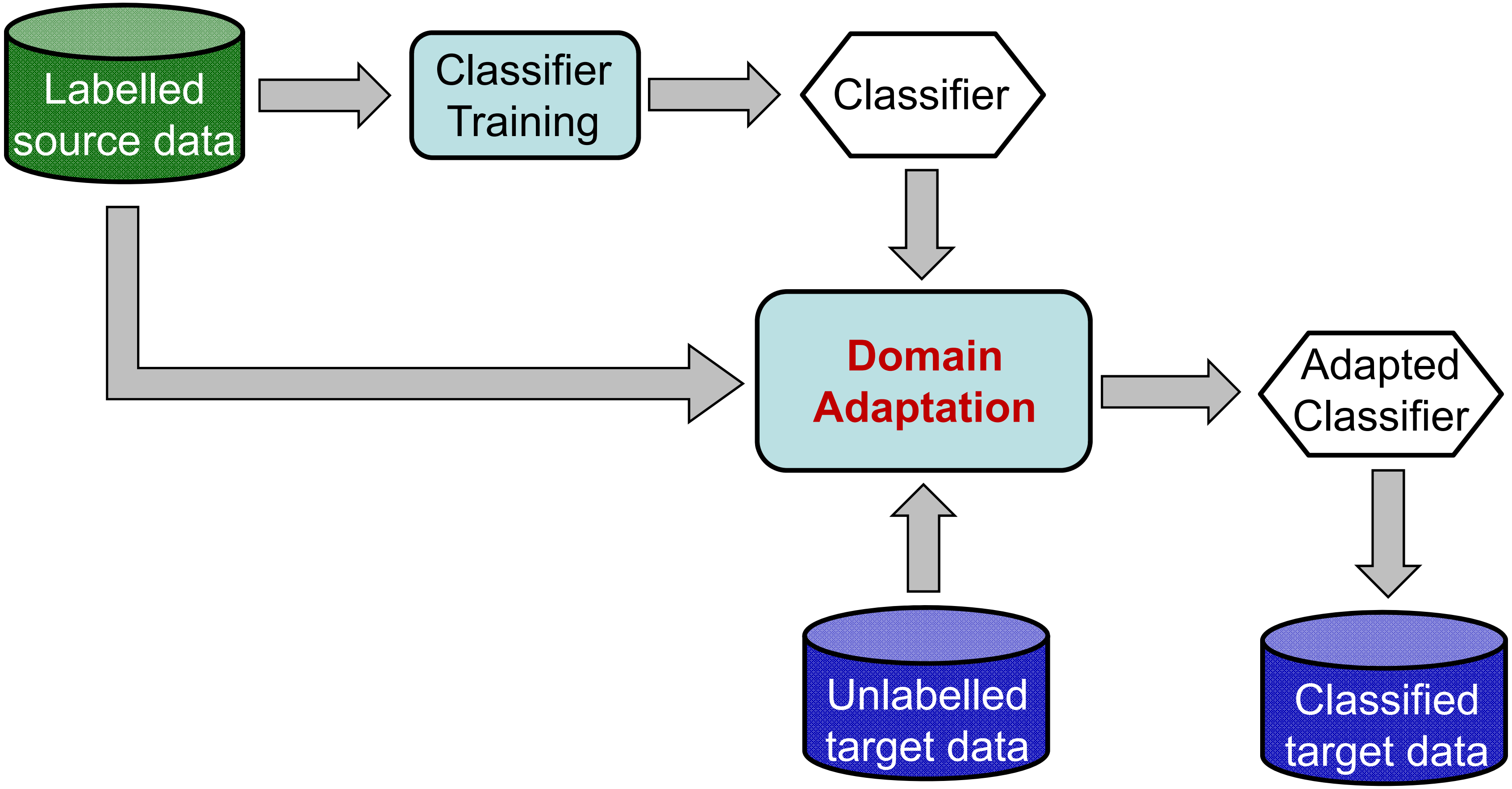


Source domain D_S: image with training samples

Target domain D_T: image, no training samples

  - Images in D_S and D_T have the same features
  - Class structures are identical

DA by Instance Transfer: General Strategy



Labelled source data

Classifier Training

Classifier

Domain Adaptation

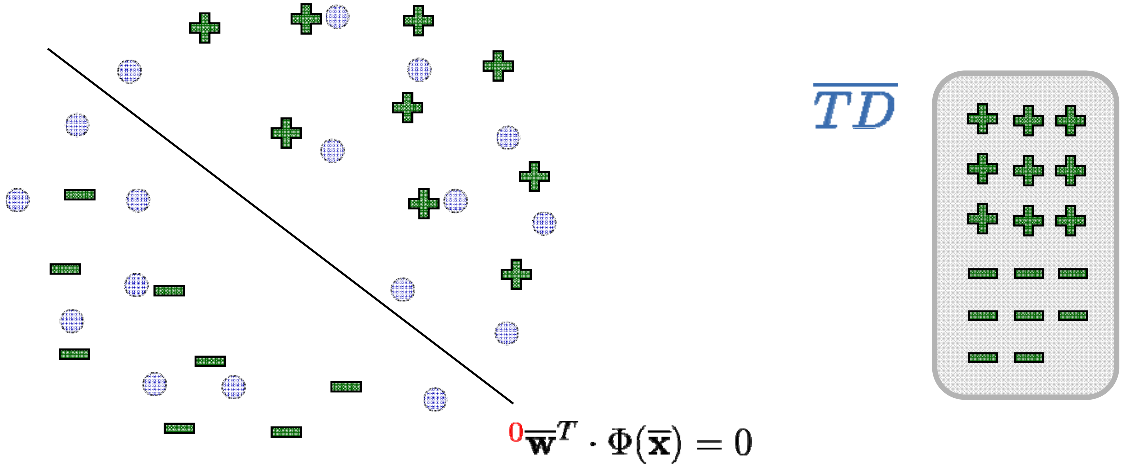
Adapted Classifier

Unlabelled target data

Classified target data

Domain Adaptation by Instance Transfer

- Current training data set $\overline{TD}$: initialized by source data
- Classifier trained on source data


$^{0}\overline{\mathbf{w}}^{T} \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

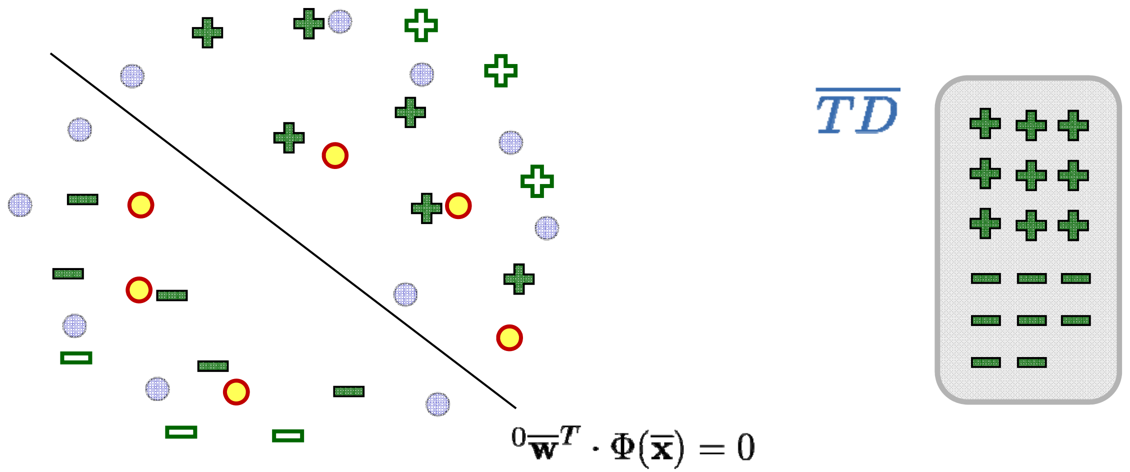
+ − labelled source samples

● unlabelled target samples

Domain Adaptation by Instance Transfer

- Domain adaptation: select samples to be added / removed

Iteration 1


$^0\overline{\mathbf{w}}^T \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

✚ ▬ labelled source samples

● unlabelled target samples

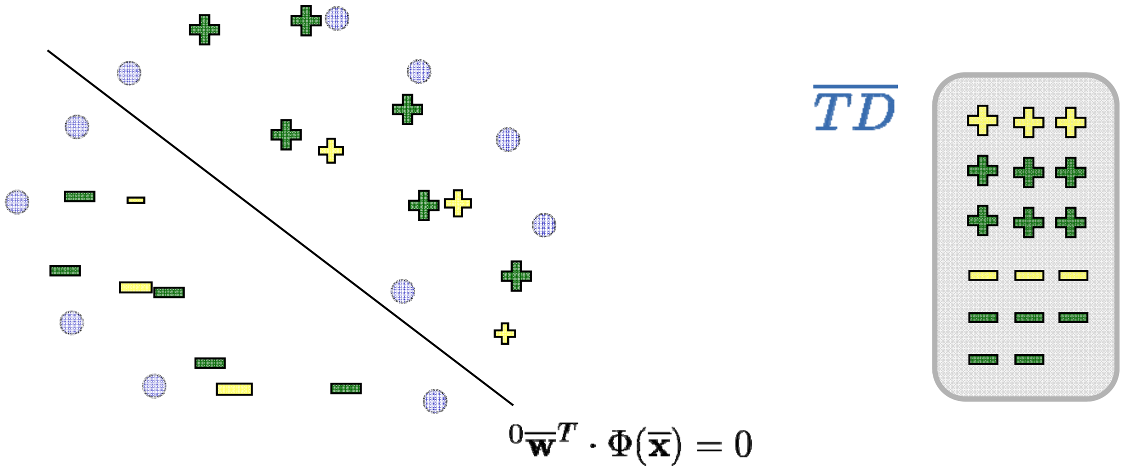
✙ ▭ source samples to be removed from $\overline{TD}$

○ target samples to be added to $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: new version of $\overline{TD}$

Iteration 1


$^0\overline{\mathbf{w}}^T \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

+ − labelled source samples

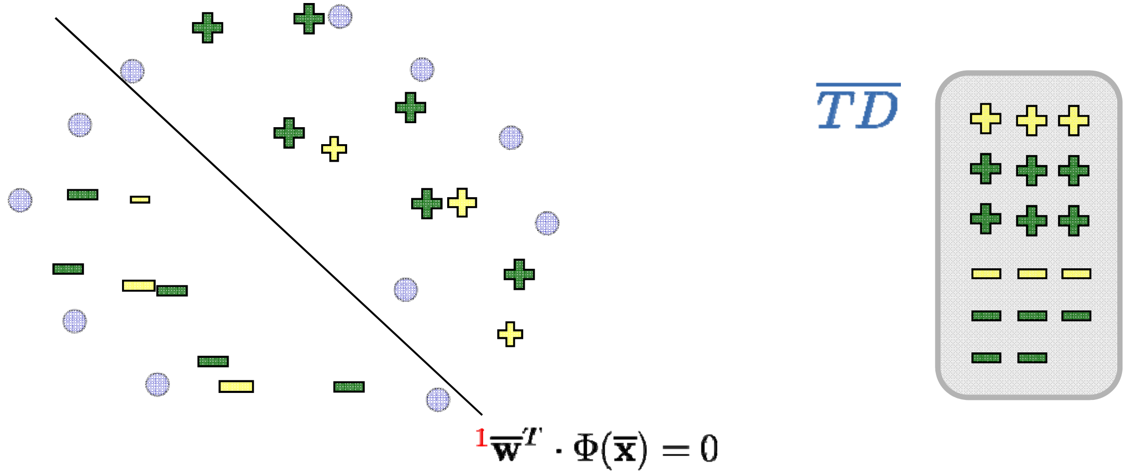
○ unlabeled target samples

+ − semi-labelled target samples in $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: train new classifier on $\overline{TD}$ / re-weighting

Iteration 1


$^{1}\overline{\mathbf{w}}^{T} \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

✚ ▬ labelled source samples

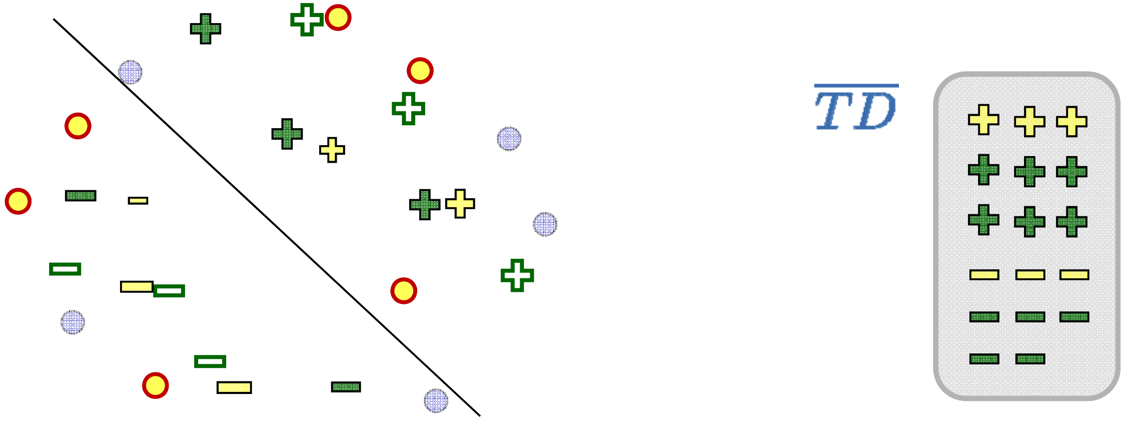
● unlabeled target samples

✚ ▬ semi-labelled target samples in $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: select samples to be added / removed

Iteration 2



✚ ▬ labelled source samples

● unlabelled target samples

✜ ▭ source samples to be removed from $\overline{TD}$

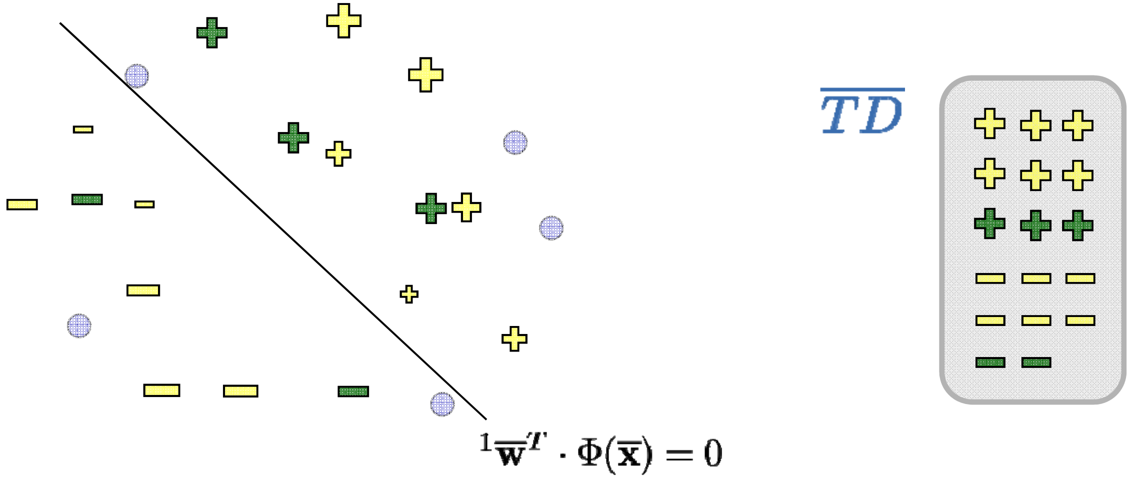
○ target samples to be added to $\overline{TD}$

✜ ▭ semi-labelled target samples in $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: new version of $\overline{TD}$

Iteration 2


${}^{1}\overline{\mathbf{w}}^{T} \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

✚ ▬ labelled source samples

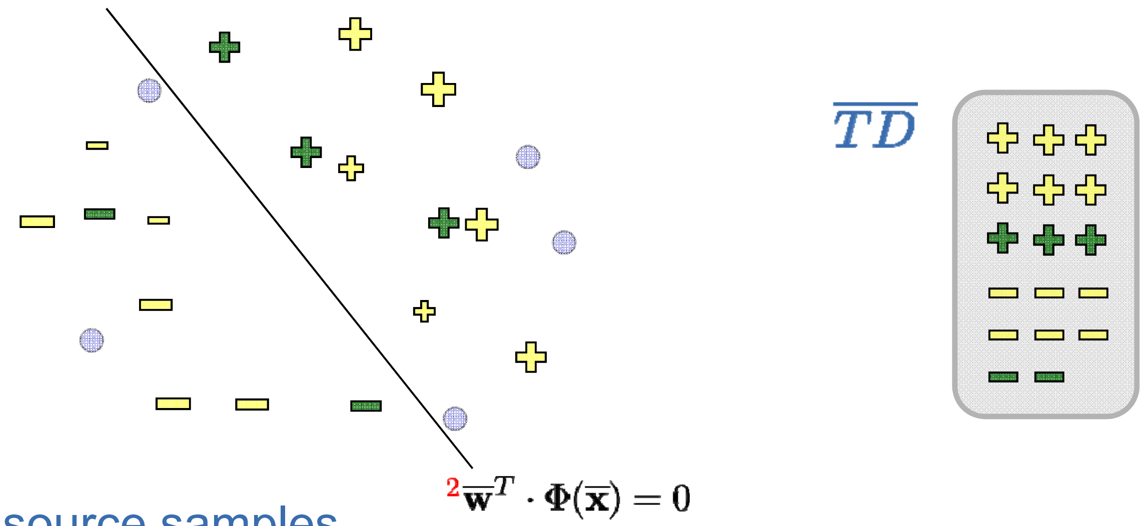
● unlabelled target samples

✚ ▬ semi-labelled target samples in $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: train new classifier on $\overline{TD}$ / re-weighting

Iteration 2


$^{2}\overline{\mathbf{w}}^{T} \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

✚ ▬ labelled source samples

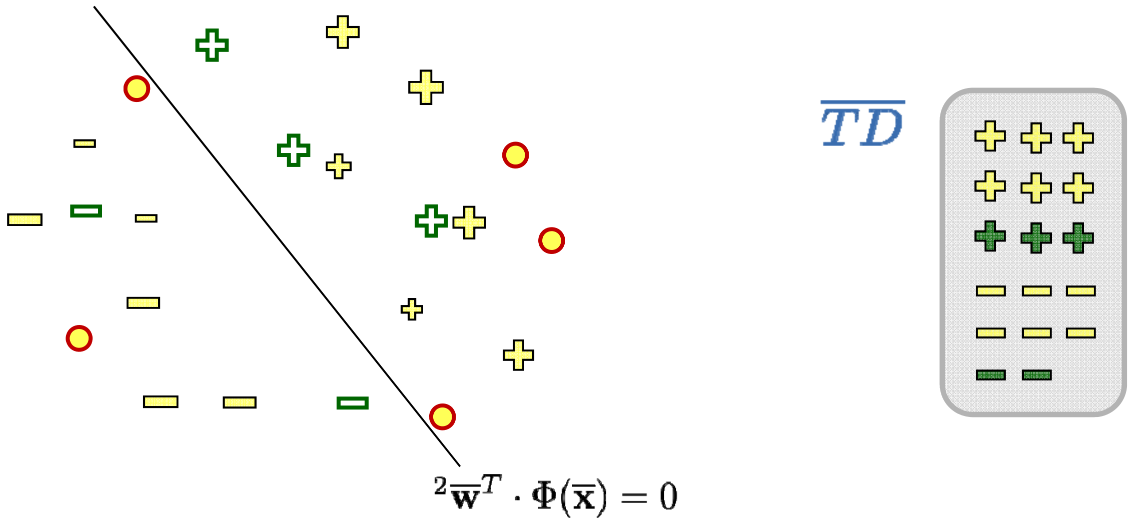
● unlabeled target samples

✚ ▬ semi-labelled target samples in $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: select samples to be added / removed

Iteration 3


$^{2}\overline{\mathbf{w}}^{T} \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

✚ ▬ source samples to be removed from $\overline{TD}$

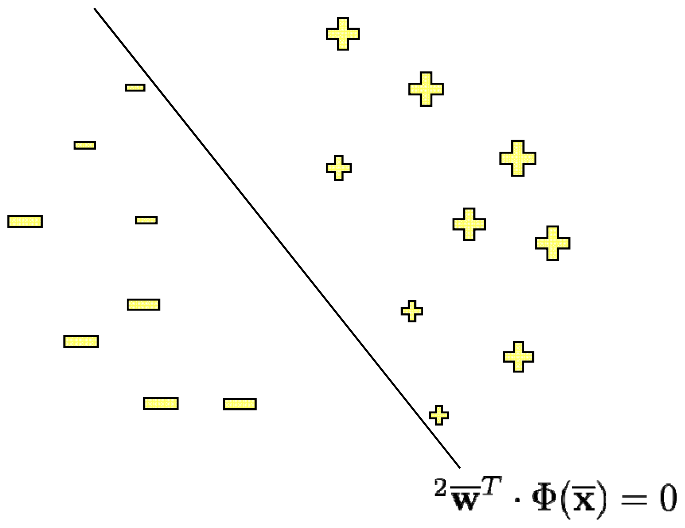
○ target samples to be added to $\overline{TD}$

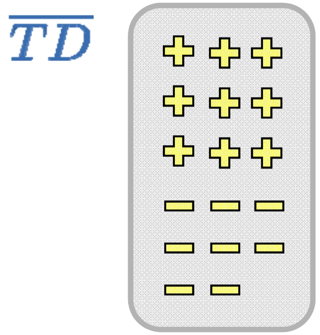
✚ ▬ semi-labeled target samples in $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: new version of $\overline{TD}$

Iteration 3


${}^{2}\overline{\mathbf{w}}^{T} \cdot \Phi(\overline{\mathbf{x}}) = 0$

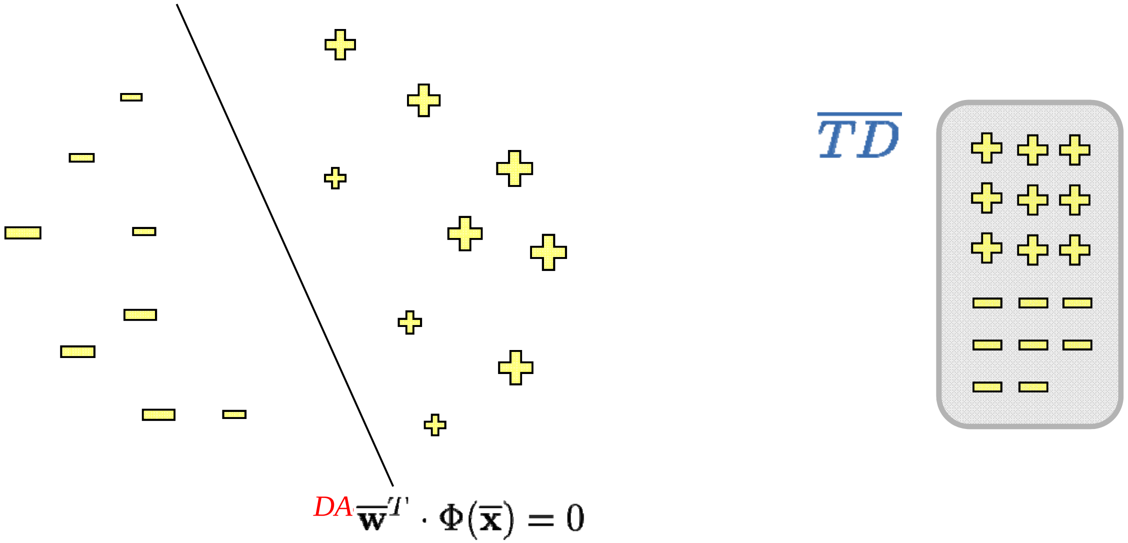


+ − semi-labelled target samples in $\overline{TD}$

Domain Adaptation by Instance Transfer

- Domain adaptation: train new classifier on $\overline{TD}$ / re-weighting

Iteration 3


${}^{DA}\overline{\mathbf{w}}^T \cdot \Phi(\overline{\mathbf{x}}) = 0$
$\overline{TD}$

+ − semi-labelled target samples in $\overline{TD}$

- No source domain samples in $\overline{TD}$ → **adapted classifier**

DA by Instance Transfer: Key Ingredients

- Base classifier: multiclass logistic regression

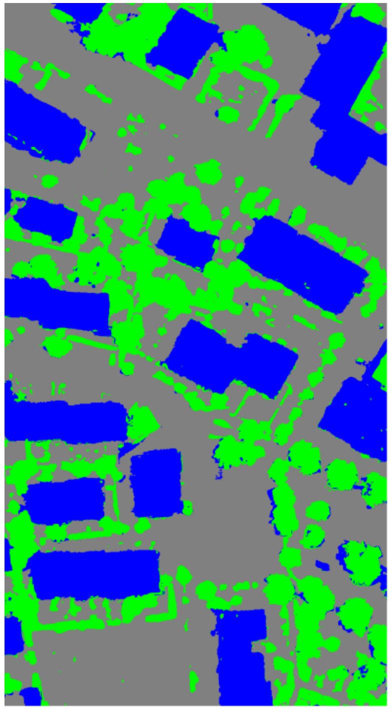
$$p\left(C = C^k \,|\, \mathbf{x}\right) = \frac{exp\left(\mathbf{w}_k^T \cdot \boldsymbol{\phi}(\mathbf{x})\right)}{\sum_j exp\left(\mathbf{w}_j^T \cdot \boldsymbol{\phi}(\mathbf{x})\right)}$$

model parameters **w**

- Criteria for sample selection:
  - Source samples to be removed: distance from decision boundary
  - Target samples to be added: distance from nearest points in $\overline{TD}$
- Definition of semi-labels: Current state of the classifier
- Sample weights in training: distance from decision boundary
- Regularization: previous state of the classifier [Paul et al., 2015; 2016]

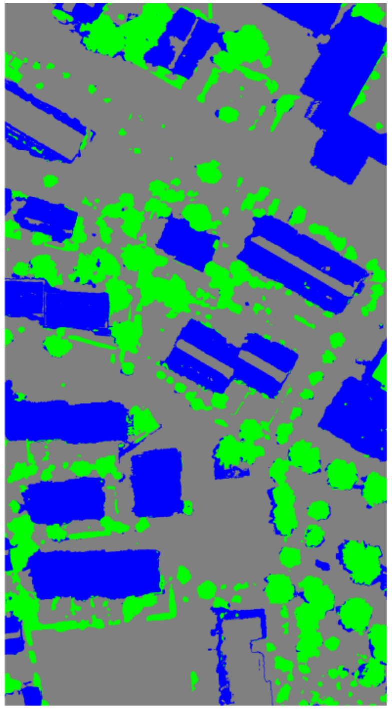
DA Example: Vaihingen Labelling Challenge

- Image and height data; evaluate overall accuracy (OA)



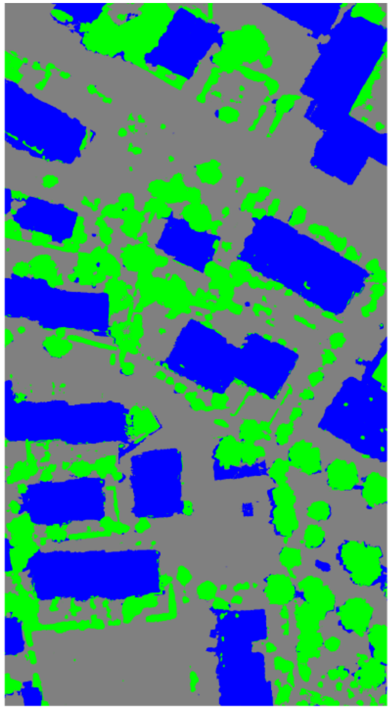
OA = 85.9 %

Training on target data
→ optimal case



OA = 80.9 %

Training on source data, no DA
5 % loss in OA



OA = 85.6 %

Result after DA

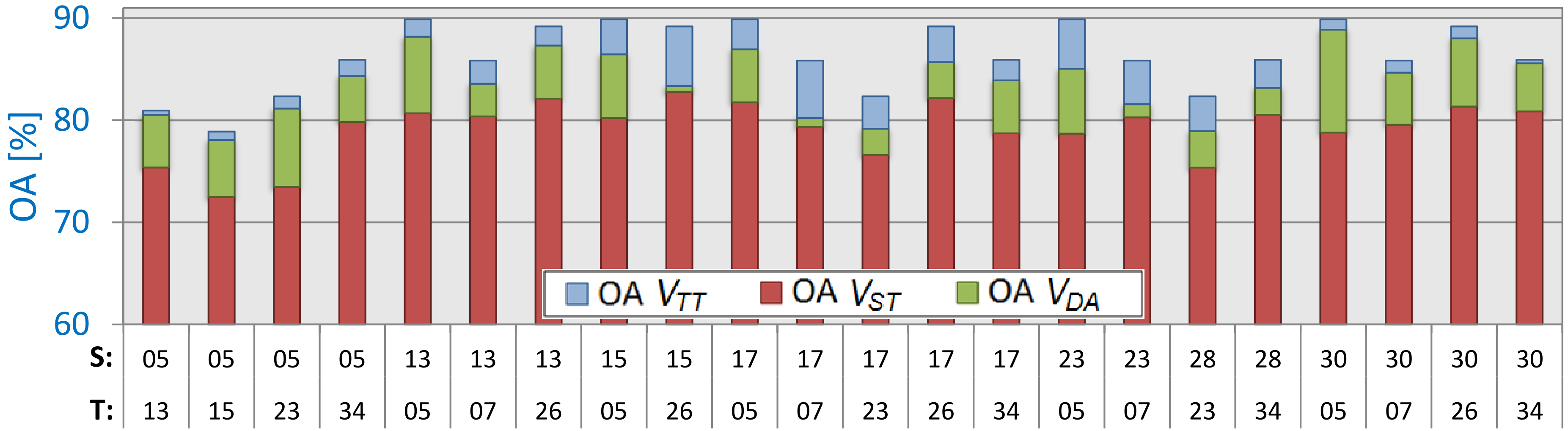
only 0.3 % loss

Results for target image:

ground
building
tree

DA Example: Cases with Positive Transfer

- Positive Transfer: 22 of 36 patch pairs (61% of test set)



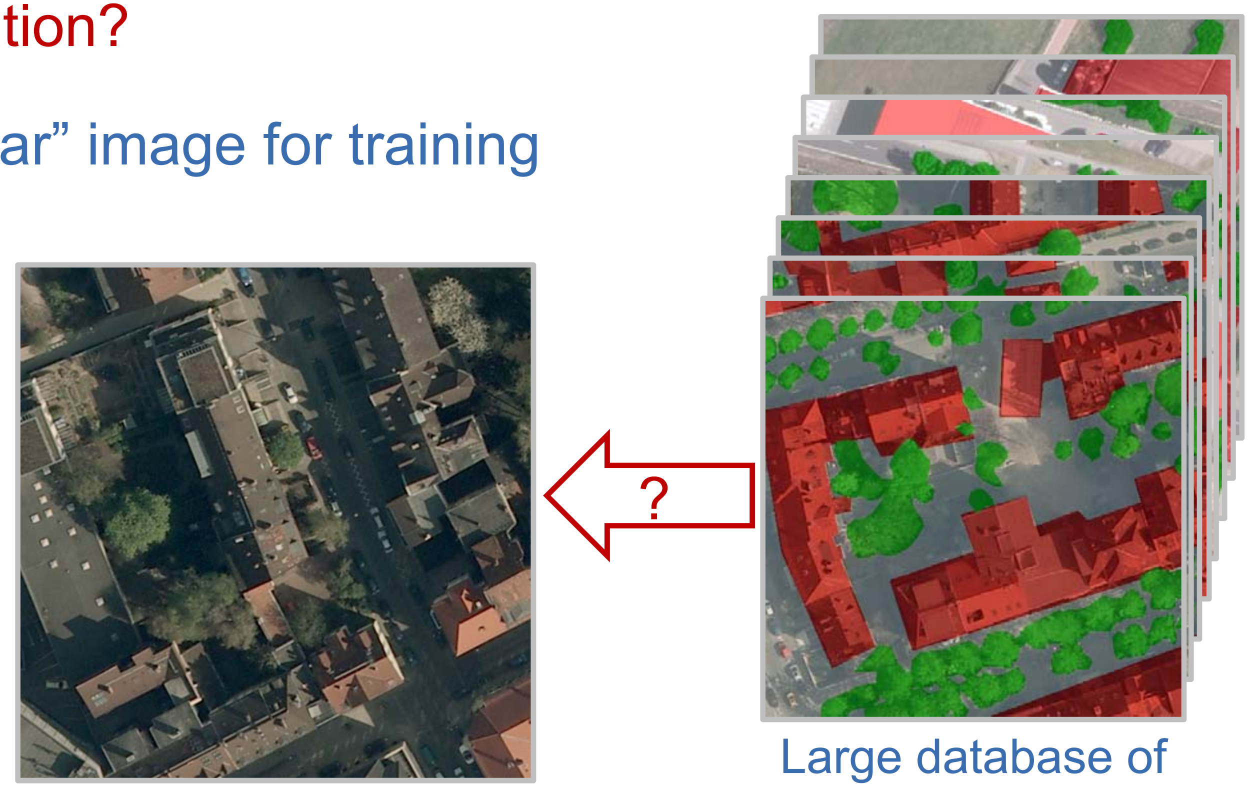
  - Green: compensation of loss in OA due to domain adaptation
  - Blue: remaining loss in OA after domain adaptation
  - Average improvement in OA over 22 test pairs: 4.7%
- 14 instances of negative transfer: average loss in OA of -3.7%

Outline

- Introduction
- Transfer Learning:
  - Domain adaptation by instance transfer
  - Creating a synthetic domain by source selection
- Training under label noise:
  - Using existing maps for training and classification
- Conclusion

Source Selection: Motivation

- Different scenario: assumes large data base of labelled images
- Which images from the database are suited as source domains for Domain Adaptation?
  - Use “most similar” image for training
  - Avoid negative transfer



Target image

Large database of labelled images

Source Selection: Distance Measures

- Source selection requires distance measure between distributions

- Two variants for such domain distances [Vogt et al., 2017]
  - Unsupervised: $d_{UDA} = 2\,\underbrace{d_{MMD}(\overline{TD}_T, \overline{TD}_S)}$

    Maximum Mean Discrepancy
    [Gretton et al., 2012]

  - Supervised: $d_{SDA} = d_{UDA} + \underbrace{\epsilon(h_S(x), \overline{TD}_S)}$

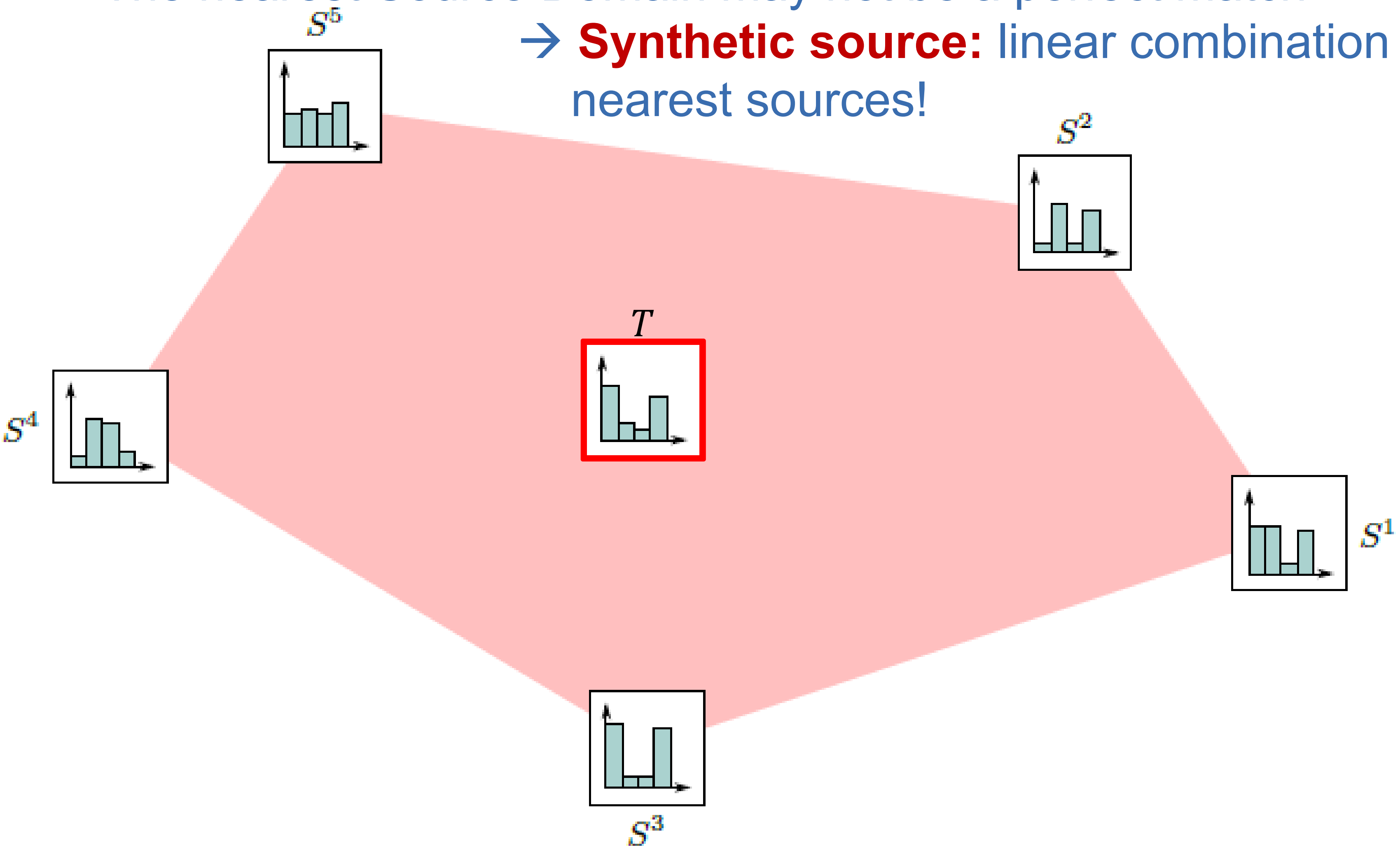
    Classification error in source domain

→ Optimal Source: $\bar{S} = \arg\min_{S \in \mathbb{S}} d_{\{SDA,UDA\}}$

Synthetic Source Generation

- The nearest Source Domain may not be a perfect match

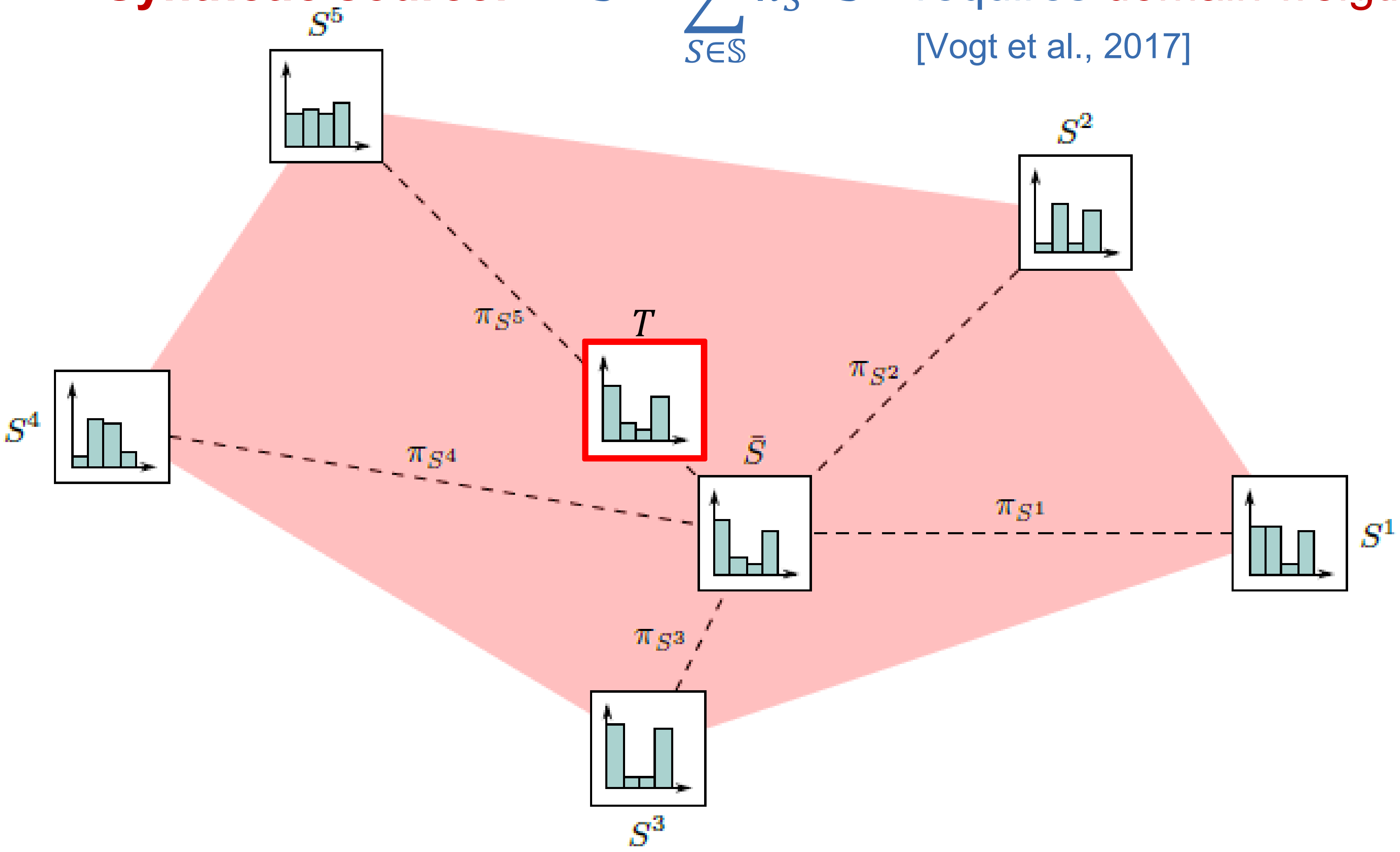
  → **Synthetic source:** linear combination of nearest sources!



Synthetic Source Generation

- **Synthetic source:** $\bar{S} = \sum_{S \in \mathbb{S}} \pi_S \cdot S$ requires domain weigthts π_S

[Vogt et al., 2017]



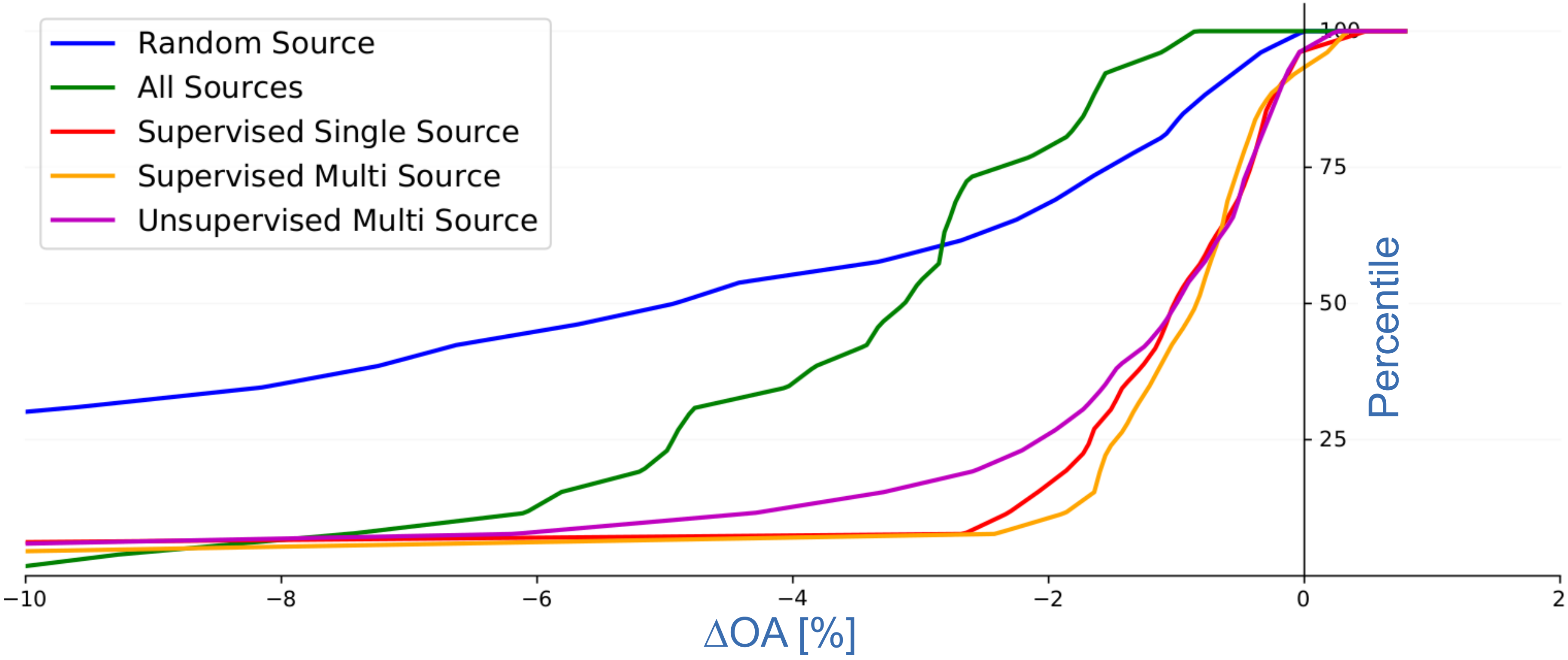
Source Selection: Experiments

- Compare different variants of source selection using aerial images from three German cities

- Measure difference in *Overall Accuracy* $\Delta\mathrm{OA}$ compared to using target labels

3CityDS

Source Selection: Results for 3CityDS



- Combined source selection + Domain Adaptation [Vogt et al., 2017]:
  - Synthetic source generation improves prospects for DA
  - Improvement due to DA is small but significant

Outline

- Introduction
- Transfer Learning:
  - Domain adaptation by instance transfer
  - Creating a synthetic domain by source selection
- Training under label noise:
  - Using existing maps for training and classification
- Conclusion

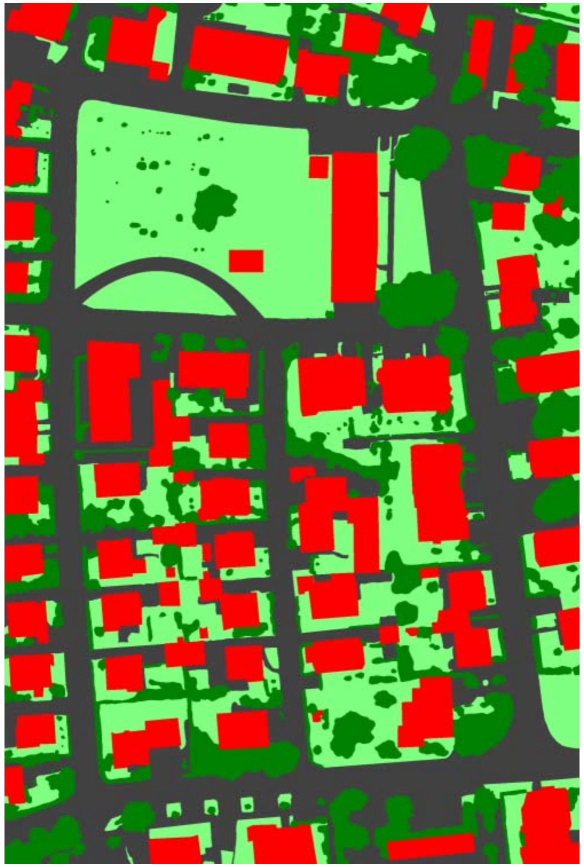
Learning under Label Noise: Motivation

- Topographic applications:
  - Maps do exist, but may be outdated
- **Observation: Most areas do not change over time**
  - Use existing map for deriving training labels
  - Leads to errors in the training labels (label noise)
    → Learning under label noise [Frénay & Verleysen, 2014]

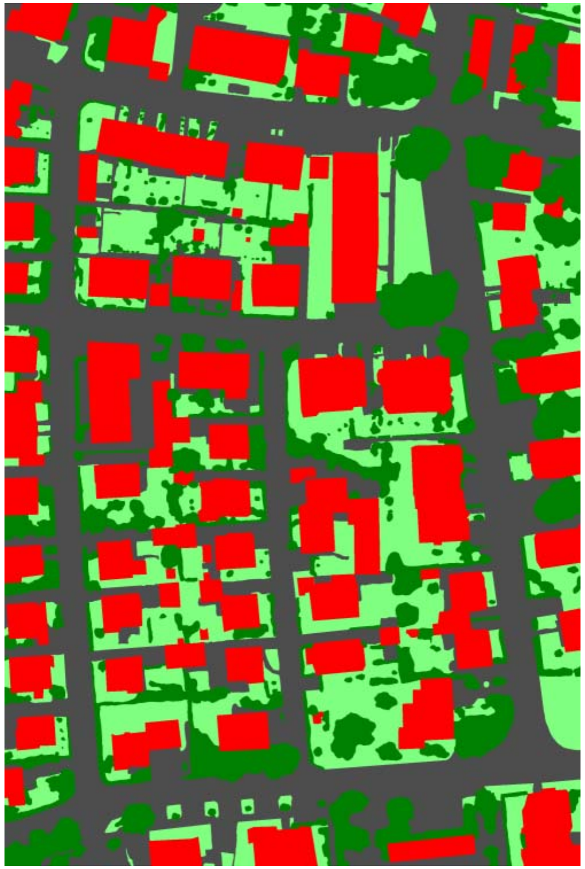
Learning under Label Noise: Motivation

ImageData
→ Features $\mathbf{x}$

Outdated map
→ Observed class labels $\underline{C}$



Updated map (wanted)
→ true class labels C



Label Noise Robust Logistic Regression

- Multiclass logistic regression

$$p\left(C = C^k \mid \mathbf{x}, \mathbf{w}\right) = \frac{exp\left(\mathbf{w}_k^T \cdot \boldsymbol{\phi}(\mathbf{x})\right)}{\sum_j exp\left(\mathbf{w}_j^T \cdot \boldsymbol{\phi}(\mathbf{x})\right)}$$

- Training:
  - Determine **w** so that $p\left(C = C^k \mid \mathbf{x}, \mathbf{w}\right)$ delivers the **true labels** *C*

- **Problem:** True class labels *C* are unknown in training

Label Noise Robust Logistic Regression

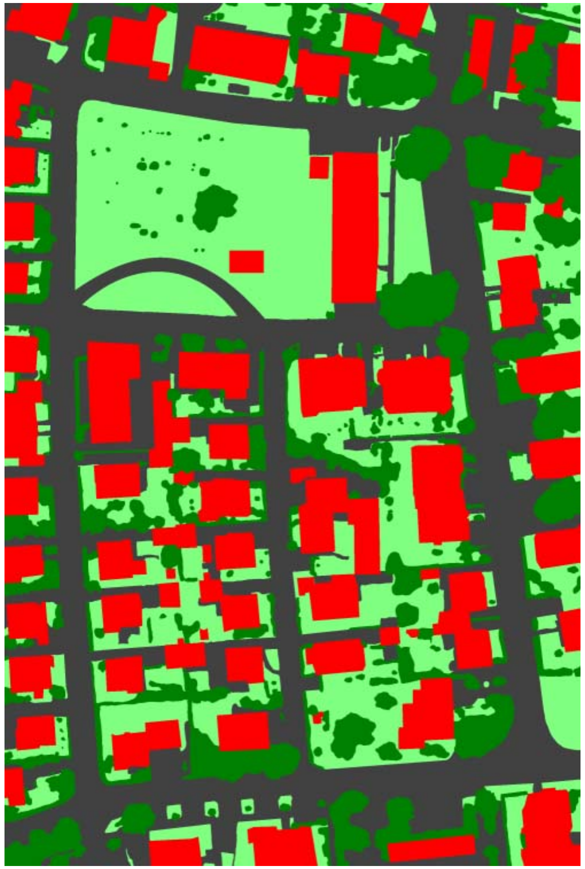
- Solution: Determine **w** from observed map labels $\underline{C}$ via $p(\underline{C} = C^k | \mathbf{x}, \mathbf{w})$:

$$p(\underline{C} = C^k | \mathbf{x}, \mathbf{w}) = \sum_a \underbrace{p(\underline{C} = C^k | C = C^a)}_{\text{Transition probability noise model}} \cdot \underbrace{p(C = C^a | \mathbf{x}, \mathbf{w})}_{\text{Posterior for true labels } C}$$

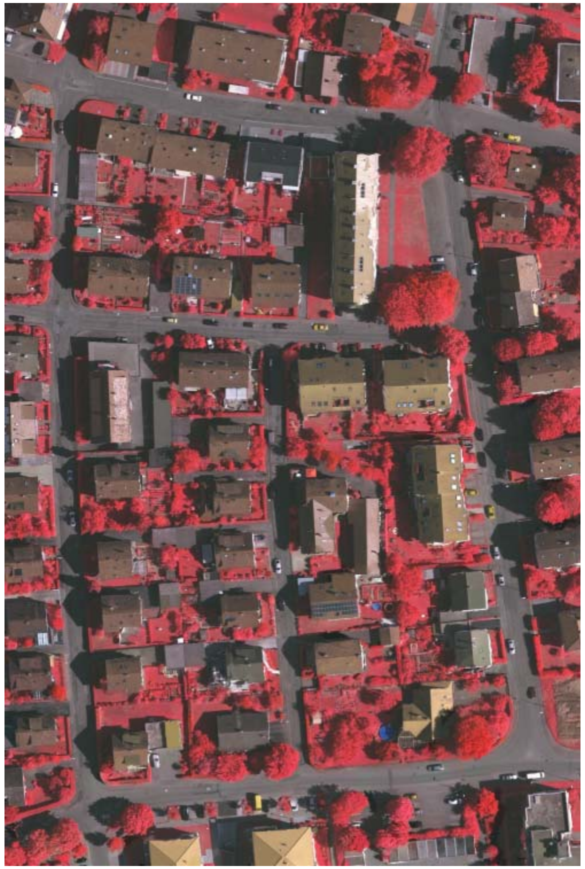
- Iterative training [Bootkrajang & Kabán, 2012; Maas et al., 2016]:
  - Parameters **w** of the classifier
  - Parameters of the **noise model**:
    Matrix Γ with $\Gamma_{ka} = p(\underline{C} = C^k | C = C^a)$

Experiments (Vaihingen Data): Simulated Changes

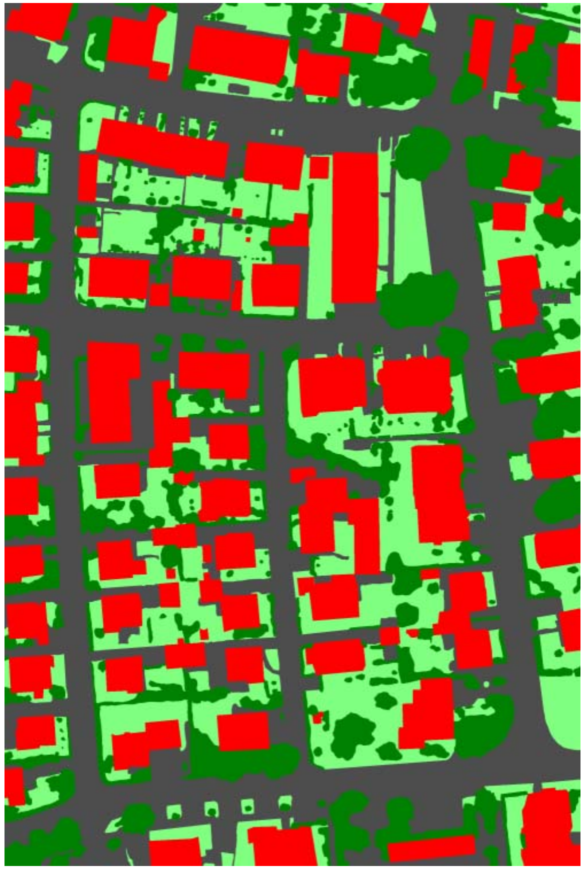
Outdated map



Orthophoto



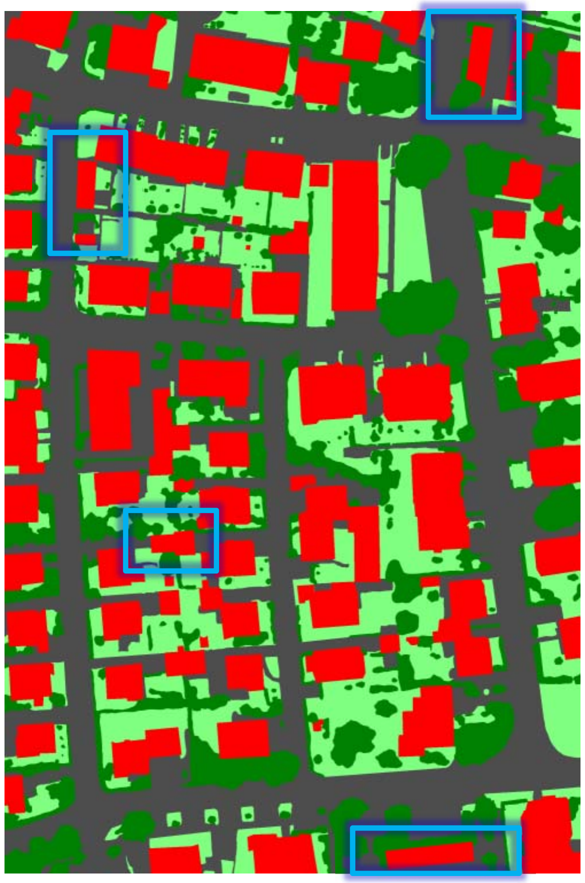
Reference

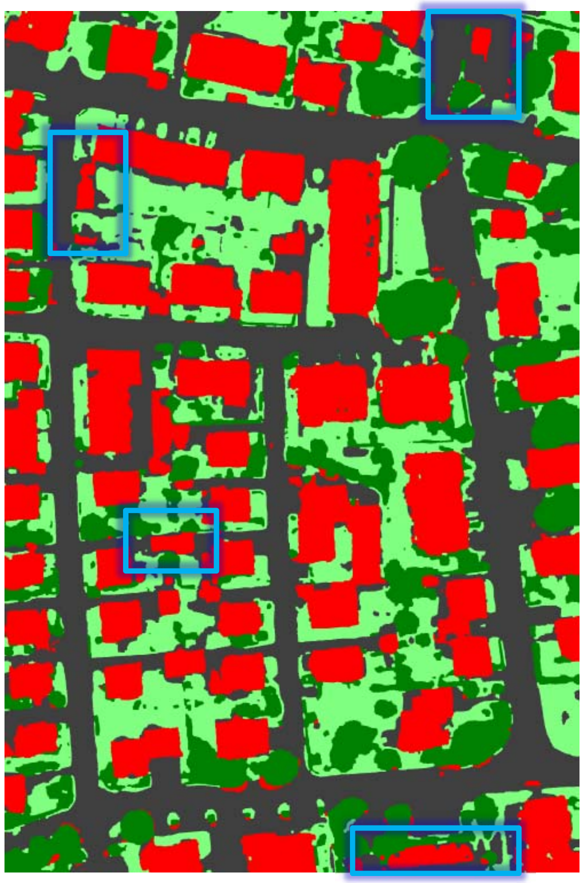


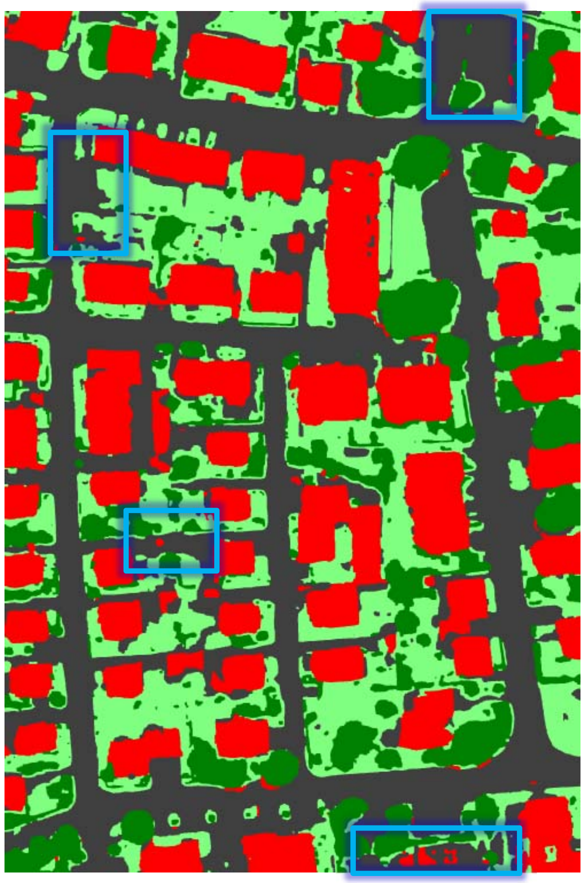
Experiments: Simulated Changes

[Maas et al., 2016]

- Reference LN (84.0% OA) MLR (81.9% OA)







Learning under Label Noise: Motivation

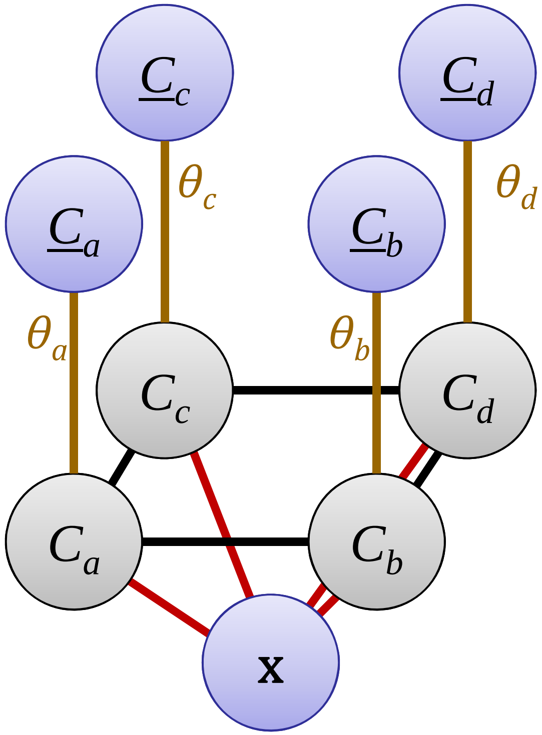
- Topographic applications:
  - Maps do exist, but may be outdated
- **Observation: Most areas do not change over time**
  - Use existing map for deriving training labels
  - Leads to errors in the training labels (label noise)
    → Learning under label noise [Frénay & Verleysen, 2014]
  - Use existing map as prior information in classification
  - Consider the fact that changes occur in clusters

Classification Considering the Existing Map

- Contextual classification: Conditional Random Field (CRF)
  [Kumar & Hebert, 2006]

- Simultaneous determination of all class labels C_a given
  - observed image data $\mathbf{x}$
  - observed class labels $\underline{C}_a$

- Maximisation of the joint posterior
  $p\left(\mathbf{C}|\,\mathbf{x},\underline{\mathbf{C}}\right)$



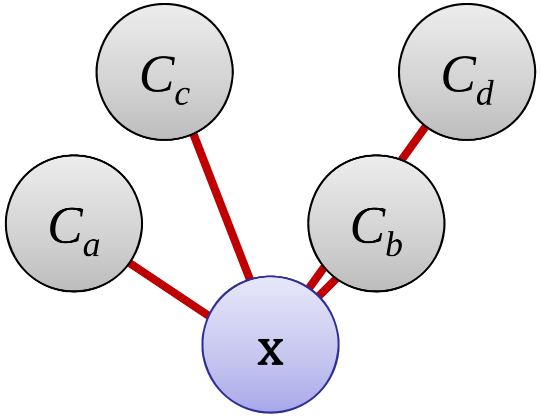
Factorisation of the Joint Posterior

- Factorisation of $p(\mathbf{C}|\,\mathbf{x},\underline{\mathrm{C}})$ according to the graphical model

$$p(\mathbf{C}|\,\mathbf{x},\underline{\mathrm{C}}) \propto \boxed{\prod_{\boldsymbol{n}} \varphi(C_n,\mathbf{x})} \cdot \prod_{\boldsymbol{n,m}} \psi(C_n, C_m, \mathbf{x}) \cdot \prod_{\boldsymbol{n}} \gamma^{\boldsymbol{\theta_n}}\left(\underline{C_n}, C_n\right)$$

  - Association potential —

  Label noise robust logistic regression



Factorisation of the Joint Posterior

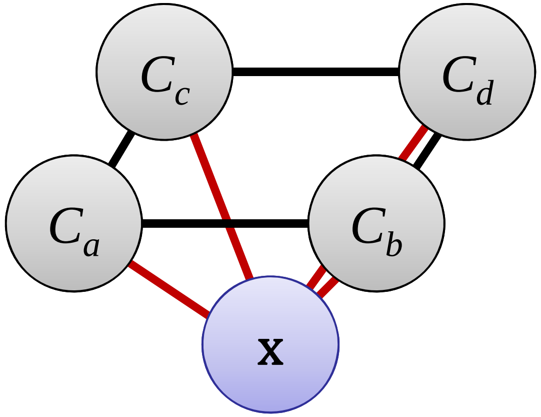
- Factorisation of $p(\mathbf{C}|\,\mathbf{x},\underline{\mathbf{C}})$ according to the graphical model

$$p(\mathbf{C}|\,\mathbf{x},\underline{\mathbf{C}}) \propto \prod_{n}\varphi(C_n,\mathbf{x}) \cdot \boxed{\prod_{n,m}\psi(C_n,C_m,\mathbf{x})} \cdot \prod_{n}\gamma^{\theta_n}\left(\underline{C}_n,C_n\right)$$

  - Association potential —
  - Interaction potential —

  Data-dependent smoothing

  [Boykov et al., 2001]



Factorisation of the Joint Posterior

- Factorisation of $p(\mathbf{C}|\,\mathbf{x},\underline{\mathbf{C}})$ according to the graphical model

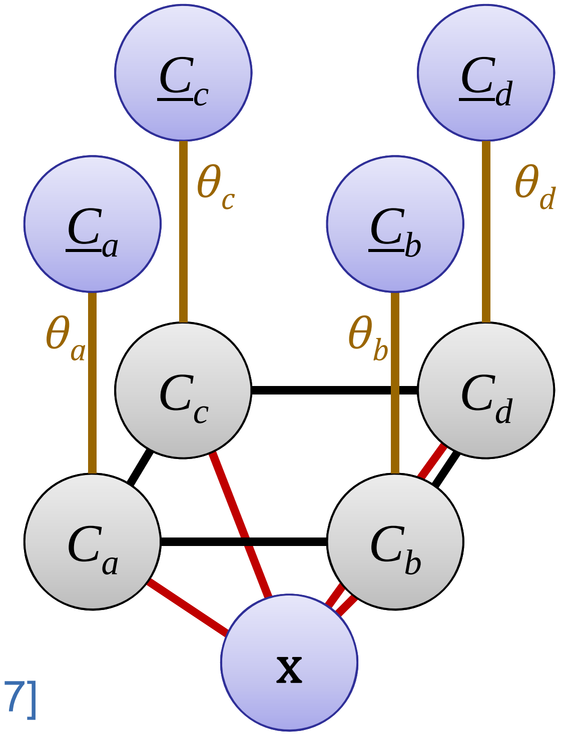
$$p(\mathbf{C}|\,\mathbf{x},\underline{\mathbf{C}}) \propto \prod_{n} \varphi(C_n,\mathbf{x}) \cdot \prod_{n,m} \psi(C_n,C_m,\mathbf{x}) \cdot \boxed{\prod_{n} \gamma^{\theta_n}\left(\underline{C_n},C_n\right)}$$

  - Association potential —
  - Interaction potential —
  - Temporal assoc. pot. —

    Labels from old map: observations

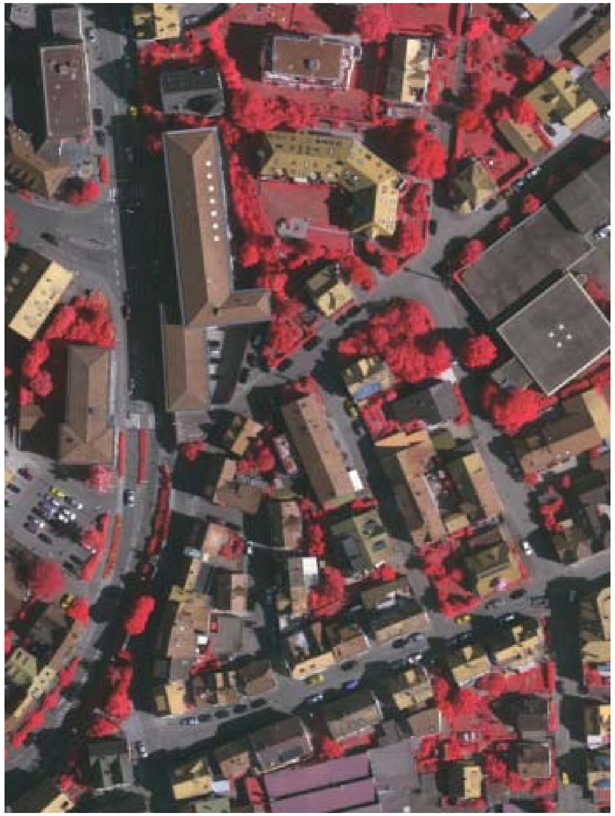
    Transition probabilities $p(C_n \,|\, \underline{C}_n)$

    Map weights θ_n: reduce weights in compact areas of change [Maas et al., 2017]

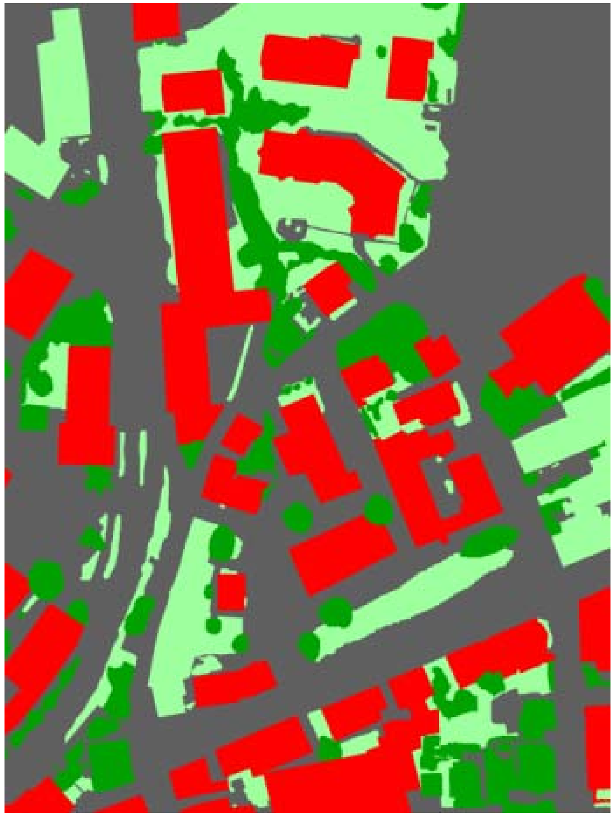


Example: Vaihingen, Patch 1

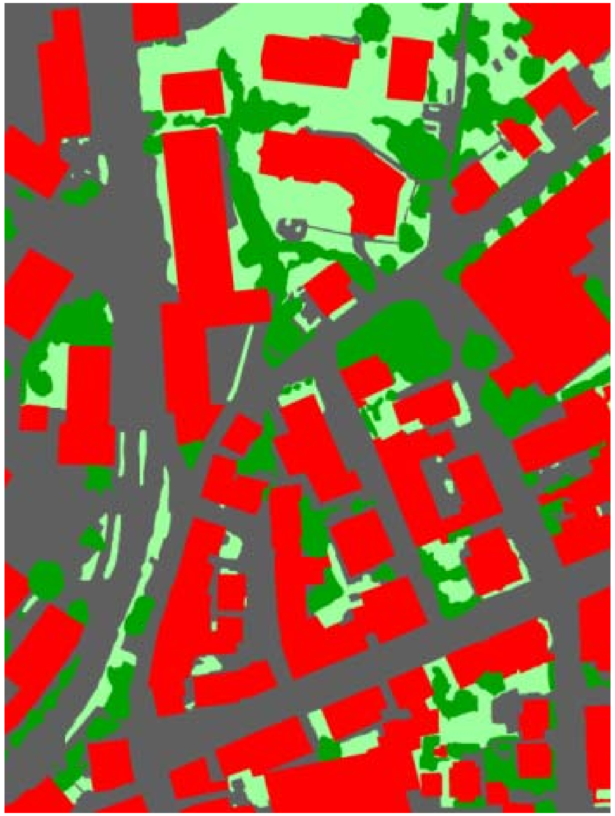
Orthophoto



Outdated map 3

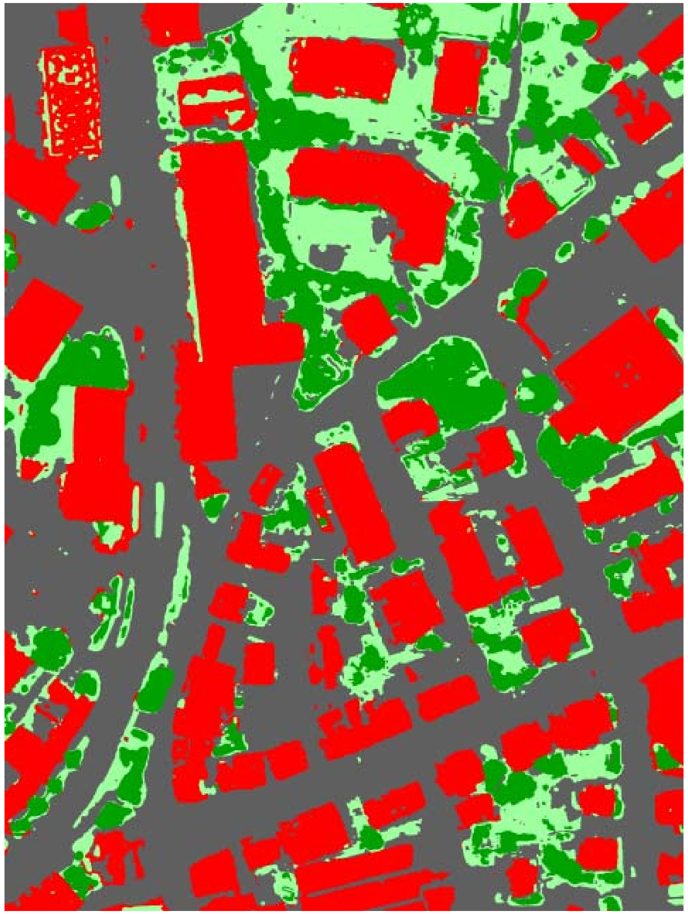


Reference



Example: Vaihingen, Patch 1

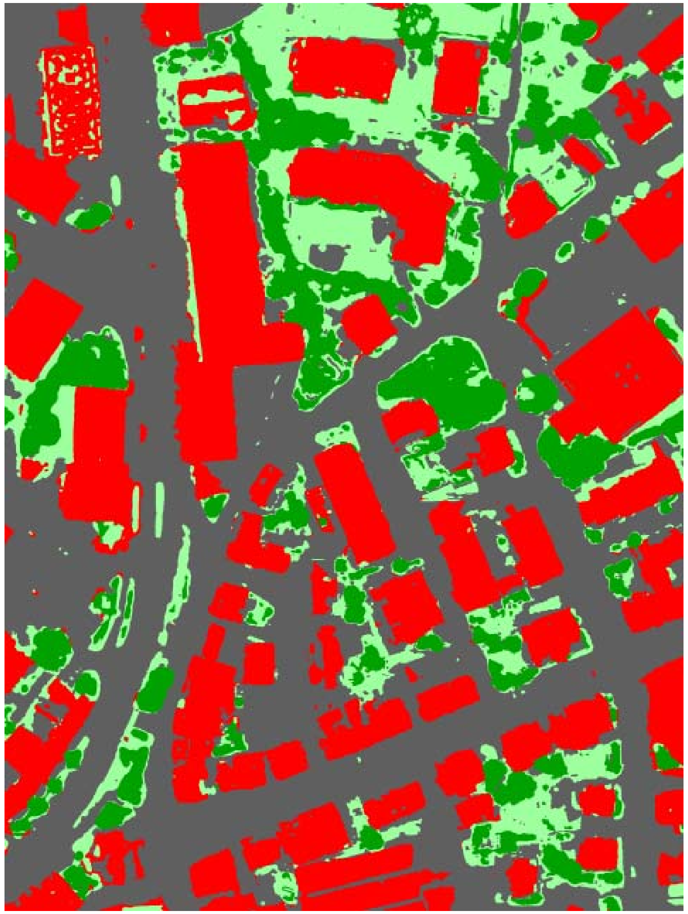
Init: Without iterative re-training and classification [Maas et al., 2016]



Overall Accuracy: 80.1 %

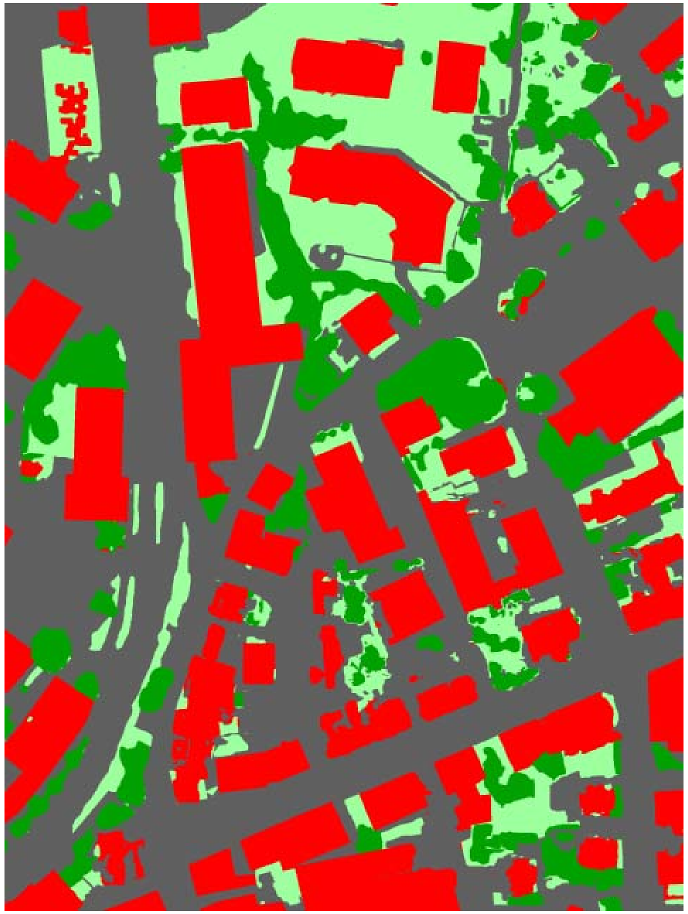
Example: Vaihingen, Patch 1

Init



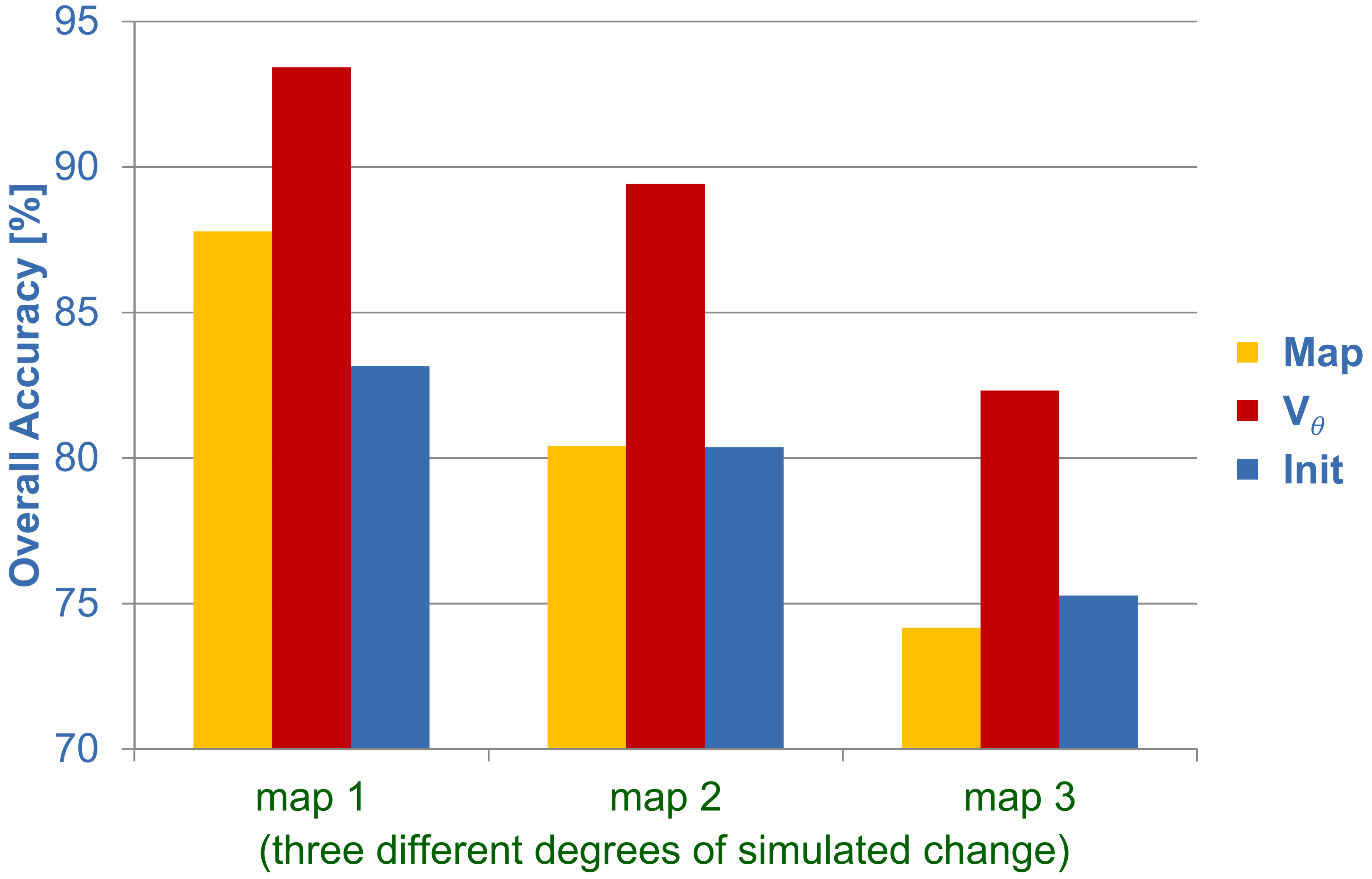
Overall Accuracy: 80.1 %

$\mathbf{V}_{\theta}$: Consider existing map [Maas et al., 2017]



Overall Accuracy: 88.5 %

Mean Overall Accuracy (Vaihingen)



Outline

- Introduction
- Transfer Learning:
  - Domain adaptation by instance transfer
  - Creating a synthetic domain by source selection
- Training under label noise:
  - Using existing maps for training and classification
- Conclusion

Conclusion

- Reduce efforts for manual generation of training data:
  - Domain adaptation:
    - ➢Can improve classification considerably
    - ➢Allows for limited degree of change only
  - Source selection
    - ➢Works well if a large pool of training data exists
    - ➢Scenario without such data needs to be investigated
  - Use existing maps for classification:
    - ➢No manual generation of training data at all
    - ➢Main limitation: New objects with unusual appearance

Future Work

- Deep neural networks (DNN) outperform other classifiers

  → Can similar principles be applied to DNN?

  – Transfer Learning: Representation transfer

    ➢ Usually requires target labels for retraining [Yosinski et al., 2014]

    ➢ First methods requiring no target labels:
    Deep Adaptation Networks [Long et al., 2015]

  – Learning under label noise:

    ➢ May be tackled by specific loss functions in training

    ➢ Example: road extraction using existing road database [Mnih & Hinton, 2012]

References I

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

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

Bruzzone, L., 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.

Frénay, B., Verleysen, M., 2014. Classification in the presence of label noise: a survey. IEEE Transactions on Neural Networks on Learning Systems 25(5):845–869.

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

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

Kumar, S., Hebert, M., 2006. Discriminative random fields. Int’l Journal of Computer Vision 68(2):179–201.

Long, M., Cao, Y., Wang, J., Jordan, M. I., 2015. Learning transferable features with deep adaptation networks. Proc. 32nd Int‘l Conf. on Machine Learning – Proceedings of Machine Learning Research, Vol. 37, pp. 97–105.

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

References II

Maas, A., Rottensteiner, F., Heipke, C., 2017. Classification under label noise using outdated maps ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-1/W1, pp. 215–222.

Mnih, V., Hinton, G., 2012. Learning to label aerial images from noisy data. Proc. 29th Int.'l Conference on Machine Learning, pp. 567-574.

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

Paul, A., Rottensteiner, F., 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.

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

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

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

Yosinski, j., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? Advances in Neural Information Processing Systems (NIPS) 27, pp. 3320–3328.