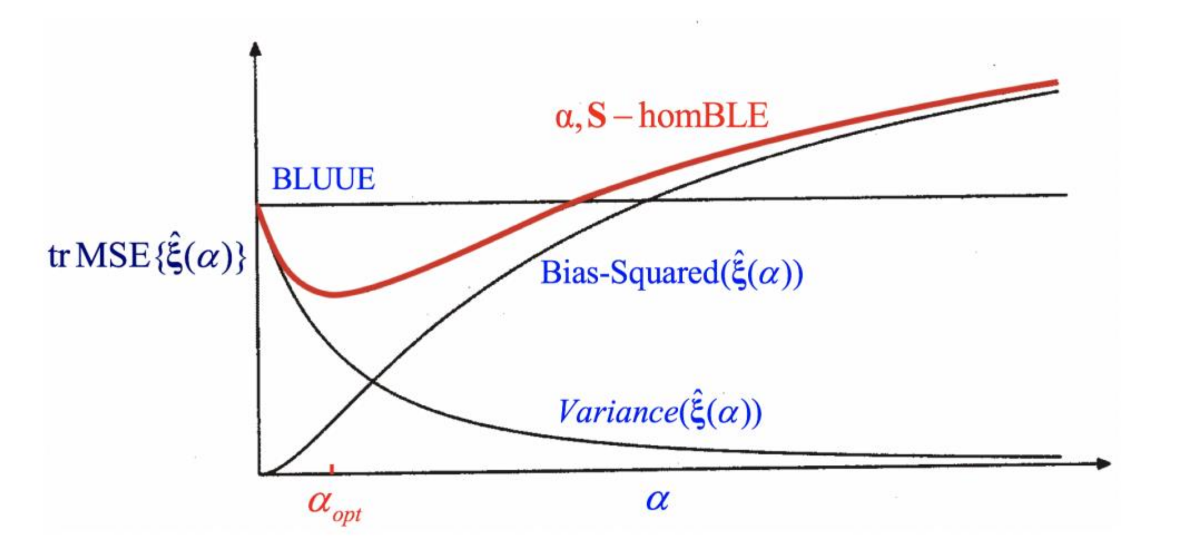
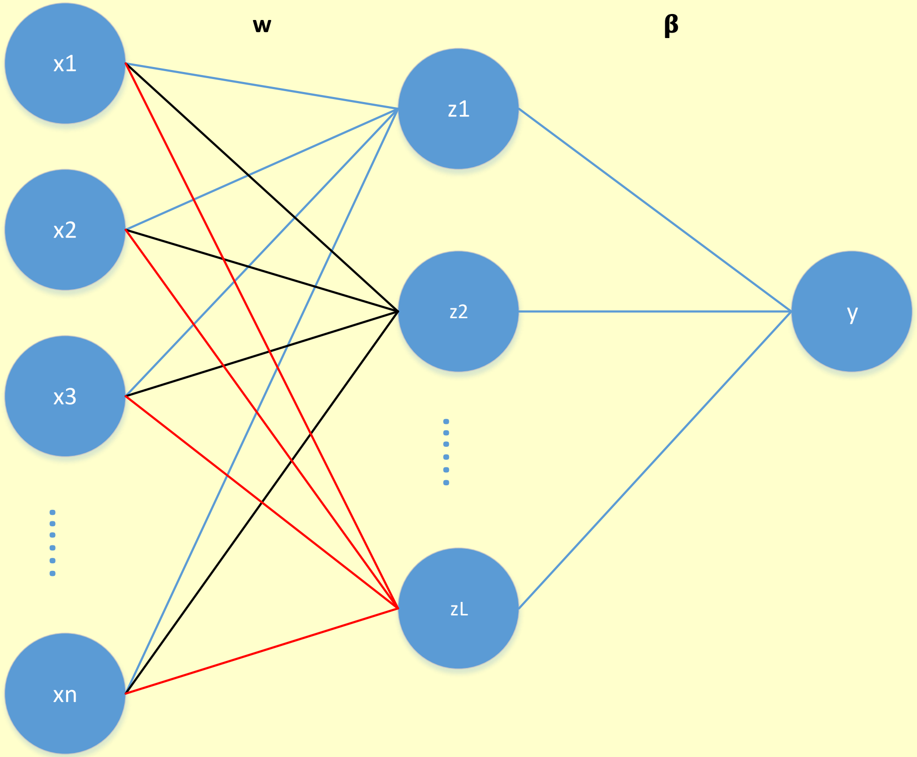
The Optimal Regularization and its Application in Extreme Learning Machine for Regression Analysis and Multiclass Classification

Qian Kun, Cai Jianqing, Lin Yi, Li Weijie, Nico Sneeuw





Contents

- Basic Theory of Extreme Learning Machine (ELM)
- Regularized ELM
- A-optimal design regularization
- Simulated Case: Approximation of “Sine Function”
- Real-World Regression Analysis
- Image Multiclass Classification
- Conclusion
- Outlook

Basic Theory of Extreme Learning Machine (ELM)

ELM is a newly developed single layer feedforward neural network (SLFN), proposed by Huang (2006). The model of ELM can be described as:

$$\underset{N\times L}{\mathbf{H}}\ \underset{L\times m}{\boldsymbol{\beta}} = \underset{N\times m}{\mathbf{Y}} \qquad \text{with}\quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \mathrm{M} \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L\times m},\ \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \mathrm{M} \\ \mathbf{y}_N^T \end{bmatrix}_{N\times m}$$

Where ***Y*** is the output matrix (vector), **β** is the connecting matrix between hidden layer and output layer, **H** is the hidden layer matrix (feature mapping matrix).

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_{11}) & \mathrm{L} & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_{1L}) \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_{N1}) & \mathrm{L} & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_{NL}) \end{bmatrix}_{N\times L}$$

w - connecting matrix between input layer and hidden layer

x - input matrix

b - bias matrix

The estimated solution of **β** based on least squares estimation is

$$\overset{\$}{\boldsymbol{\beta}} = \mathbf{H}^{+}\mathbf{Y}$$

H^{+} is the Moore-Penrose generalized inverse of the feature mapping matrix H.

Regularized ELM

ELM has fast training speed and shows high accuracy. But there exists two main problems for ELM.

1. Using Moore-Penrose generalized inverse to estimate the solution of $\widehat{\boldsymbol{\beta}}$ tend to generate an over-fitting model

$$L(\mathbf{H},\mathbf{Y};\boldsymbol{\beta}) = \mathrm{P}\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\,\mathrm{P}^2 = \min$$

2. Instability of solution of $\widehat{\boldsymbol{\beta}}$ because of ill-pose in normal matrix $\mathbf{N} = \mathbf{H}^{\mathbf{T}}\mathbf{H}$

In order to improve generalization performance and stability of ELM, regularization is brought in to penalizes the coefficients of weight matrix $\widehat{\boldsymbol{\beta}}$. The model of ELM with regularization is as follows:

$$L(\mathbf{H},\mathbf{Y};\boldsymbol{\beta},\lambda) = \mathrm{P}\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\,\mathrm{P}^2 + \lambda\,\mathrm{P}\boldsymbol{\beta}\,\mathrm{P}^2 \qquad (1)$$

In such case, the solution of $\widehat{\boldsymbol{\beta}}$ can be described as:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T + \lambda\mathbf{I})^{-1}\mathbf{Y} \qquad (2)$$

or

$$\boxed{\hat{\boldsymbol{\beta}} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{Y}} \qquad (3)$$

How to choose the expression of $\widehat{\boldsymbol{\beta}}$

a) Number of training samples N $<$ number of hidden layer nodes L, equation (2) is chosen

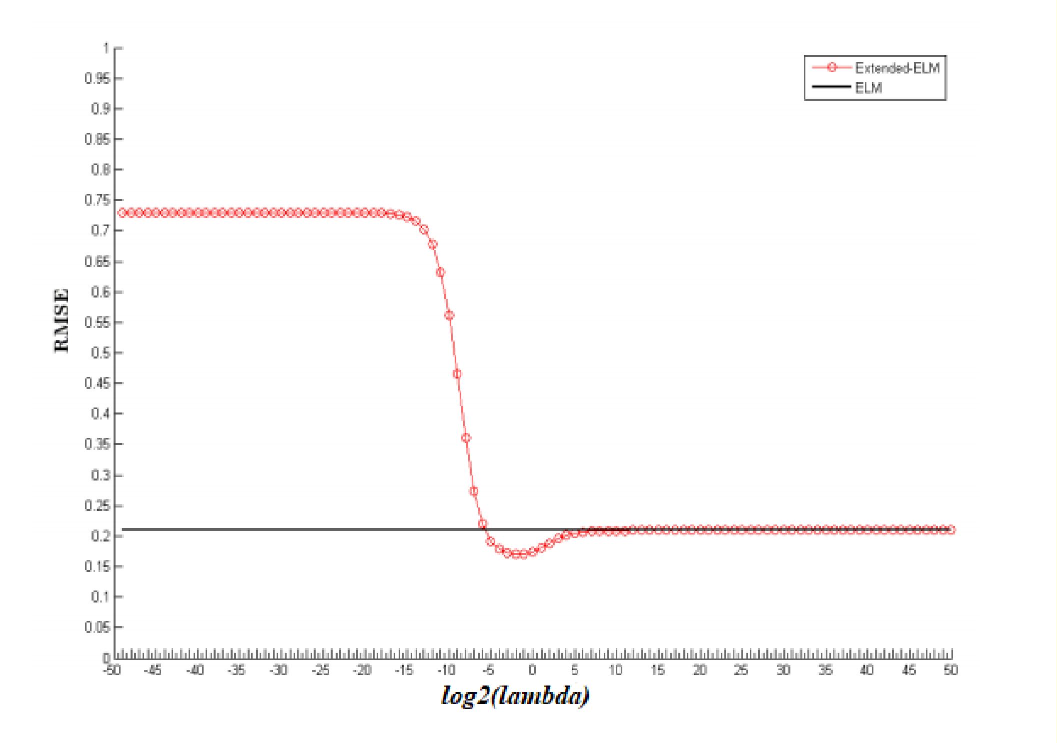
b) Number of training samples N $>$ number of hidden layer nodes L, equation (3) is chosen

In normal cases, we have sufficient training samples, so that we choose equation (3) as the solution of $\widehat{\boldsymbol{\beta}}$. But how to choose an optimal regularization parameter λ is still a problem.

Deng (2009) has proposed a heuristic method.

$$\lambda=[2^{-50}, 2^{-49}, L\ 0, L\ , 2^{49}, 2^{50}]$$

Cross-validation is used to choose a regularization parameter with minimum RMSE for validation set.



A-optimal design regularization

Cai (2004) proposed A-optimal design regularization. With A-optimal design regularization, the regularization parameter is determined by the minimum trace of mean square error (MSE) of $\widehat{\boldsymbol{\beta}}$.

$$\begin{aligned}\mathbf{MSE}\{\hat{\boldsymbol\beta}\} &:= \mathbf{E}\{(\hat{\boldsymbol\beta}-\boldsymbol\beta)(\hat{\boldsymbol\beta}-\boldsymbol\beta)'\} = \mathbf{E}\|\hat{\boldsymbol\beta}-\boldsymbol\beta\|^2 \\ &= \mathbf{E}[(\hat{\boldsymbol\beta}-\mathbf{E}\hat{\boldsymbol\beta})+(\mathbf{E}\hat{\boldsymbol\beta}-\boldsymbol\beta)]'[(\hat{\boldsymbol\beta}-\mathbf{E}\hat{\boldsymbol\beta})+(\mathbf{E}\hat{\boldsymbol\beta}-\boldsymbol\beta)]' \\ &= \mathbf{E}[(\hat{\boldsymbol\beta}-\mathbf{E}\hat{\boldsymbol\beta})'(\hat{\boldsymbol\beta}-\mathbf{E}\hat{\boldsymbol\beta})]+[(\mathbf{E}\hat{\boldsymbol\beta}-\boldsymbol\beta)][+(\mathbf{E}\hat{\boldsymbol\beta}-\boldsymbol\beta)]' \\ &= \mathbf{Cov}\{\hat{\boldsymbol\beta}\}+\mathbf{BB}'\end{aligned} \tag{4}$$

Where **Cov**($\widehat{\boldsymbol{\beta}}$) is Variance-Covariance matrix

$$\mathrm{Cov}(\hat{\boldsymbol\beta}) = (\mathbf{H}^T\mathbf{H}+\lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{H}(\mathbf{H}^T\mathbf{H}+\lambda\mathbf{I})^{-1} \tag{5}$$

and **B** is bias vector (matrix).

$$\begin{aligned}\mathbf{B} &= E(\hat{\boldsymbol\beta}-\boldsymbol\beta) \\ &= -[\mathbf{I}-(\mathbf{H}^T\mathbf{H}+\lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{H}]\boldsymbol\beta \\ &= \lambda(\mathbf{H}^T\mathbf{H}+\lambda\mathbf{I})^{-1}\mathbf{I}\boldsymbol\beta\end{aligned} \tag{6}$$

Then we can calculate MSE($\hat{\boldsymbol{\beta}}$)

$$\mathbf{MSE}(\hat{\boldsymbol{\beta}}) = (\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda\mathbf{I})^{-1}[\mathbf{H}^{\mathrm{T}}\mathbf{H} + (\lambda\mathbf{I})\boldsymbol{\beta}\boldsymbol{\beta}'(\lambda\mathbf{I})](\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda\mathbf{I})^{-1} \qquad (7)$$

The regularization parameter λ follows by A-optimal design in the sense of $\mathit{trace}\left(\mathbf{MSE}\right) = \min$

if and only if

$$\hat{\lambda} = \frac{\mathrm{trace}(\mathbf{H}^{\mathrm{T}}\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \hat{\lambda}\mathbf{I})^{-3})}{\boldsymbol{\beta}'(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \hat{\lambda}\mathbf{I})^{-2}\mathbf{H}^{\mathrm{T}}\mathbf{H}(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \hat{\lambda}\mathbf{I})^{-1}\boldsymbol{\beta}} \qquad (8)$$

The performance of ELM with A-optimal design regularization will be evaluated on 3 case studies.

Simulated Case: Approximation of "Sine Function"

Simulation 1 (without outliers):

Dataset: 10000 samples uniformly distributed in $(-10,\ 10)$ of sine function

Training data: 5000 samples with random noise distributed in $(-0.2,\quad 0.2)$

Testing data: other 5000 noise-free samples

Simulation without outliers		
	ELM	RELM
RMSE (training data)	0.1150	0.1157
RMSE (testing data)	0.0145	0.0151

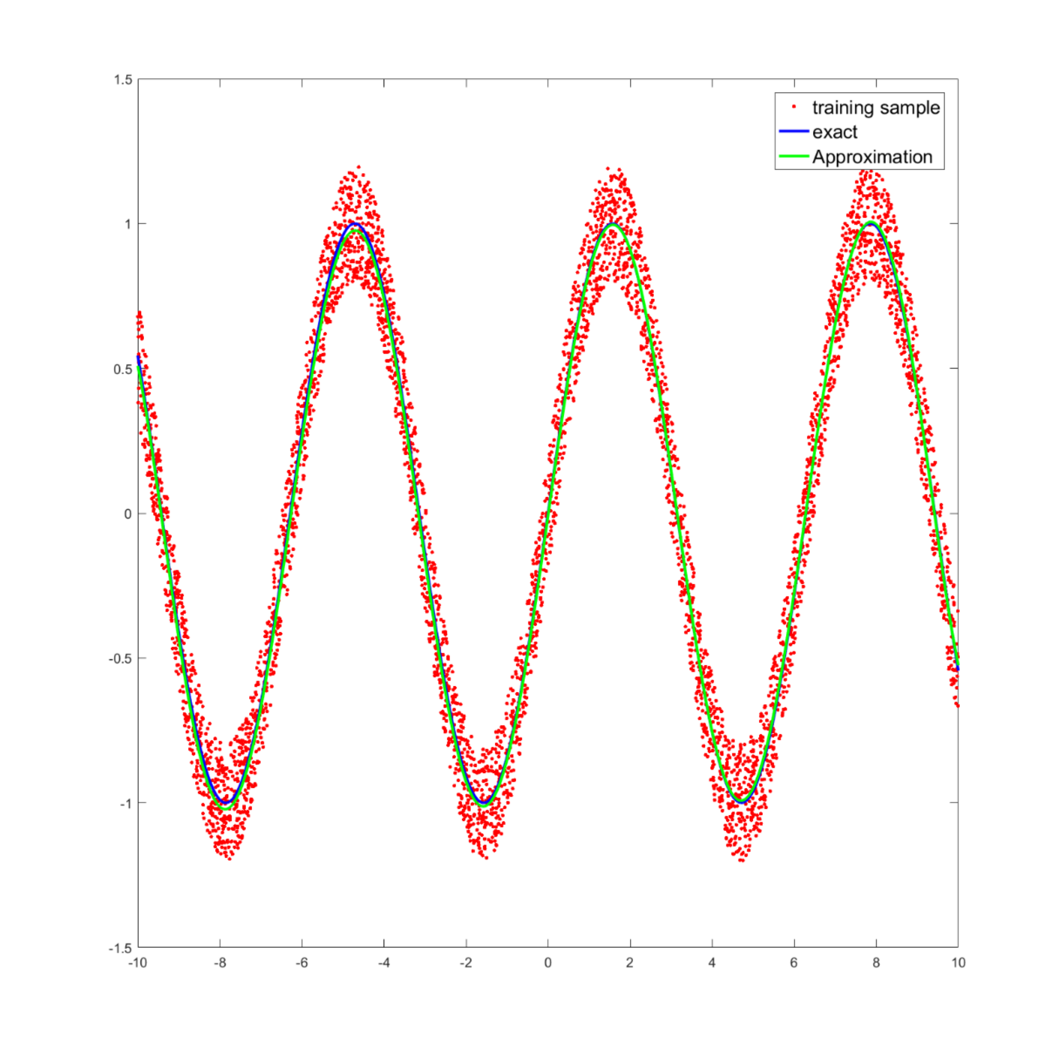
Simulation 2 (with outliers):

Dataset: 9900 samples uniformly distributed in $(-10{,}10)$ of sine function

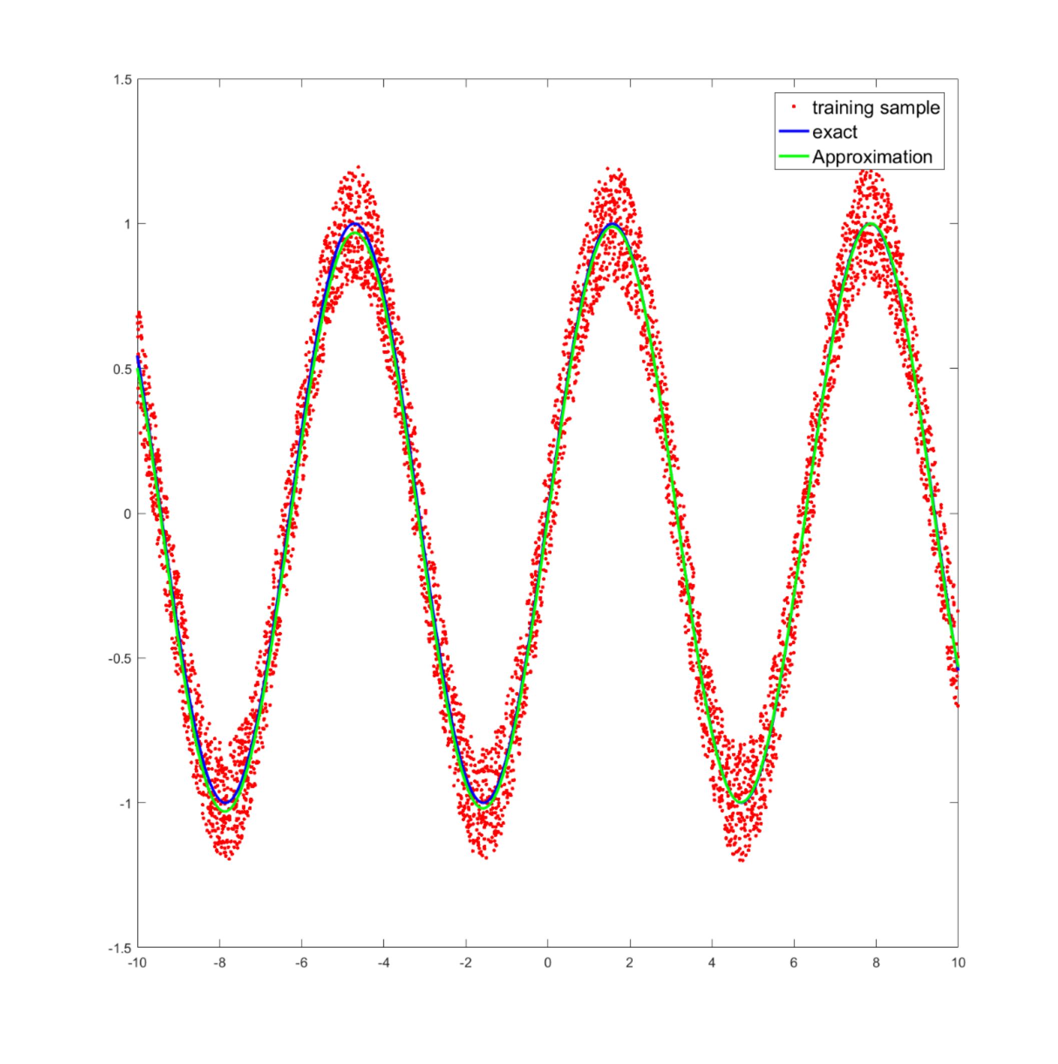
Training data: 4900 samples with random noise distributed in $(-0.2, 0.2)$ and 100 outliers distributed in $(-2,\ 2)$

Testing data: other 5000 noise-free samples

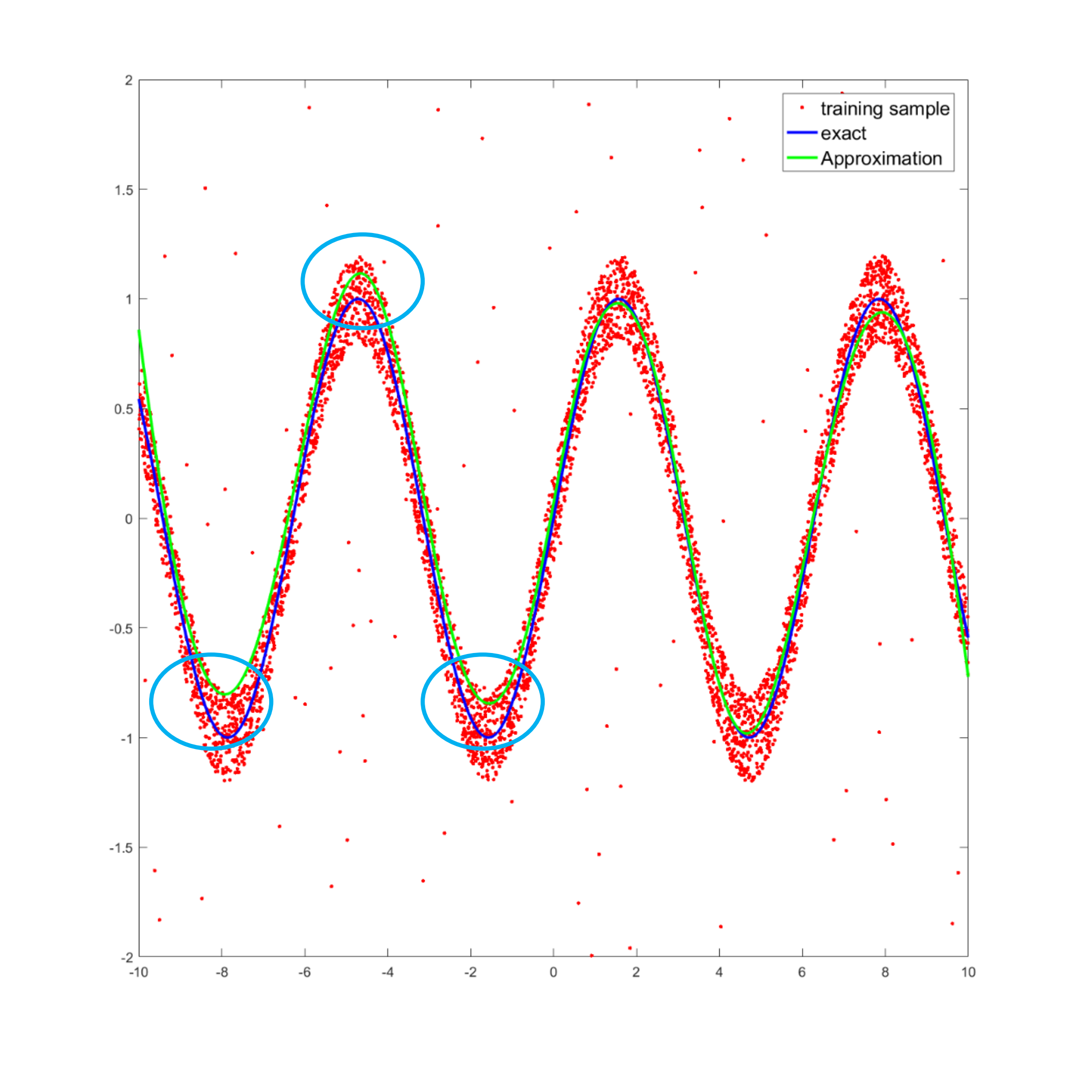
Simulation with outliers		
	ELM	RELM
RMSE (training data)	0.2566	0.2405
RMSE (testing data)	0.1006	0.0524



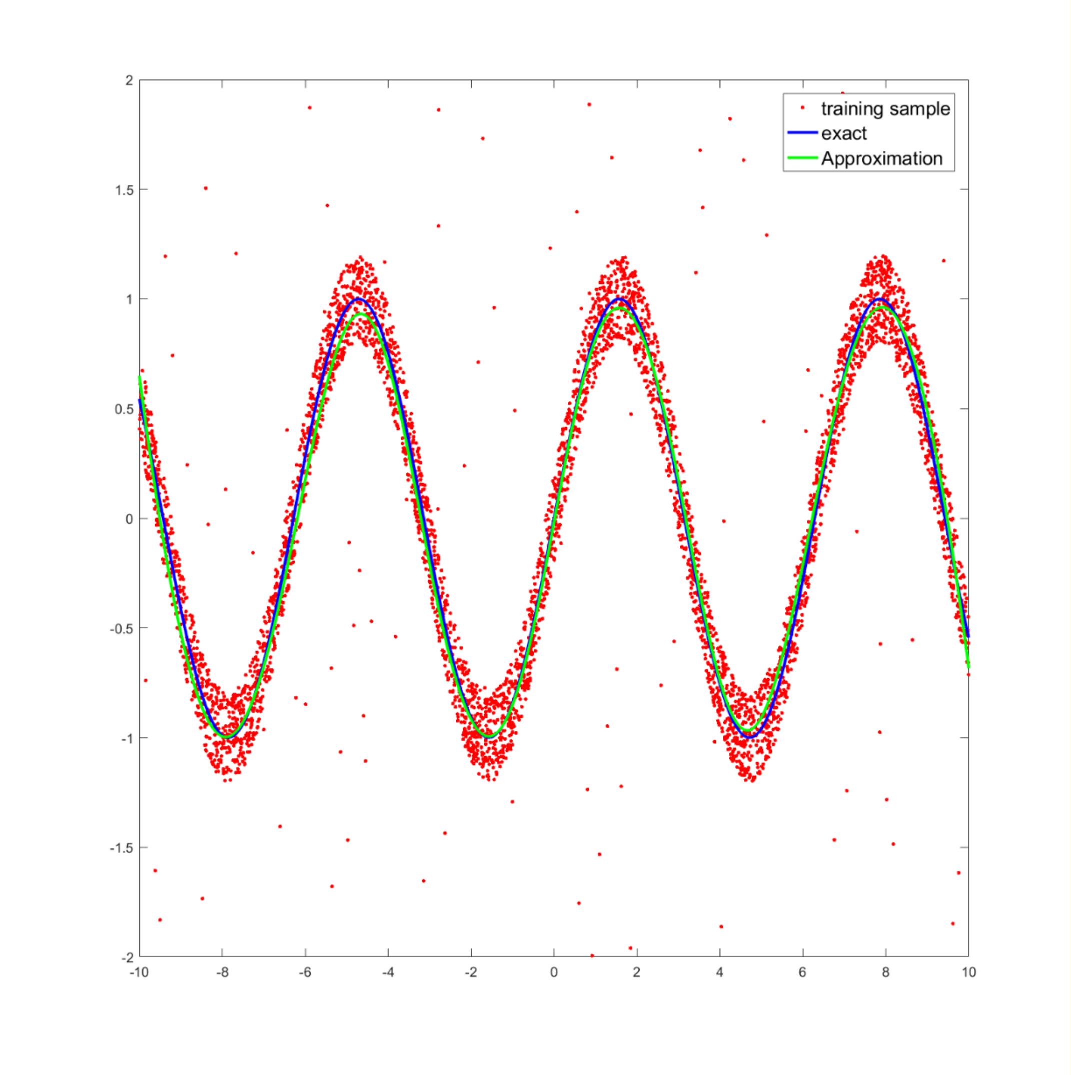
Approximation by ELM without outliers



Approximation by RELM without outliers



Approximation by ELM with outliers



Approximation by RELM with outliers

Real-World Regression Analysis

Data source:

http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html

An example of datasets: Bank domains

- Synthetically generated data from a simulation of how bank-customers choose their banks.
- 32 numerical features as input and 1 numerical decision as output
- 8192 samples, 4500 for training and 3692 for testing

Datasets	Training data	Testing data	Feature
Bank domains	4500	3692	32
Puma	4499	3693	32
Triazines	124	62	60
Pyrim	49	25	27
Machine CPU	139	70	6
Kinematic	5461	2731	8
California housing	13760	6880	8
Stocks domain	633	317	9
Fried_delve	27179	13589	10

Testing results: comparison of RMSE between ELM and RELM

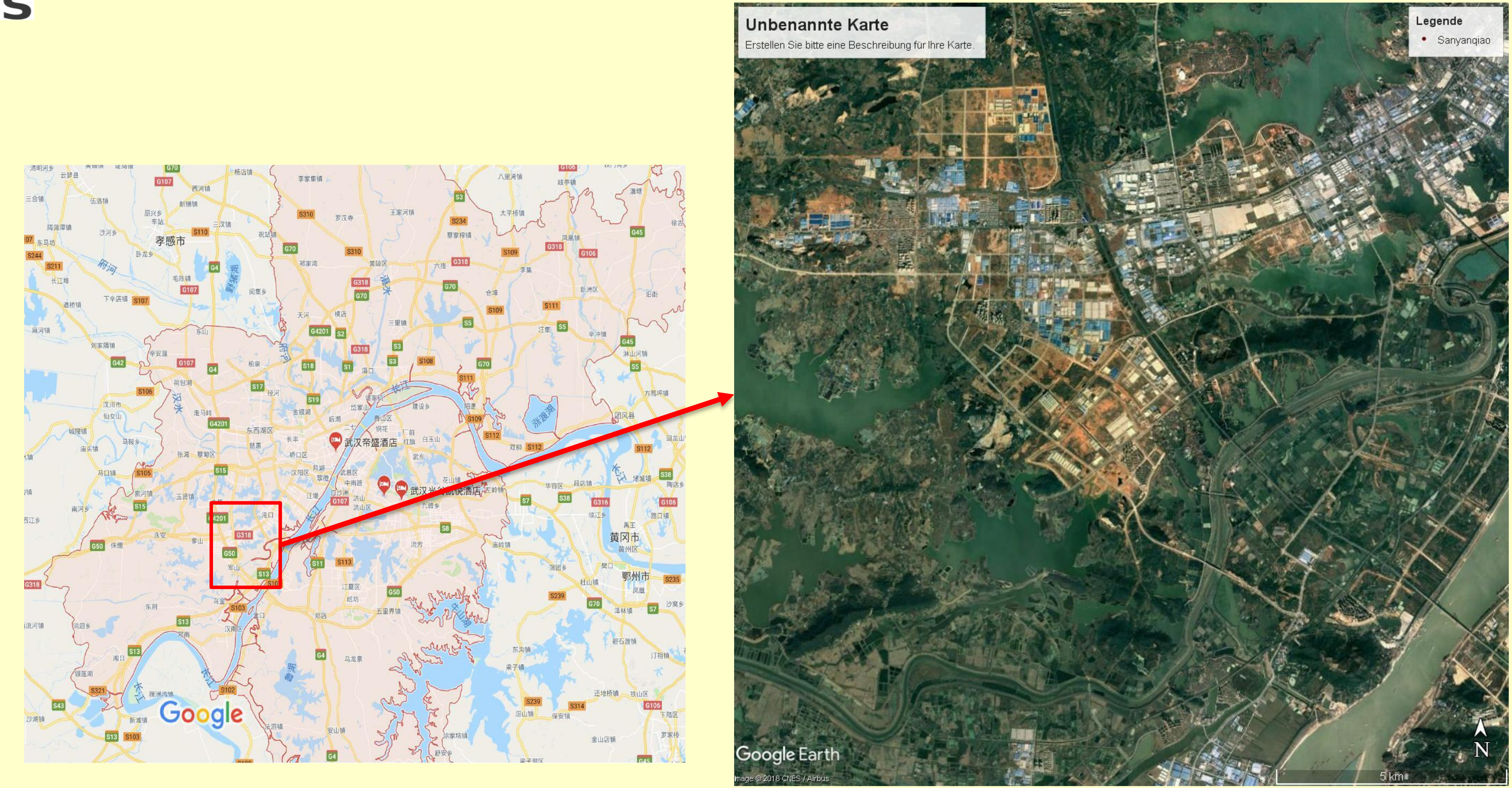
Dataset	**RMSE**			
	Training data		Testing data	
	ELM	RELM	ELM	RELM
Bank domains	0.0795	0.0806	0.0901	0.0819
Puma	0.0245	0.0248	0.0296	0.0251
Triazines	0.1479	0.1494	0.1661	0.1391
Pyrim	0.0776	0.0780	0.1004	0.0876
Machine CPU	0.0461	0.0506	0.0594	0.0511
Kinematic	0.0891	0.0903	0.1021	0.0968
California housing	0.1221	0.1246	0.1256	0.1251
Stocks domain	0.0297	0.0311	0.0396	0.0316
Fried_delve	0.1976	0.2011	0.3169	0.2466

Image Multiclass Classification

Data source: https://glovis.usgs.gov/
Study area: a part of Wuhan, China
Data: Landsat 8 satellite image
Resolution of image: 30m × 30m
Image size: 598 × 597 pixels
Feature: 7 spectral bands
5 classes: grass, tree, bare land, building, water
3550 labeled pixels as samples.

Number of labeled pixels for each class					
building	grass	tree	bare land	water	total
750	569	512	989	730	3550

Training data:
Each class: 100 randomly chosen pixels
Testing data:
Other 3050 pixels.



Unbenannte Karte

Erstellen Sie bitte eine Beschreibung für Ihre Karte.

Konfusionsmatrix

Original ELM:

		Class from reference data				
		Water	Bare land	Tree	Grass	Building
Class from classification	Water	730	0	0	0	0
	Bare land	80	890	1	7	11
	Tree	31	2	156	323	0
	Grass	0	0	0	569	0
	Building	31	7	0	0	712

A-optimal design regularized ELM:

		Class from reference data				
		Water	Bare land	Tree	Grass	Building
Class from classification	Water	730	0	0	0	0
	Bare land	35	941	2	5	6
	Tree	7	14	384	107	0
	Grass	0	0	1	568	0
	Building	13	7	0	7	723

	Accuracy	**Cohens kappa coefficient (κ)**
Original ELM	84.11%	0.7596
Regularized ELM	93.16%	0.9189

$$\text{accuracy} = \frac{\text{number of correct pixels}}{\text{number of total pixels}}$$

$$\kappa = \frac{p_0 - p_c}{1 - p_c}$$

with p_0 – the relative observed agreement

$$p_0 = \frac{1}{N} \sum_i K_{ii}$$

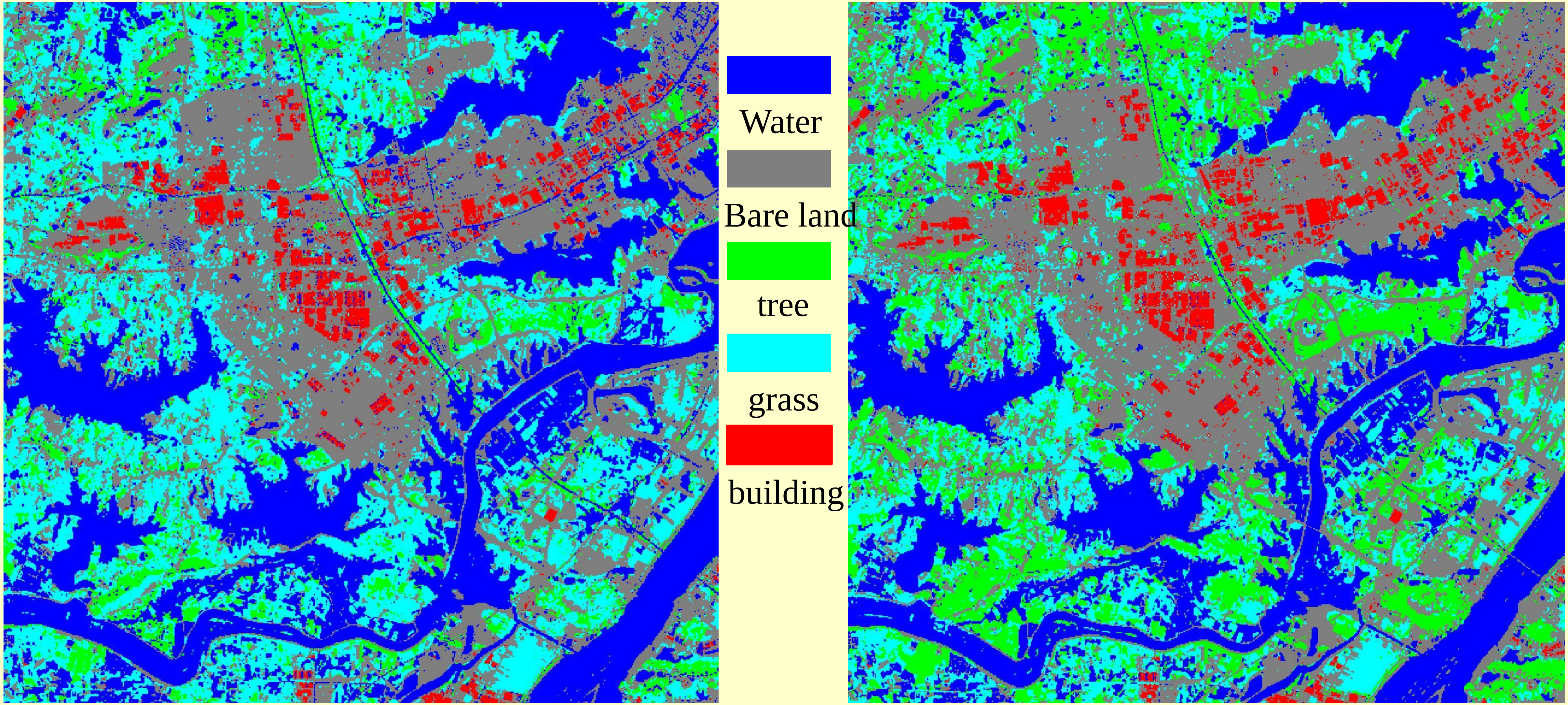
N – sum of elements in the konfusionsmatrix

p_c – the hypothetical probability of chance agreement

$$p_c = \frac{1}{N^2} \sum_i (\sum_j K_{ji} \cdot K_{ij})$$

Original ELM

A-optimal design regularized ELM



Conclusion

1. With A-optimal design regularization, the robustness of ELM is obviously improved.

2. Overfitting model can be effectively avoided in training process, so that generalization performance can be advanced.

3. In image classification, A-optimal design regularization helps original ELM to improve the accuracy of classification.

Outlook

1. Apply the A-optimal design regularization in multi-hidden-layer neural networks.

2. Try to solve other regularization problems in machine learning, e.g. for Support Vector Machine (SVM).

3. Study the prospect of A-optimal design regularization in deep learning, e.g. for convolutional neural networks (CNNs).

Thank you

Contact: Qian Kun
E-mail: qian9361kun@gmail.com
Institute of Geodesy
University of Stuttgart