Probabilistic Machine Learning for Modeling Environmental Systems and their Uncertainties

Sebastian Haan

1 Content

2 Summary

This software transforms sparse soil data and surface measurements into spatial-temporal prediction maps of soil properties and their uncertainty. The model uses Gaussian Process regression with complex base functions and is particularly well-suited to agricultural applications because it can capture the underlying patterns and trends in soil data, as well as the inherent uncertainties associated with soil properties. The prediction models are implemented using Gaussian Processes (GP) with custom 3D and spatial-temporal kernels to take into account spatial-temporal correlations and multiple mean function options that depend on additional environment covariates (e.g. terrain, vegetation, top soil properties, depth). While the focus of this project is to make more accurate and reliable predictions of soil properties, and improve our understanding of the complex interactions between soil, weather, and crop growth, the same data-driven machine learning techniques can be applied for a wide range of environmental model applications.

3 Introduction

The use of data-driven machine learning techniques in agriculture has the potential to revolutionize the way we grow crops and manage our land. By leveraging the vast amounts of data that are generated by modern farming operations, we can gain valuable insights into the complex interactions between soil properties, weather, and crop growth.

One key area where data-driven machine learning can have a significant impact is in the prediction of soil properties and their uncertainties. Soil is a complex and dynamic environment, and accurately predicting its properties is critical for successful crop growth. Traditional approaches to soil prediction rely on costly and time-consuming laboratory tests, which can be difficult to scale and often provide only a snapshot of soil conditions at a specific point in time.

Data-driven machine learning, on the other hand, allows us to predict soil properties and their uncertainties using a range of data sources, including remote sensing data, weather data, and soil sensor data. By taking into account spatial and temporal correlations, we can build predictive models that provide more accurate and up-to-date information about soil conditions.

One powerful machine learning technique that can be used for soil property prediction are probabilistic Mixture Models using Gaussian Process Regression (GPR) models with multiple complex based functions. This model is particularly well-suited to agricultural applications because it can capture the underlying patterns and trends in soil data, as well as the inherent uncertainties associated with soil properties. By using probabilistic Mixture Models models, we can generate more accurate and reliable predictions of soil properties, which can be used to inform decision making and optimize crop management.

4 Method and Description of Probabilistic Framework

Here we propose a probabilistic framework that combines Gaussian Process (GP) model with complex multi-variate mean functions and sparse covariate 3D/spatial-temporal radial basis kernel to generate maps of soil properties and their uncertainties. Some of the main advantages and features of this method are:

- generation of predictive cubes (including uncertainty cubes and covariances) at any scale or resolution.
- options for complex mean function models and their uncertainty propagation to GP spatial covariance function.
- considering heteroscedatic input uncertainties (independent uncertainties for each location or measurement) uncertainties (can be location
- considering uncertainties in measurement positions (e.g. soil depth or temporal intervals)
- options to spatially integrate predictions and uncertainties over custom block sizes or area shapes (defined in polygon shapefiles)
- option to calculate change of soil properties and their uncertainties by taking into temporal and spatial covariances.

4.1 Gaussian Process Model

GPs are a flexible, probabilistic approach using kernel functions with non-parametric priors, and are successfully used in a large range of machine learning problems (see Rasmussen and Williams 2006 for more details). Some of the important advantages of the GP method are that it generates a predictive distribution with a mean $\mu(\Phi)$ and variance $\sigma^2(\Phi)$ and that the GP marginal likelihood function is well defined by the values of their hyper-parameters, which allows it to optimise them exactly. This reasoning about functions under uncertainty and their well-tuned interpolation character allows GPs to work extremely well for sparse data as well as for big data (see (Melkumyan and Ramos 2009) for solving large covariance matrix).

A GP

$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right)$

is completely determined by its mean function

$m(\mathbf{x}) = \mathrm{E}[f(\mathbf{x})]$ and covariance

$k(\mathbf{x}, \mathbf{x}') = \mathrm{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))\left(f(\mathbf{x}') - m(\mathbf{x}')\right)\right]$,

essentially placing a multivariate Gaussian distribution over the space of functions that map the input to the output, or informally, to measure the similarity between points as function of, e.g., their distance (the kernel function) and to predict a Gaussian distribution over $f(x^*)$ (i.e., a mean value and variance) at any new sampling location (unseen point) x^* from training data.

4.1.1 Sparse Covariance Function

The choice for an appropriate covariance function is important, as the GP's output directly depends on it. There are many stationary (invariant to translation in input space) and non-stationary covariance functions (for a review see Rasmussen and Williams 2006); the former are typically defined by radial basis functions (RBF) such as the squared-exponential covariance function,

$$k(x, x\prime) = \sigma_0^2 \exp\left(-\frac{1}{2l^2}(x - x')^2\right)$$

which depends on the radial distance between points x and x' and at least two parameters, one defining the length-scale l (multiple dimensions can have multiple length-scales) and the other the signal variance σ_0^2. These parameters of the covariance function are referred to as the hyperparameters Θ of the GP, which can be either given by a fixed covariance scale and noise, or learned from data by optimising the marginal likelihood. The latter allows us to optimise the GP hyperparameters such as the length-scale parameters l and the variance parameter σ_0^2. Computationally this is done by using scipy optimize. In general, it is advisable to compute the joint inversion at least for a range of l parameters and to compare the reconstructed models against each other.

To handle the computational problem of inverting a large covariance matrix, (Melkumyan and Ramos 2009) proposed an intrinsically sparse covariance function

$$\begin{cases} k(r, l, \sigma_0) = \sigma_0\Big[\frac{1}{3}(2 + \cos(2\pi\frac{r}{l}))(1 - \frac{r}{l}) + \frac{1}{2\pi}\sin(2\pi\frac{r}{l})\Big], & \text{if } r < l. \\ 0, & \text{if } r \geq l. \end{cases}$$

where $\sigma_0 > 0$ is a constant coefficient, $l > 0$ is a given scale and r is the distance between the points. This covariance function can be extended to multiple dimensions (MD) in the following two ways: 1) using direct products for all axes:

$$K_{MD}(\mathbf{x}, \mathbf{x}', \mathbf{l}, \sigma_0) = \sigma_0 \prod_{i=1}^{D} K(x_i, x_i', l_i, 1),$$

where D is the number of dimensions (here D=3) and $\mathbf{l}$ is the vector of the characteristic lengths. 2) using Mahalanobis distance:

$$\begin{cases} k(r, \sigma_0, \Omega) = \sigma_0\Big[\frac{1}{3}(2 + \cos(2\pi r))(1 - r) + \frac{1}{2\pi}\sin(2\pi r)\Big], & \text{if } r < 1. \\ 0, & \text{if } r \geq 1, \end{cases}$$

where

$$r = \sqrt{(\mathbf{x} - \mathbf{x}')^T \Omega (\mathbf{x} - \mathbf{x}')}$$

and with $\mathbf{\Omega}$ expressed via diagonal matrix of characteristic lengths:

$$\mathbf{\Omega} = [\frac{1}{l_1}, \frac{1}{l_2}, ..., \frac{1}{l_D}]\mathbf{I_{D,D}}$$

it follows:

$$r = \sqrt{\sum_{k=1}^{D} \left(\frac{x_k - x'_k}{l_k} \right)^2}.$$

4.1.2 Including Measurement and Positional Uncertainties

To take into account that positions of the measurements i and j are not point coordinates but have an uncertainty (here along the vertical direction), dX_i and dX_j, we need to modify the covariance function by assuming for the vertical coordinate $X_z = N(\mu(X_z)|dX_z)$, and it follows that

$$K(r, l, \sigma_0) = \frac{\sigma_0}{|1 + l^{-1}(dX_i + dX_j)|^{1/2}} \Big[\frac{1}{3}(2 + \cos(2\pi \frac{r}{l + dX_i + dX_j}))(1 - \frac{r}{l + dX_i + dX_j}) +$$

$$\frac{1}{2\pi} \sin(2\pi \frac{r}{l + dX_i + dX_j}) \Big],$$

if $r < l$, and $K(r, l, \sigma_0) = 0$ if $r >= l$.

In addition we have to include measurement uncertainties as a noise term (or regularisation parameter) added to the covariance function:

$$\mathbf{K} = \mathbf{k} + {\sigma_\mathbf{s}}^2 \mathbf{I}$$

where the uncertainties σ of the sensor observations $\mathbf{y}$ are included and can be either a constant σ (same uncertainty for all sensors) or a vector (σ_s) that is given by the individual uncertainty of each sensor observation. In case the sensor uncertainty is unknown, it can be estimated in conjunction with the other GP hyperparameters by maximizing the marginal log-likelihood (see next section).

4.1.3 Hyperparameter Optimisation

One advantage of the Bayesian framework is the ability to calculate the marginal log-likelihood,

$$\mathcal{L}(\Theta) = -\frac{1}{2}\mathbf{y}^T \mathbf{K}_{x,x}^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K}_{x,x}| - \frac{\log 2\pi}{2} \sum_{i=1}^{M} N_i$$

to optimize the parameters Θ of the GPs, which include the hyper-parameters (amplitude, kernel length-scales) of the covariance functions $K(x, x)$, and optionally the noise variances σ_s^2 if not given as input data.

4.1.4 Predictive Distributions and Uncertainties

The predictive distribution y' at new locations x' is given by

$$y'|\mathbf{G}, \mathbf{y} \sim \mathcal{N}(\phi|\mu_{y'|y}, \mathbf{\Sigma}_{\phi|y})$$

with the predictive mean at each location

$$\mu = \mathbf{K}_{x,x'}^T \mathbf{K}_{x,x}^{-1} \mathbf{y}$$

and the predicted covariance

$$\mathbf{\Sigma} = \mathbf{K}_{x',x'} + \sigma_s^2 I - \mathbf{K}_{x,x'}^T \mathbf{K}_{x,x}^{-1} \mathbf{K}_{x,x'}.$$

The predicted variance is given by the diagonal elements of the covariance matrix.

4.2 Mean Functions

If the mean function is not zero but given by a predictive deterministic function $m_\phi = f(\mathbf{x_c})$, the predicted GP posterior distribution is given by

$$y'|\mathbf{G}, \mathbf{y} \sim \mathcal{N}(m_{\phi'} + \mathbf{K}_{x,x'}^T \mathbf{K}_{x,x}^{-1} \mathbf{y} - \mathbf{m}_\phi, \mathbf{K}_{x',x'} + \sigma_s^2 I - \mathbf{K}_{x,x'}^T \mathbf{K}_{x,x}^{-1} \mathbf{K}_{x,x'})$$

The covariates x_c are mostly surface measurements such as DEM, terrain slope, NDVI, radiometrics. The uncertainty of the mean model function is used to construct the diagonal noise matrix of the GP $\sigma_\mathbf{s}$ and is proportional up to a constant scaling (determined by the hyperparameter optimisation).

4.2.1 Bayesian Linear Regression

First, we standardise the data and apply a feature-wise power transform scaler via scikit-learn implementation (default option). Power transforms are a family of parametric, monotonic transformations that are applied to make data more like normal distributed. This is useful for modeling issues related to heteroscedasticity (non-constant variance), or other situations where normality is desired. In detail, the Yeo-Johnson transform (Yeo and Johnson 2000) is applied, which supports both positive or negative data. Note that the Bayesian Regression implementation implemented here is not limited to power transform scaler only, but includes multiple options of scalers, such as StandardScaler or RobustScaler.

After the feature-wise scaling of data, a Bayesian Ridge regression is performed using the scikit-learn implementation 'BayesianRidge'. This particular Bayesian ridge implementation is based on the algorithm described in Appendix A of Tipping (2001) where updates of the regularization parameters are done (MacKay 1992). To make the regression more robust, we add an automated feature selection after the initial regression by selecting only features with a ratio of correlation coefficient to standard deviation larger than one. Then a second BayesianRidge regression is made using only the selected features, and the model and coefficients are stored with non-significant coefficients set to zero. Predictions and their uncertainties are then obtained by using the trained model and are scaled back with the powerlaw transform.

4.2.2 Random Forest

The scikit-learn Random Forest model implementation is applied. No data scaler required. Prediction uncertainties are currently estimates by using the standard deviation and Confidence Intervals of all decision trees.

4.3 Model Accuracy and Residual Analysis

The residual analysis is performed on the validation data with a split with 10 cross validations. The residual error $RE = Y_{pred} - Y_{ground}$ is defined as the predicted value minus the ground truth for each data point in the validation set. The normalised Root-Mean-Square-Error ($nRMSE$) is computed as RMSE divided by the standard deviation of the validation data,

$$nRMSE = \frac{\sqrt{\frac{1}{n}\sum RE^2}}{\sigma_{Y_{ground}}}.$$

The coefficient of determination R^2 is based on the variance explained by the model compared to the total variance, which is defined as

$$R^2 = 1 - \frac{\sum(RE^2)}{\sum(Y_{ground} - \frac{1}{n}\sum Y_{ground})^2}.$$

The lower the $nRMSE$ value, or alternatively, the higher R^2, the better is the prediction. To test whether the predicted uncertainties are consistent with the residual error of the prediction, we calculate the ratio

$$\theta = \frac{1}{n}\sum \frac{RE^2}{\sigma^2_{Ypred}}.$$

5 Feature Importance and Selection

Feature importance can be used for inference as well as feature selection, which is based on their importance ranking or their respective statistical significance. However, feature importance is dependent on which model is applied and can show significant difference for different models, in particular if non-linear or feature interaction effects are taken into account by the model (or not). The workflows here use a multi-model Feature Importance scoring, currently the following models for feature importance scoring are included: - Spearman rank analysis (see 'selectio.models.spearman') - Correlation coefficient significance of linear/log-scaled Bayesian Linear Regression (see 'selectio.models.blr') - Random Forest Permutation test (see 'selectio.models.rf') - Random Decision Trees on various subsamples of data (see 'selectio.models.rdt') - Mutual Information Regression (see 'selectio.models.mi') - General correlation coefficients (see 'selectio.models.xicor')

5.1 Spearman rank analysis

The Spearman rank-order correlation coefficient is a nonparametric measure of the monotonicity of the relationship between two datasets. Unlike the Pearson correlation, the Spearman correlation does not assume that both datasets are normally distributed. Like other correlation coefficients, this one varies between -1 and +1 with 0 implying no correlation.

5.2 Bayesian Linear Regression correlation significance

Features are ranked based on their statistical significance which is given for each feature by the ratio of the amplitude to standard deviation of the regression correlation coefficient. The correlation coefficients can be estimated for both, the linear Bayesian Ridge Regression and power-law scaled Bayesian Ridge Regression.

5.3 Random Forest with permutation based feature importance

Feature importance based on impurity can be misleading for high cardinality features (many unique values). An alternative is feature importance based on feature permutation which has no bias towards high-cardinality features and can be computed on a test set. Basically all features are shuffled n-times and the model is refitted to estimate the importance. The permutation feature importance is then defined to be the decrease in a model score when a single feature value is randomly shuffled. Note that this technique is model agnostic. The draw-back is that this permutation method is computationally more costly than others.

5.4 Random Decision Trees

Similar to Random Forest permutation. Here the factor importance is implemented using randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy.

5.5 Mutual Information

This function relies on nonparametric methods based on entropy estimation from k-nearest neighbors distances as described in (Ross 2014). The mutual information (MI) between two random variables is a non-negative value, which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency.

5.6 General correlation coefficients

This method is based on association measurement through cross rank increments and returns correlation coefficient for testing independence. The calculation of this general correlation coefficient is described in "A new coefficient of correlation"(Chatterjee 2021). In comparison to Spearman, Pearson, and Kendalls, this new correlation coefficient can measure associations that are not monotonic or non-linear, and is 0 if X and Y are independent, and is 1 if there is a function Y= f(X). Note that this coefficient is intentionally not symmetric, because we want to understand if Y is a function X, and not just if one of the variables is a function of the other.

6 References

Chatterjee, Sourav. 2021. "A New Coefficient of Correlation." *Journal of the American Statistical Association* 116 (536): 2009–22.

MacKay, David JC. 1992. "Bayesian Interpolation." *Neural Computation* 4 (3): 415–47.

Melkumyan, Arman, and Fabio Ramos. 2009. "A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets." In *IJCAI*, 9:1936–42.

Rasmussen, Carl Edward, and Christopher KI Williams. 2006. *Gaussian Process for Machine Learning*. MIT press.

Ross, Brian C. 2014. "Mutual Information Between Discrete and Continuous Data Sets." *PloS One* 9 (2): e87357.

Tipping, Michael E. 2001. "Sparse Bayesian Learning and the Relevance Vector Machine." *Journal of Machine Learning Research* 1 (Jun): 211–44.

Yeo, In-Kwon, and Richard A Johnson. 2000. "A New Family of Power Transformations to Improve Normality or Symmetry." *Biometrika* 87 (4): 954–59.