Report

CAWa classification and valdiation scheme

Markus Möller

Halle (Saale), March 28, 2019

Julius Maximilians University Würzburg, Department of Remote Sensing, Oswald-Külpe-Weg 86, Würzburg 97074, Germany

1 Background, basic approach and test site

A multi-annual classification of aggregated crop types in Central Asia between 2000 and today based on MODIS imagery faces the following challenges:

- Samples and validation data are only available for some years.
- The full extent of target classes is not known.
- Official statistics about crops' proportions are not trustworthy.
- The coarse geometric resolution of MODIS data is related to mixed pixel information and can lead to classification inaccuracies.

Consequently, a crucial bottleneck for a aggregated crop type classification is the lack of mapped training and validation information in most years and regions. Thus, the classification approach is not based on mapped samples. Instead, we assume that aggregated crop types can be characterized so called "pure samples". They represent expert knowledge formalized as $NDVI$ temporal profiles, which are considered as typical for aggregated crop types. The actual classification procedure can be distinguished into three principle steps (Fig. 1):

1. MODIS pixel polygons are considered as reference units. They are coupled with raster-based MODIS $NDVI$ time series for specific years and regions as well with a preclassified irrigation mask by zonal statistics operations (ZS).
2. Every single resulting pixel-specific MODIS $NDVI$ profile is compared with pure sample profiles by a dissimilarity test. The resulting dissimilarity matrix is used for the derivation of crop type-specific samples (SL).
3. The training samples go into a data mining procedure. A statistical model is set up between samples and MODIS $NDVI$ profiles. The model is applied on the total MODIS $NDVI$ data set predicting aggregated crop types.

There are three options to validate the classification results with independent information. These algorithms are not part of the classification process chain and are related to the data types *points*, *polygons* and *raster data*.

In the following sections, all relevant functions for classification (Sec. 2.1) and validation (Sec. 2.2) are described in detail and explained using the example of the Fergana test site (fig. 2), for which a classification of 2015 is carried out.

The functions are implemented within the programming environment of **R** (version 3.5.1; R Core Team, 2017). All **R** srcipts are documented on two GitHub repositories:

- `CAWaClass`[1] collects scripts for MODIS classification.
- `CAWaVal`[2] offers two options to validate a `CAWaClass` classification result.

[1] `https://github.com/terrasys/CAWaClass.git`
[2] `https://github.com/terrasys/CAWaVal.git`

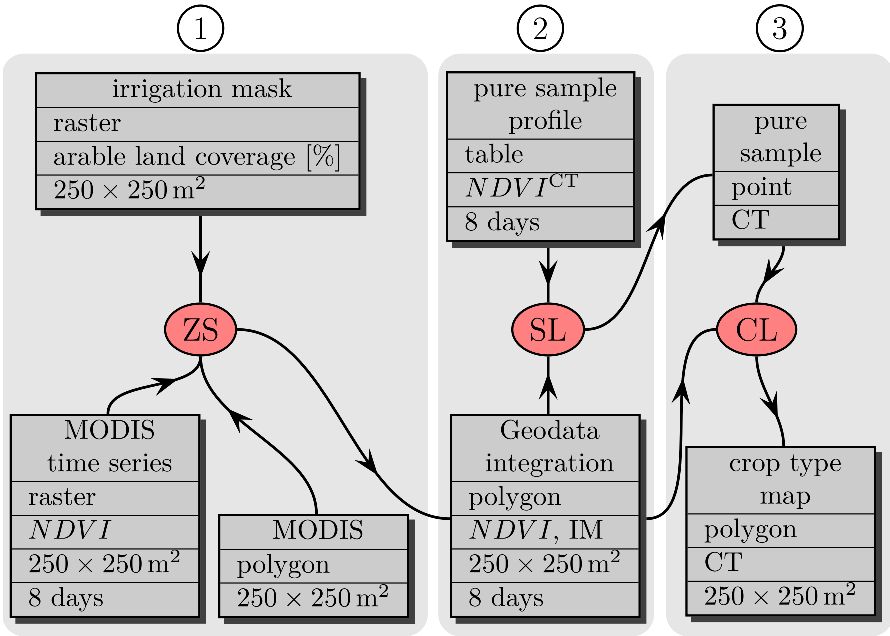


Figure 1: Principle workflow for the sample derivation and classification. OL – overlay | SL – sampling | ZS – zonal statistics | CT – (aggregated) crop type | CL – classification.

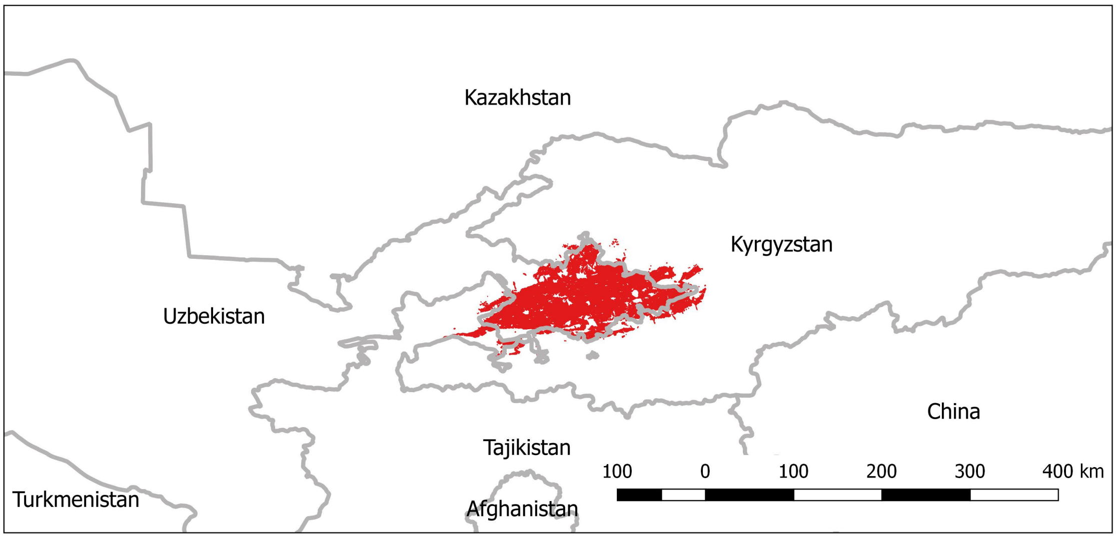


Figure 2: Location of the Fergana test site.

Table 1: Global parameters to be set in `callFunctions.R`.

Parameter	Meaning
W.DIR	working directory
FUNC.DIR	directory containing functions
IN.DIR	directory containing input data
OUT.DIR	directory containing results
MODIS.SHP	name of polygon shapefile containing reference units
IM.GRD	name of the irrigation mask raster
PS	name of the text file containing pure sample *NDVI* profiles
YEAR	year of classification

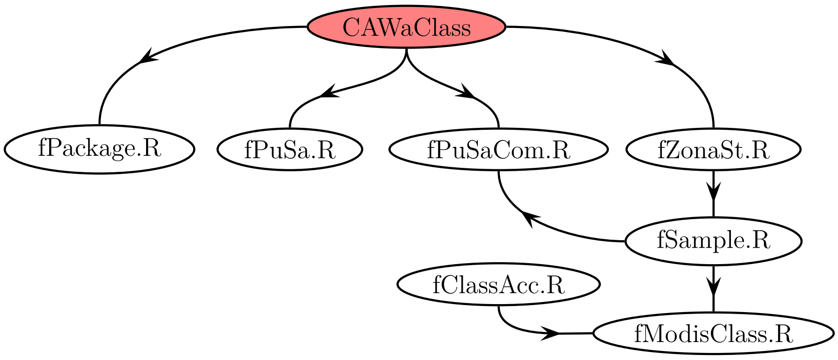


Figure 3: Relations between `CAWaClass` functions.

2 Functions

2.1 Classification

All functions, which are related to classifications, are imported, parameterized and carried out by a wrapper file `callFunctions.R`. There, global settings have to be made concerning working directories or sub-folders storing input and output data (tab. 1). Figure 3 illustrates the relations between the seven functions, which are explained in detail in the following paragraphs.

2.1.1 fPackage

`fPackage` contains the function `loadandinstall()` for an automatic download and installation of all required packages. In addition, the function `rsaga.env()` sets up a RSAGA geoprocessing environment (Brenning, 2008) referring to a pre-installed SAGA-GIS version[3] (Conrad et al., 2015).

[3] `https://sourceforge.net/projects/saga-gis/files/SAGA%20-%202/SAGA%202.2.2`

Table 2: `fPuSa`: parameters and results.

Parameters/results	**Meaning**
W.DIR	working directory
IN.DIR	directory storing input data
OUT.DIR	directory storing results
PS	`csv`-file with class-specific $NDVI$ profiles
CLASS.NAME	name of column with class names
$[W.DIR]/[OUT.DIR]/[PS]$_DM.csv	dissimilarity matrix (tab. 3)
$[W.DIR]/[OUT.DIR]/[PS]$_NDVI-profiles.pdf	$NDVI$ profile plot (fig. 4)

2.1.2 fPuSA

The function `fPuSA()` aims at the comparison of crop type-specific $NDVI$ time series (tab. 2). The comparison is based on two functions of the `TSclust` package, which contains measures of dissimilarity between time series (Montero & Vilar, 2014):

- The function `diss.COR()` "computes dissimilarities based on the estimated Pearson's correlation of two given time series" (d^{COR}).
- The function `diss.CORT()` "computes an adaptive dissimilarity index between two time series that covers both dissimilarity on raw values and dissimilarity on temporal correlation behaviors" (d^{COR}).

Both metrics are combined by creating a scaled product D resulting in a value range between 0 and 1 (Eq. (1) and (2)). Low D values stand for a high degree of similarity, and high D values for a high degree of dissimilarity.

$$d = d^{COR} \times d^{CORT} \tag{1}$$

$$D = \frac{d - d_{min}}{d_{max} - d_{min}} \tag{2}$$

For the study region, pure samples have been used, which were generated by Conrad et al. (2011). Figure 4 displays $NDVI$ profiles of five aggregated crop types. Table 3 shows the resulting dissimilarity metrics. For instance, the $NDVI$ profiles of the classes "summer crop" and "perennial crop" are characterized by a higher similarity ($D = 0,15$) than "summer crop" and "winter crop" ($D = 0,67$).

2.1.3 fZonaSt

The function `fZonaSt()` enables a coupling of arbitrary reference units with both MODIS $NDVI$ raster files for any year or region ([YEAR]-[MONTH]-[DAY]_ndvi.tif) and a pre-

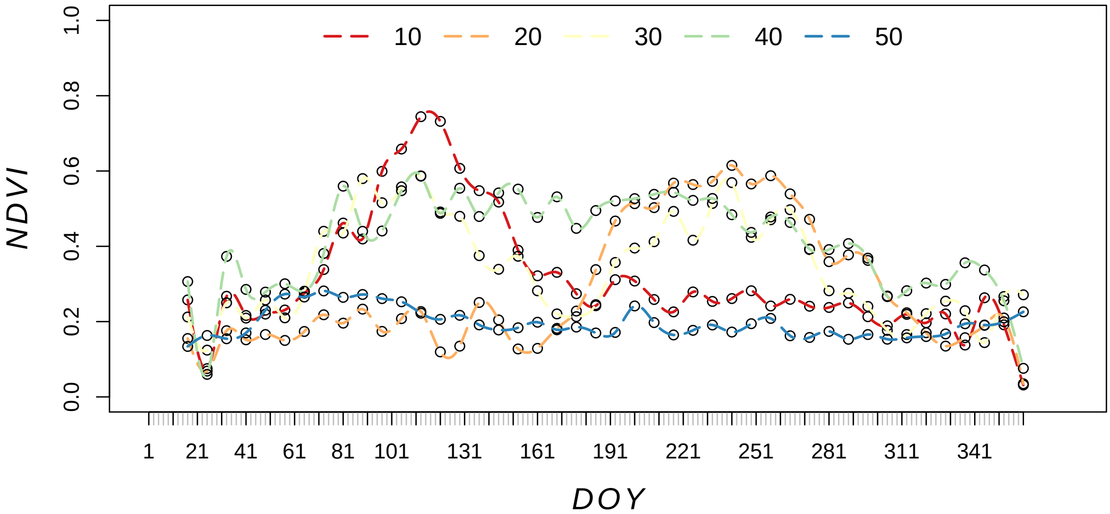


Figure 4: `fPuSa` output: $NDVI$ profiles of aggregated crop types derived from samples and MODIS imagery according to Conrad et al. (2011). 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial Crop | 50 – bare land.

Table 3: `fPuSa.R` output: matrix of class-specific dissimilarity values ($D^{[10,20,30,40,50]}$). 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

	D^{10}	D^{20}	D^{30}	D^{40}	D^{50}
D^{10}	0,00	0,67	0,25	0,15	0,66
D^{20}	0,67	0,00	0,26	0,31	0,89
D^{30}	0,25	0,26	0,00	0,17	0,53
D^{40}	0,15	0,31	0,17	0,00	1,00
D^{50}	0,66	0,89	0,53	1,00	0,00

Table 4: `fZonaSt`: parameters and results.

Parameters/results	**Meaning**
W.DIR	working directory
IN.DIR	directory storing input data
OUT.DIR	directory storing results
MODIS.SHP	name of reference unit shapefile [*.shp]
IM.GRD	name of irrigation mask raster file [*.sgrd]
V.IM	threshold for selecting the irrigation area
YEAR	year to be classified
$[W.DIR]/[OUT.DIR]/$ $[MODIS.SHP, YEAR]$.shp	shape file with a irrigation mask value ($IM \in [0,100]$) as well as $NDVI$ values for specific years and DOYs ($MD[DOY] \in [0,1]$)

defined irrigation mask raster file (tab. 4). The actual coupling procedure is realized by using the RSAGA function `rsaga.geoprocessor()`, which allows the execution of SAGA-GIS modules. The zonal statistics algorithm can be found in the library `shapes_grid`, where module 2 is related to the corresponding zonal statistics function. As a result, the reference units shape file is parametrized by a irrigation mask value ($IM \in [0,128]$) and DOY-specific $NDVI$ values ($MD[DOY] \in [0,1]$). The extent of the irrigation mask includes the total area in Central Asia, which should be classified. Here, an irrigation mask threshold of $V.IM = 100$ is used for the selection of irrigation areas in the Fergana test site.

2.1.4 fSample

The function `fSample()` aims at the localization of samples for specific years and sites (tab. 5). The procedure is based on the dissimilarity test introduced in section 2.1.2 and can be structured into two steps:

1. Comparison of pure sample $NDVI$ profiles (parameter PS) with pixel-sepecific MODIS $NDVI$ profiles (parameter ZS.SHP) by applying the functions `diss.COR` and `diss.CORT` and calculation of class- and pixel-specific D values (Sec. 2.1.2),

2. Calculation of class-specific quantiles of D value distributions and selection of pixels, which fulfill a user-specific quantile-based thresholds (parameter Q) and removing duplicates.

In table 6, a subset of class- and pixel-specific dissimilarity values ($D^{[10,20,30,40,50]}$) is shown. The corresponding class-specfic density functions are displayed in figure 5. The red dashed vertical lines mark the position of user-specific quantiles, which act as dynamic thresholds for sample selection. The selection result can be controlled using

Table 5: `fSample`: parameters and results.

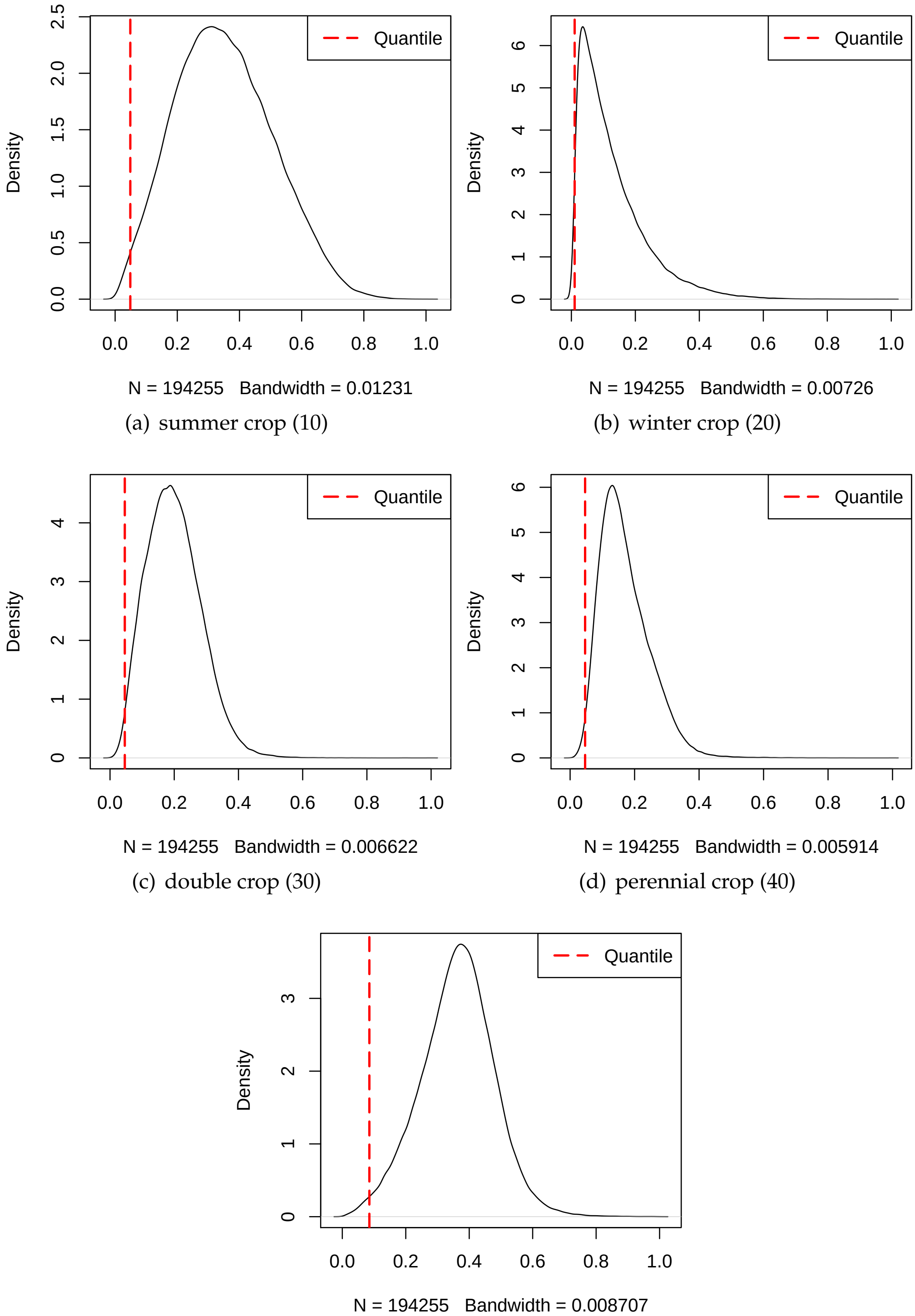
Parameters/results	Meaning
W.DIR	working directory
IN.DIR	directory storing input data
OUT.DIR	directory storing results
PS	`csv`-file with class-specific $NDVI$ profiles
MODIS.SHP	name of reference unit shapefile [*.shp]
ZS.SHP	shape file with $NDVI$ values for specific DOYs and YEARs; result from `fZonaSt()` function
Q	quantile of D value distribution (Sec. 2.1.2)
$[W.DIR]/[OUT.DIR]/$ $[MODIS.SHP, YEAR]$_SAMPLE-NDVI-densityplot_Q[Q].pdf	density plot of class- and pixel-specific dissimiliarity values (tab. 6 and fig. 5)
$[W.DIR]/[OUT.DIR]/$ $[MODIS.SHP, YEAR]$_SAMPLE-NDVI-barplot_Q[Q].pdf	barplot of class-specific samples
$[W.DIR]/[OUT.DIR]/$ $[MODIS.SHP, YEAR]$_DM.csv	class- and pixel-specific dissimiliarity matrix (tab. 6)
$[W.DIR]/[OUT.DIR]/$ $[MODIS.SHP, YEAR]$.shp	sample shape file with an irrigation mask value ($IM \in [0, 100]$) as well as $NDVI$ values for specific years and DOYs ($MD[DOY] \in [0, 1]$)
$[MODIS.SHP, YEAR]$_SAMPLE-NDVI_agg_Q[Q].csv	aggregated samples of class-specific $NDVI$ values

Table 6: `fSample.R` output: subset of class- and pixel-specific dissimilarity values ($D^{[10,20,30,40,50]}$). ID – MODIS pixel ID | 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

ID	D^{10}	D^{20}	D^{30}	D^{40}	D^{50}
316	0,28	0,52	0,38	0,13	0,43
317	0,24	0,50	0,35	0,14	0,38
318	0,22	0,60	0,40	0,17	0,31
319	0,25	0,54	0,36	0,19	0,43
1374	0,22	0,67	0,44	0,17	0,44
1375	0,21	0,63	0,38	0,17	0,38
1376	0,22	0,66	0,41	0,17	0,39
2425	0,27	0,42	0,27	0,13	0,53
2426	0,26	0,56	0,33	0,15	0,42
2427	0,20	0,60	0,43	0,15	0,41
...	...	...	...	...	...

figure 6, where the sample number and the class-specific sample proportions are displayed. Accordingly, for 2015 a total sample number of 9587 could be selected. The sample class proportions are nearly equal, which is related to the quantile-based selection approach.

Figure 6: `fSample` output: barplot of class-specific samples. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial Crop | 50 – bare land.



(a) summer crop (10)

(b) winter crop (20)

(c) double crop (30)

(d) perennial crop (40)

(e) bare land (50)

Figure 5: `fSample` output: Density plots of class- and pixel-specific dissimiliarity values (see tab. 6).

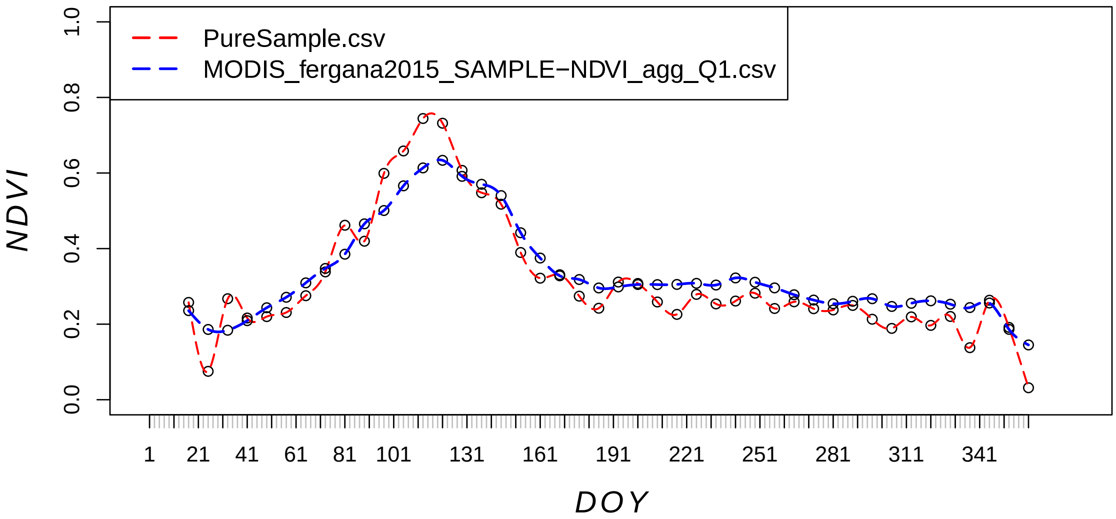


Figure 7: `fPuSaCom` output: plot of class-specific $NDVI$ profiles based on both pure samples and aggregated samples derived with `fSample()` function on the example of land use class 10 (summer crops).

Table 7: `fPuSaCom`: parameters and results.

Parameters/results	**Meaning**
W.DIR	working directory
IN.DIR	directory storing input data
OUT.DIR	directory storing results
CLASS.NAME	name of column with class names
PS1	`csv`-file with class-specific $NDVI$ profiles
PS2	`csv`-file with aggregated class-specific $NDVI$ profiles derived from applying funtion `fSample()`
PS1PF	name prefix in column name of PS1
PS2PF	name prefix in column name of PS2
$[W.DIR]/[OUT.DIR]/\ [PS1]\_[PS2]$.pdf	plot of class-specific $NDVI$ profiles based on both pure samples and aggregated samples derived with `fSample()` function

2.1.5 fPuSaCom

Function `fPuSaCom()` enables the comparison of pure samples (Sec. 2.1.2) and aggregated samples derived with function `fSample()` (Sec. 2.1.4). As a result, comparing plots of class-specific $NDVI$ profiles are generated. As shown in figure 7 on the example of class "summer crops", the profiles can be visually assessed regarding their similarity.

2.1.6 fModisClass & fClassAcc

The test site- and year-specific classification is realized by executing the function `fModis-Class()` (tab. 8), where some functions of the **R** package `caret` are combined (Kuhn & Johnson, 2013; Kuhn et al., 2018). The classification process starts with a data partition procedure (here: 75 % training data and 25 % test data) considering the class proportions (function `caret::createDataPartition`), which is applied to the sample shape file derived from the function `fSample()` (Sec. 2.1.4). The option `caret::upSample` enables to adapt the sample number of the minority class to the same size as the majority class.

The function `caret::train` is used for the actual model building. There, the classifier can be set by user (option M.TRAIN)[4]. The classification of the Fergana test site is based on "Random Forest" (RF) algorithm, which has been proven as robust classifier for remore sensing applications (Belgiu & Drăguţ, 2016). RF is representative of data mining algorithms and stands for a regression- and ensemble-based decision tree algorithm. RF splits the feature space of the explanatory variables until the resulting tree shows the best statistical correlation by minimizing the variance. Based on bootstrapped samples, RF generates a large number of independent trees (ensembles). Two thirds of the samples are used for growing trees (*in-bag* data), and one third are randomly drawn with a replacement for the calculation of error estimates by cross-validation (*out-of-bag* data; Breiman, 2001).

Figure 8 illustrates the shape file of classified crop types and their proportional coverage of areas. The shape file also contains information about the pixel-specific class probability, which expresses the proportion of the trees that voted for each class. Apart from that, following information about the classification accuracy are provided:

- The overall accuracy is based on a repeated 10-fold cross validation of the training data set and can be considered as the internal model accuracy.
- The fitted model is used for the classification of the test data set, for which a confusion matrix and accuracy metrics are calculated (external validation) by activating function `fClassAcc()`[5]. The *F1* score represents the weighted harmonic mean of the metrics "precision" and "recall" (Eq. (3)). Both metrics result from the quotient of the "number of correctly classified instances per class" with "the number of predictions per class" (Eq. (4); precision) and the number of instances per class (Eq. (5); recall), respectively.

The internal accuracy assessment for the MODIS-based classification of the Fergana test site are listed in table 9 and 10.

[4] `https://topepo.github.io/caret/train-models-by-tag.html`
[5] `https://github.com/saidbleik/Evaluation/blob/master/eval.R`

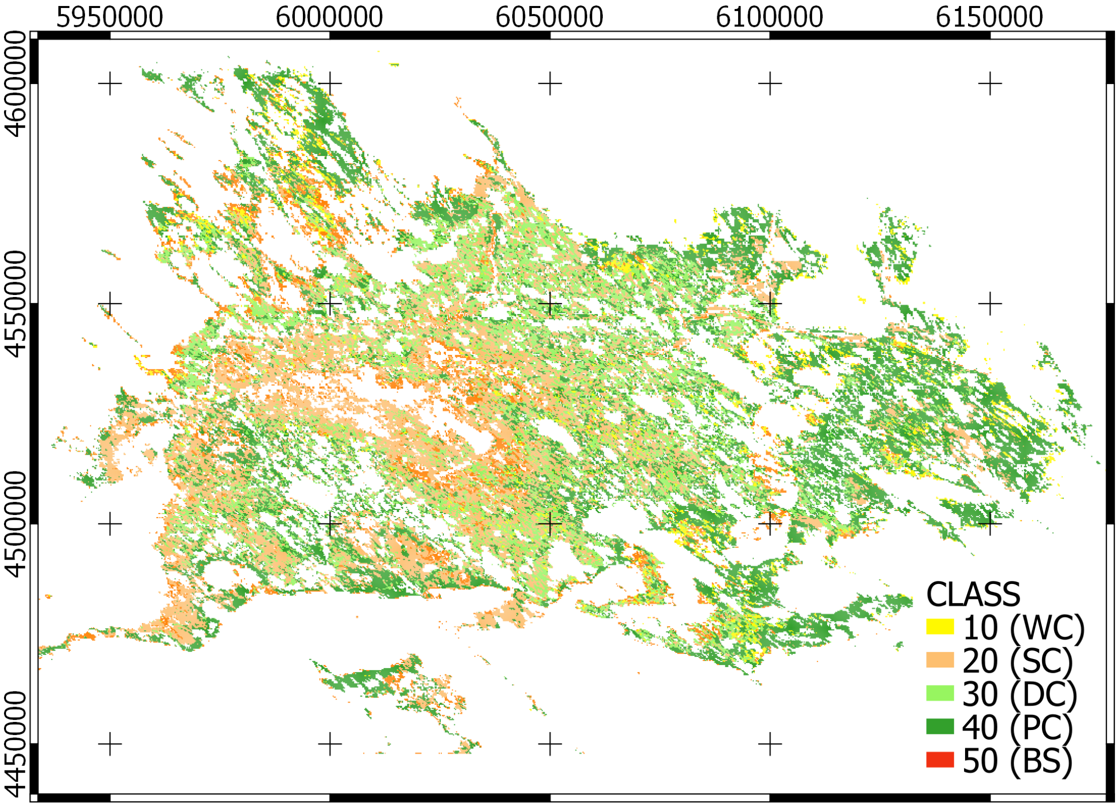
$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Table 8: `fModisClass`: parameters and results.

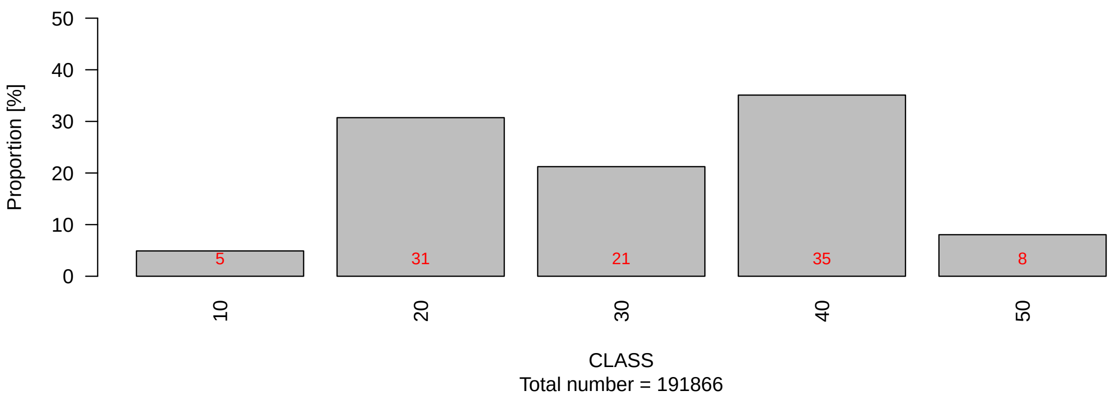
Parameters/results	Meaning
W.DIR	working directory
IN.DIR	directory storing input data
OUT.DIR	directory storing results
ZS.SHP	shape file with $NDVI$ values for specific DOYs and YEARs; result from `fZonaSt()` function
SAMPLE.SHP	sample shape file resulting from applying function fSample()
PART	proportion of SAMPLE.SHP which should be used for training [$\in 0,1$] and validation
T.CLASS	name of column within SAMPLE.SHP file with target class names
M.TRAIN	classification method
UpSample=TRUE	randomly sample (with replacement) the minority class to be the same size as the majority class
$[W.DIR]/[OUT.DIR]/$ $[SAMPLE.SHP]$_$[M.TRAIN]$_CV.txt	cross validation result
$[W.DIR]/[OUT.DIR]/$ $[SAMPLE.SHP]$_$[M.TRAIN]$_AM.csv	accuracy metrics based on test data set derived from data partition of SAMPLE.SHP
$[W.DIR]/[OUT.DIR]/$ $[SAMPLE.SHP]$_$[M.TRAIN]$_CM.csv	confusion matrix based on test data set derived from data partition of SAMPLE.SHP
$[W.DIR]/[OUT.DIR]/$ $[ZS.SHP]$_$[M.TRAIN]$_CLASS.shp	shape file with classification result and corresponding class probobility (columns [CLASS] and [CLASS]_PB)
$[W.DIR]/[OUT.DIR]/$ $[ZS.SHP]$_$[M.TRAIN]$_CLASS.pdf	barplot of classified crop types

$$\text{Precision} = \frac{\text{Diag}}{\text{colsums}} \tag{4}$$

$$\text{Recall} = \frac{\text{Diag}}{\text{rowsums}} \tag{5}$$



(a)



(b)

Figure 8: `fModisClass` output: Classification result and barplot of classified crop types on the example of Fergana test site and 2015. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

Table 9: `fModisClass` output: Confusion matrix based on test data set on the example of Fergana test site and 2015. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

	10	**20**	**30**	**40**	**50**
10	447	0	2	6	9
20	0	507	0	0	0
30	5	0	463	8	1
40	1	0	8	464	1
50	3	0	1	0	470

Table 10: `fModisClass` output: Accuracy metrics based on test data set on the example of Fergana test site and 2015. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

	10	**20**	**30**	**40**	**50**
Accuracy	0,98	0,98	0,98	0,98	0,98
Precision	0,98	1,00	0,98	0,97	0,98
Recall	0,96	1,00	0,97	0,98	0,99
F1	0,97	1,00	0,97	0,97	0,98
Kappa	0,98	0,98	0,98	0,98	0,98

2.2 Validation

Apart from the internal `CAWaClass` validation (Sec. 2.1.6), MODIS classification results can also be validated externally by independent information. The script collection `CAWaVal` (fig. 9) includes two options:

1. valdiation based on (point-related) sample shape file (Sec. 2.2.1) and
2. validation based on a raster-based classification result (Sec. 2.2.2).

In both cases, the actual validation procedure is based on the same function `fClassAcc()`, which was already introduced in section 2.1.6. This is also true for the script `fPackage` (Sec: 2.1.1).

2.2.1 fClassCompareP

The function `fClassCompareP()` combines the accuracy assessment procedure with a geometric overlay operation, where a point-related data set containing validation information is coupled with a classification result.

Figure 10 displays the barplot of samples mappped 2015 in Fergana test site. The class-specific sample numbers are quite unbalanced. This concerns especially the classes 40 (perennial crops) and 50 (bare soils), for which only few samples are available. The confusion matrix reveals, that no bare soil samples were mapped within the irrigation

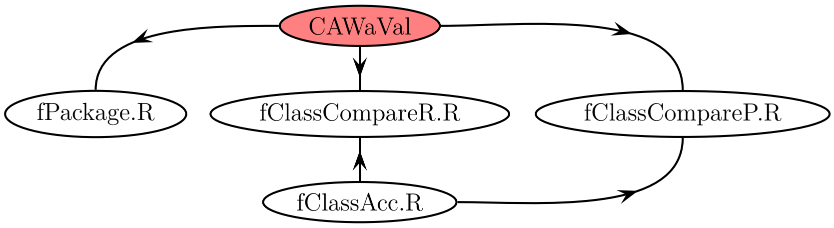


Figure 9: Relations between `CAWaVal` functions.

Table 11: `fClassCompareP`: parameters and results.

Parameters/results	Meaning
W.DIR	working directory
IN.DIR	directory storing input data
OUT.DIR	directory storing results
CLASS.SHP	MODIS classification result (Sec. 2.1.6)
POINT.SHP	Point-related validation data set
RASTER.FORMAT	Raster format of CLASS.RASTER
[*W.DIR*]/[*OUT.DIR*]/ [*CLASS.SHP*]__[*POINT.SHP*]_AM.csv	accuracy metrics based on an overlay of CLASS.SHP and POINT.SHP
[*W.DIR*]/[*OUT.DIR*]/ [*CLASS.SHP*]__[*POINT.SHP*]_CM.csv	confusion matrix based on an overlay of CLASS.SHP and POINT.SHP
[*W.DIR*]/[*OUT.DIR*]/ [*POINT.SHP*]_BARPLOT.csv	Class-specific barplot of samples mapped 2015 in Fergana test site.

Table 12: `fClassCompareR`: parameters and results.

Parameters/results	Meaning
W.DIR	working directory
IN.DIR	directory storing input data
OUT.DIR	directory storing results
CLASS.SHP	MODIS classification result (Sec. 2.1.6)
CLASS.RASTER	Raster-based classification
RASTER.FORMAT	Raster format of CLASS.RASTER
[*W.DIR*]/[*OUT.DIR*]/ [*CLASS.SHP*]__[*CLASS.RASTER*]_AM.csv	accuracy metrics based on overlay of CLASS.SHP and CLASS.RASTER
[*W.DIR*]/[*OUT.DIR*]/ [*CLASS.SHP*]__[*CLASS.RASTER*]_CM.csv	confusion matrix based on an overlay of CLASS.SHP and CLASS.RASTER

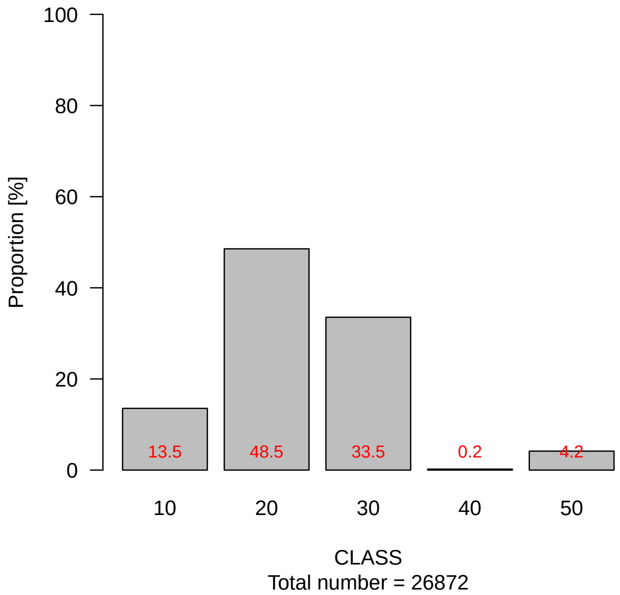


Figure 10: Class-specific barplot of samples mapped 2015 in Fergana test site.

Table 13: `fClassCompareP` output: Confusion matrix based on an overlay of a point-related data set containing validation information and classification result for Fergana test site in 2015. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

	10	**20**	**30**	**40**	**50**
10	1739	38	931	227	431
20	0	6341	2371	1526	76
30	1082	263	5021	1378	0
40	0	0	15	42	0
50	0	0	0	0	2

Table 14: `fClassCompareP` output: Accuracy metrics based on an overlay of a point-related data set containing validation information and classification result for Fergana test site in 2015. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

	10	**20**	**30**	**40**	**50**
Accuracy	0,61	0,61	0,61	0,61	0,61
Precision	0,62	0,95	0,60	0,01	0,00
Recall	0,52	0,61	0,65	0,74	1,00
F1	0,56	0,75	0,62	0,03	0,01
Kappa	0,44	0,44	0,44	0,44	0,44

Table 15: `fClassCompareR` output: Confusion matrix based on an overlay of a raster-based data set containing validation information and a classification result for the Fergana test site in 2015. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

	10	**20**	**30**	**40**	**50**
10	533	0	6	2	169
20	0	8210	37	106	243
30	16	0	802	3	0
40	13	38	40	1982	225
50	0	0	0	0	3

Table 16: `fClassCompareR` output: Accuracy metrics based on an overlay of a raster-based data set containing validation information and a classification result for the Fergana test site in 2015. 10 – summer crop | 20 – winter crop | 30 – double crop | 40 – perennial crop | 50 – bare land.

	10	**20**	**30**	**40**	**50**
Accuracy	0,68	0,68	0,68	0,68	0,68
Precision	0,48	0,98	0,87	0,35	0,00
Recall	0,52	0,61	0,65	0,74	1,00
F1	0,59	0,97	0,92	0,50	0,01
Kappa	0,54	0,54	0,54	0,54	0,54

area during the field campaign (tab. 13). These boundary conditions also affect the accuracy metrics summarized in table 14, which show especially for the classes 40 and 50 poor results.

2.2.2 fClassCompareR

The function `fClassCompareR()` enables a validation, where a pre-classified raster data set is coupled with the MODIS classification result. The data coupling is realized by zonal statistics operations (sec. 2.1.3). In doing so, the maximum class value within each MODIS polygon is detected. Since the reference classification can be characterized by a higher geometric resolution, the range of class values is also derived. The actual accuracy assessment only considers MODIS polygones where the value range is zero.

For the validation of the MODIS classification from 2015, a Landsat classification from the same year was available. Tables 15 and 16 provide the confusion matrix and accuracy metrics.

References

Belgiu, M. & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31.

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

Brenning, A. (2008). Statistical geocomputing combining r and saga: The example of landslide susceptibility analysis with generalized additive models. In *SAGA – Seconds Out (= Hamburger Beitraege zur Physischen Geographie und Landschaftsoekologie, vol. 19)* (pp. 23–32). J. Boehner, T. Blaschke, L. Montanarella.

Conrad, C., Colditz, R., Dech, S., Klein, D., & Vlek, P. (2011). Temporal segmentation of MODIS time series for improving crop classification in Central Asian irrigation systems. *International Journal of Remote Sensing*, 32(23), 8763–8778.

Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V., & Böhner, J. (2015). System for automated geoscientific analyses (saga) v. 2.1.4. *Geoscientific Model Development*, 8(7), 1991–2007.

Kuhn, M. & Johnson, K. (2013). *Applied Predictive Modeling*. New York, Heidelberg, Dordrecht, London: Springer.

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., the R Core Team, Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., Candan, C., & Hunt., T. (2018). *caret: Classification and Regression Training*. R package version 6.0-81.

Montero, P. & Vilar, J. A. (2014). TSclust: An R package for time series clustering. *Journal of Statistical Software*, 62(1), 1–43.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.