Detecting changes in essential ecosystem and biodiversity properties- towards a Biosphere Atmosphere Change Index: BACI

Deliverable 4.1: Newly developed machine learning techniques for upscaling EEVs and EFPs

Project title:	Detecting changes in essential ecosystem and biodiversity properties- towards a Biosphere Atmosphere Change Index
Project Acronym	BACI
Grant Agreement Number:	640176
Main pillar:	Industrial Leadership
Topic:	EO-1-2014: New ideas for Earth-relevant space applications
Start date of the project:	1st April 2015
Duration of the project:	48 months
Dissemination level:	Public
Responsible of the deliverable:	Joachim Denzler Phone: +49 3641 9 46420 Email: joachim.denzler@uni-jena.de
Contributors:	Erik Rodner, Sven Sickert, Yanira Guanche, Martin Jung, Paul Bodesheim
Date of submission:	March 29, 2016

Contents

Summary

In this deliverable Machine Learning methods are proposed to upscale Essential Ecosystem Variables and Ecosystem Functional Properties. Random Decision Forests are proposed to upscale in-situ measured half-hourly carbon and energy fluxes, which is a large-scale regression problem. Random Decision Forests fulfill the necessary requirements in terms of robustness and computational speed for both training and forward prediction. A combination of Random Decision Forests and Gaussian Process is proposed to upscale Ecosystem Functional Properties. This approach is tailored to the problem of accounting for, and producing additionally uncertainty estimates of the prediction, which is a key requirement of this task. Various tests with dedicated kernel functions to handle variables with different nature of variability (spatial vs spatio-temporal) database were performed using the FLUXNET to verify the effectiveness of the algorithm.

1 Introduction

Understanding key processes of biosphere-atmosphere interactions requires knowledge on diurnal variations of biosphere-atmosphere fluxes [6] on the one hand, and information on spatial variations of key ecosystem functional properties on the other hand. Previous efforts combined the global database FLUXNET with Earth Observations using Machine Learning approaches but those produced only fluxes at monthly resolution [4],[5],[1]. This deliverable identifies and presents suitable Machine Learning methods for the following two tasks in WP4 of BACI:

- Task 4.1 Land-atmosphere carbon and energy fluxes with sub-daily resolution: This task of upscaling EEVs is a large scale regression problems. Very large training data sets (several million observations) need to be considered and the forward prediction task requires computations for roughly 10^9 data points.
- Task 4.2 Spatially explicit ecosystem functional properties: Mapping Ecosystem Functional Properties at a global scale requires techniques that are able to measure its uncertainties, are capable of handling missing data, and are able to produce multivariate outputs that preserve the observed existing co-variation of the target variables.

2 Machine Learning Methods

After a thoroughly examination of the two problems detailed before we came to the following conclusions:

- Random Decision Forest (RDF) are suitable for the large scale regression problem (working with EEVs). These models are capable of handling large data sets in the training as well as in the test phase. The forward prediction for many points ($\sim 10^{11}$ data points to be predicted) is due to the tree composition computationally feasible. The method is very well established and has been used widely in different scientific communities. Another advantage is the robustness with respect to overfitting due to different types of randomness utilized during training.
- A new machine learning method was designed to solve the problem of regression where the uncertainty of the prediction is estimated and can be also integrated as prior knowledge for input variables (in the case of working with EFP). This new method combines Random Decision Forest (RDF) with Gaussian Process Regression (GP).

In the following, we will give a brief overview of the RDF-GP method used in our experiments. As mentioned before, the method proposed is a combination of *random decision forests* (RDF) and *Gaussian process* (GP) regression in the leaf nodes of the decision trees [7]. For a more detailed explanation for each of these machine learning techniques the reader is referred to [3] and [9], respectively.

2.1 Random Decision Forests

An RDF is an ensemble method consisting of several decision trees that can be applied to both classification and regression tasks. It is able to handle large sets of training examples $(\mathbf{x}, y)_i,\ 1 \leq\ i \leq\ N$ and is using linear base classifiers (decision stumps) to iteratively cluster the data in the attribute space. Beginning from a root node, simple comparisons of attribute values of one dimension $d,\ 1 \leq\ d \leq\ D$ with a threshold θ decide whether a data example is handed over to the left ($x_d < \theta$) or the right ($x_d \geq \theta$) child node of a currently processed node.

The trees of an RDF are trained with specific randomization techniques in order to avoid over-fitting to the training data. Each tree is learned with a random subset of the complete training set. The data is recursively split by axis orthogonal hyperplanes which are optimized with respect to a certain purity criterion (*e.g.* mean square error) until a stopping criterion is reached. In each node, both attributes and thresholds for the splitting are drawn randomly. The procedure is stopped if the current set contains too few examples or a certain depth level of the tree is reached.

In order to get a target value y_* for a given test example $\mathbf{x}_*$ the example traverses each of the learned trees of the RDF. It will reach one of the child nodes in each tree. Each child node contains a set of examples which reached that node during training. The average of their associated target values is the prediction value for that particular leaf node. Accordingly, the prediction y_* for a certain test example $\mathbf{x}_*$ is the average of predictions from all leaf nodes it landed in.

2.2 Gaussian Processes

The mapping from input data $\mathbf{x}$ to an target value y is often modeled by $y = f(\mathbf{x}) + \epsilon$, where f is a noise-free latent function and ϵ a noise term. It is common to assume that f belongs to a parametric family and the parameters which best describe the data have to be learned. However, using GP the underlying function f can be modeled directly without any fixed parametrization by assuming the function to be sampled from a specific distribution. Such a distribution on functions can be defined in a non-parametric manner by GPs.

For the task of regression, we assume the latent function f to be a sample from a GP prior $f \sim GP(\mathbf{0}, K(\cdot,\cdot))$ with zero mean and a kernel function $K\mathbb{R}^D \times \mathbb{R}^D \longrightarrow \mathbb{R}$. The target values y are conditionally independent given the latent function values $f(\mathbf{x})$ and are described using a noise model $p(y|f(\mathbf{x}))$. A standard assumption for GP regression is to model noise as a zero-mean Gaussian noise with variance σ_N^2: $p(y|f(\mathbf{x})) = \mathcal{N}(y|f(\mathbf{x}), \sigma_N^2)$. This allows for tractable predictions for unseen points $\mathbf{x}_*$. However, also heteroscedastic noise, *i.e.* non-homogeneous uncertainties of examples, can be modeled by be specifying different noise variances for each of the training examples

Let $\mathbf{K}$ be the kernel matrix with pairwise kernel values of the training examples $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. Furthermore, let $\mathbf{k}_*$ be the kernel values $(\mathbf{k}_*)_i = K(\mathbf{x}_i, \mathbf{x}_*)$ corresponding to $\mathbf{x}_*$. The most likely target value y_* for $\mathbf{x}_*$ and the given training data can be predicted by:

$$y_*(\mathbf{x}_*) = \mathbf{k}_*^T(\mathbf{K} + \sigma_N^2\mathbf{I})^{-1}\mathbf{y}, \tag{1}$$

where $\mathbf{y}$ is a vector of the target values corresponding to the training examples $\mathbf{x}$ and $\mathbf{I}$ being the identity matrix.

Uncertainties of specific variables in the examples can be integrated by modifying the kernel function used, *e.g.* by using a Gaussian kernel with a Mahalanobis distance.

2.3 Combining RDFs and GP

GPs are very powerful tools for the task of regression. However, during training the kernel matrix has to be computed and inverted which is cubic in the number of training examples N. As a consequence, it is often intractable to apply GP to large data sets directly.

As a solution, we propose to combine RDFs and GPs. The RDF is trained in the traditional manner using simple binary decision in the inner nodes. In each leaf node, however, we learn a GP with a rather small kernel matrix using only the training examples which reached a certain leaf node. We thereby construct an ensemble of powerful GP predictors in an efficient manner since the complexity for training and testing is reduced [7].

3 Preliminary Experiments

3.1 Data

The experiments are based on in-situ measured carbon and energy fluxes from FLUXNET. FLUXNET is a network of regional networks that coordinates regional and global analysis of observations from micro-meteorological tower sites[1]. An overview on the global network and the tower sites around the world are given in Fig. 1.

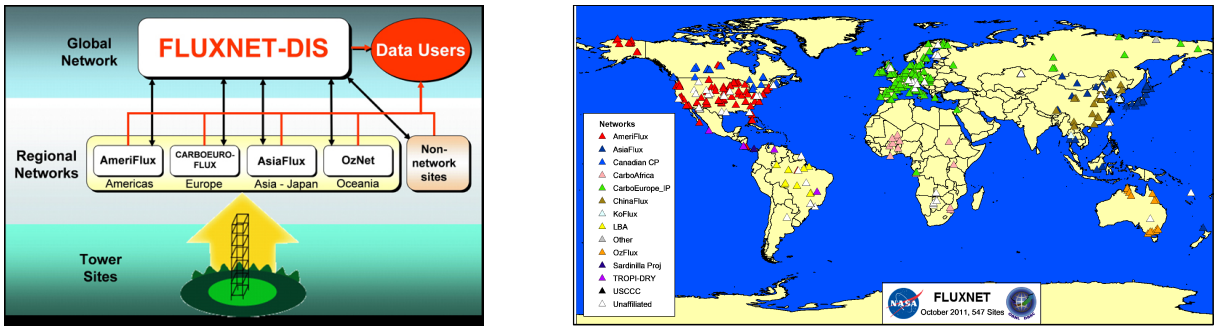


Figure 1: FLUXNET is a network of regional networks (left) of flux towers which are positioned around the world in different climate zones (right).

In these particular tests the so-called *global primary production* (GPP) at 8-daily temporal resolution is the target value. It can only be measured at the tower sites using the eddy covariance methods. However, there is a set of large-area input variables that can be recorded using satellites. A corresponding list of these input variables is given in Fig. 2. With the help of these globally available variables the GPP can be predicted globally as well. Since both GPP and the input variables are available at each tower site regression techniques can be used for prediction. It should be noted that some variables are fixed for a specific tower site, *e.g.* the mean seasonal cycle attributes.

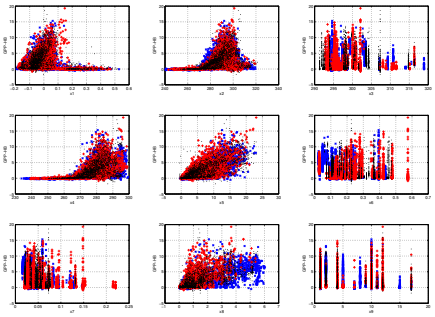
3.2 Experimental Setup

The FLUXNET based training dataset used consists of 11,501 data points with nine dimensions each. The data points are stored as a sorted time-series in a *8-daily* setting. Since all points originate from flux towers the target variable GPP is given for all data samples as well. In order to evaluate the approach experiments using a 10-fold cross-validation were conducted.

For a quantitative evaluation of the method, the so-called Nash-Sutcliffe model efficiency $NSE = 1 - SMSE$ [8] was utilized which makes use of the standardized mean squared error (SMSE). The basis is the mean squared error (MSE) for T test cases:

$$\frac{1}{T}\sum_{i=1}^{T}(y_*^{(i)} - \bar{f}(\mathbf{x}_*^{(i)}))^2 \ , \tag{2}$$

[1] see: *fluxnet.ornl.gov/introduction*



dim	meaning
1	satellite surface water indicator
2	satellite land surface temperature (daytime)
3	max of mean seasonal cycle of land surface temperature (daytime)*
4	satellite land surface temperature (night-time)
5	product of a satellite vegetation indicator and shortwave radiation
6	amplitude of mean seasonal cycle of a vegetation indicator*
7	amplitude of mean seasonal cycle of a middle infrared reflectance*
8	vegetation indicator
9	vegetation type*

Figure 2: Plots of the nine input dimensions of the FLUXNET dataset against the target attribute *GPP* (left) and their corresponding description (right). Attributes marked with an asterisk (*) vary only between flux towers and are therefore fixed for each site.

Name	Splitting	Prediction	minLeaf	SMSE
RDF (standard)	Variance	Mean	50	0.291
RDF (linear)	Linear LSE	Linear model	50	0.385
RDF (knn)	MinDist	k-NN Mean	50	0.281
RDF (exp1)	Linear LSE	k-NN Mean	50	0.386
RDF (exp2)	MinDist	Linear model	50	**0.268**

Table 1: Results using random forests with varying splitting and prediction functions. See text for a more detailed analysis.

where $\bar{f}(\mathbf{x}_*)$ is the regression method's prediction for a single test sample and y_* a corresponding target value. In order to retrieve the SMSE the MSE is again normalized using the variance of the target variable σ_{GPP}. As a consequence a simple mean guessing using all training target values would lead in $SMSE \approx 1$. Smaller SMSE values indicate better prediction performances. While the optimal value would be 0 the best reachable model efficiency NSE would be 1, accordingly.

3.3 Results using Random Forests

For the first series of experiments we used standard regression trees in a random forest setting. In the splitting function the sum of variances in both of the child subsets with respect to the target variable is minimized. Therefore, a certain amount of random splits on different input dimensions it tested and evaluated using the aforementioned criterion. Consequently, prediction in a leaf node can be done by returning the mean of target values from training examples that landed in that leaf. In a configuration using ten trees we were able to reach a SMSE of 0.29.

We continued with varying the splitting and prediction functions. First of all, we allowed a more general linear model. In contrast to ordinal splits linear least-squares estimation (LSE) was applied in inner nodes and a linear model was fit in the leaves. The SMSE dropped drastically which could be explained by over-fitting and a sub-optimal configuration. For instance, in all experiments of this series the minimum amount of examples in a leaf node (*minLeaf*) was set to 50 which is rather small when fitting a

Name	$\mathbf{K_c}$	$\mathbf{K_d}$	$\mathbf{K_c}$ dims	$\mathbf{K_d}$ dims	Comb	SMSE
GP (se)	SE		1-9			0.313
GP (se+g4)	SE	G4	1,2,4,5,8	3,6,7,9	+	**0.285**
GP (se+ov)	SE	OV	1,2,4,5,8	3,6,7,9	+	0.288
GP (se*g4)	SE	G4	1,2,4,5,8	3,6,7,9	*	0.304
GP (se*ov)	SE	OV	1,2,4,5,8	3,6,7,9	*	0.307
GP (se+g4)	SE	G4	1-8	9	+	0.293

Table 2: Results using Gaussian process regression with different kernel functions. See text for a more detailed analysis.

linear model of nine dimensions.

In a second modified version of the regression trees splitting is done by minimizing the distances in the input dimensions without taking target values into account (clustering). For the prediction in the leaves, a mean of target values of the *k*-nearest neighbors is used ($k = 10$). As can be seen from Table 1 this version was performing comparable to the standard approach by reaching a SMSE of 0.28.

Some experimental combinations of the mentioned techniques completed this evaluation. A clustering approach in combination with linear model fitting performed best. This is encouraging since it means that regression trees can be used as a sort of preclustering technique for other more powerful regression tools.

One of these powerful tools are Gaussian processes which are a kernel-based technique. In order to get a benchmark for plain GP regression performance we utilized the standard squared exponential kernel (SE, see above). We used all nine input dimensions to create the kernel matrix which is then used for the regression task. As can be seen in Table 2 we were able to reach a SMSE of 0.31 using this configuration. Note, that an automatic relevance determination and a hyper-parameter optimization using 100 iterations was applied.

As stated earlier some input dimensions appear in a more discrete way since their values only vary spatially between flux towers. To account for this setting, we choose to combine different kernel functions. On the one side, we continued to use SE kernel for the continuous dimensions (see Fig. 2). On the other side we used different distance functions from [2] for discrete variables, explicitly, the *Goodall4* (G4) and *Overlap* (OV) functions.

In Table 2 we report on results using these different techniques in a variety of combinations. We combined discrete kernels ($\mathbf{K}_d$) and continuous kernels ($\mathbf{K}_c$) in both a multiplicative and additive way. It turned out that the latter has a small performance advantage. We were even able to outperform the standard configuration using only SE kernel by reaching a SMSE of 0.28.

Name	Splitting	Prediction	minLeaf	K	SMSE
RDF-GP (se)	Variance	GP regr.	1000	SE	0.363
RDF-GP (se+g4)	Variance	GP regr.	1000	SE+G4	0.295
RDF-GP (se+ov)	Variance	GP regr.	1000	SE+OV	**0.286**

Table 3: Results using random forests with Gaussian process regression in leaves with different kernel functions.

3.4 Results for RDF-GP

In a final evaluation, we combined the above explained methods of random forests and Gaussian processes to account for the large amount of data that is to be processed in the actual interpolation task. Since creating the kernel matrix takes cubic time we used the regression trees of the random forest to pre-cluster the data. In each of the leaf nodes a GP regression is applied with a much smaller kernel size. We set the minimum amount of examples in a leaf node to 1000 and split the input dimensions for different kernels as in the best performing configuration of preceding experiments.

The experimental results of our RDF-GP approach are depicted in Table 3. As can be seen from the results, we were able to reach a SMSE of 0.29 using an additive combination of SE and G4 kernels. The combination of SE and OV was performing only slightly worse. However, it is worth noting that this configuration was advantageous in the GP experiments without pre-clustering using random forests. For all experiments in this series, we used standard splitting on the variance criterion in regression trees.

4 Conclusions

The overall objective of Work Package 4 is to derive novel synergistic products of EEVs and EFPs by integrating ground measurements and Earth Observation data with advanced mathematical methods.

This first deliverable of the WP 4 provides guidance regarding the upscaling of EEVS and EFPs using machine learning methods. Potential solutions for two different kinds of regression problems in WP4 were proposed.

In the case of EEVs, the main problem is the very large amount of data for training and prediction. Random Decision Forests, a well established and widely used tool, were suggested as a promising approach. This has already been implemented by MPI-BGC and the results will be presented in the respective report.

For the problem of upscaling ecosystem functional properties a method was designed which combines Random Forest together with a Gaussian Process model and tailored kernel functions to accommodate different types of predictor variables. The Gaussian Process regression performed in the leaf nodes of Random Forests provides uncertainty estimates of the prediction, as required by the project. Several tests with FLUXNET data were conducted to evaluate and identify an optimal setup of the algorithm.

References

[1] C. Beer. Terrestrial gross carbon dioxide uptake: Global distribution and covariation with climate. *Science*, 329:834–838, 2010.

[2] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the eighth SIAM International Conference on Data Mining*, pages 243–254, 2008.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] M. Jung et al. Recent decline in the global land evapotranspiration trand due to limited moisture supply. *Nature*, 467:951–954, 2010.

[5] M. Jung et al. Global patterns of land-atmosphere fluxes of carbon dioxide, latent heat, and sensible heat derived from eddy covariance, satellite, and meteorological observations. *Journal of Geophysical Research - Biogeosciences*, 116, 2011.

[6] A.D. Friend and N.Y. Kiang. Land surface model development for the giss gcm: Effects of improved canopy physiology on simulated climate. *Journal of Climate*, 18:2883–2902, 2005.

[7] B. Fröhlich, E. Rodner, M. Kemmler, and J. Denzler. Large-scale gaussian process classification using random decision forests. *Pattern Recognition and Image Analysis*, 22(1):113–120, 2012.

[8] J. E. Nash and J. V. Sutcliffe. River flow forecasting through conceptual models part i — a discussion of principles. *Journal of Hydrology*, 10(3):282–290, April 1970.

[9] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.