Supplementary Information

High-resolution digital soil mapping of multiple soil properties: an alternative to the traditional field survey?
Trevan Flynn, Willem de Clercq, Andrei Rozanov and Cathy Clarke
South African Journal of Plant and Soil 2019. https://doi.org/10.1080/02571862.2019.1570566

Appendix S1: Feature selection implementation

All covariates were centred and scaled before running the feature selection. The univariate feature selection (UFS) fits a model for many iterations after filtering of the covariates. Covariate filtering selects covariates by determining the correlation of individual covariates to the soil property of interest. The robust feature selection (RFS) takes all the covariates and progressively eliminates each until the error rate reaches an optimal level. No tuning parameters were optimized for either technique.

The least absolute shrinkage and selection operator (LASSO) is a generalized linear model which minimizes covariate coefficients based on the absolute error of the residuals (L_1 regularisation) through coordinate descent (Friedman et al., 2010). This process shrinks covariate coefficients which are correlated to one another. The degree of shrinkage is controlled by the λ value which was optimised and the covariates which did not have an absolute value of zero, were selected. A LASSO feature selection was implemented because LASSO is efficient with high dimensional data sets, improves model interpretation, and does not substantially increase bias (Tibshirani, 1996).

A boosted linear model (Boost) is a novel feature selection technique suitable for high dimensional data sets (Bühlmann & Hothorn, 2007). It fits component-wise linear models as base learners and is boosted by correcting for the squared error of the residuals (L_2 regularisation). However, unlike L_1 regularisation, the coefficients are not shrunk to zero and the method of feature selection is a “black box” with little known as to how it selects the covariates. The number of boosts was optimised with pruning.

Predictive model implementation

A RR is a generalized linear model which maximises the likelihood via L_2 regularisation through coordinate descent (Friedman et al., 2010). The λ value was the only parameter optimised. An LBM is an additive model with linear step-wise base learners.

The number of boosts were optimised with pruning. The QR was implemented through regression on the median with no tuning parameters.

Both linear and radial kernel SVM were implemented to evaluate both linear and non-linear relationships. Support Vector Machines have been known to perform well for classification, however, these have been adapted to perform regression tasks. The cost function and sigma values were both optimised. The cost controls the error function of the model. The sigma value determines the width of the gaussian distribution for the radial kernel.

Random forest is a decision tree ensemble model which grows trees in parallel and the final prediction is the mean of the prediction for all trees grown (Breiman, 2001). The number of covariates randomly chosen at each split was optimised and the number of trees grown was held constant at 1000 trees. The number of trees was held at 1000 because Breiman (2002), states that at least 1000 trees are required for a stable variable importance measure. Random forest was used because it is suitable for small and large data, can handle non-linear relationships, and is robust against over fitting (Breiman, 2001).

Stochastic gradient boosting is a type of decision tree ensemble which creates decision trees in sequence rather than parallel (Friedman, 2001, 2002). Therefore, the model builds decision trees to correct for the errors of the previous decision tree. The SGB algorithm implements a gaussian exponential loss function through Friedman gradient decent (Friedman, 2001). The learning rate, minimum number of observations in each terminal node, and the bag fraction were held constant at 0.01, ten observations, and 0.5 resamples, respectively. However, the number of trees grown and number of interactions was optimised. A SGB was used because it represents an alternative to RF and has been shown to achieve similar accuracies (Forkuor et al*.,* 2014).

The Cubist algorithm is a rule based model which runs linear regression as a smoothing parameter (Quinlan, 1993). The cubist model is similar to an ensemble of decision trees; however, linear regression is performed at each node. The cubist model has two main tuning parameters. The number of committees is the number of trees grown in sequence (like boosting). The number of neighbours is the number of k-nearest neighbours used to correct for errors. The cubist model was selected

because cubist is a complex yet an interpretable model as the output defines each rule made.

Penalized boosted splines is an additive model which uses splines as a smoothing base learner for each covariate and the model is boosted on the residuals (Bühlmann & Hothorn, 2007). The number of boosts was optimised with pruning, knots were set to 20, and degrees of freedom set to four. The knots and degrees of freedom values were held constant based on the recommendations of Bühlmann and Hothorn (2007). Penalized boosted splines were implemented because it is a novel algorithm which has been shown to be a powerful tool in machine learning competitions (Taieb & Hyndman, 2013).

References

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

Breiman, L. 2002. Manual on setting up, using, and understanding random forests v3.1. Technical Report. Available at http://oz.berkeley.edu/users/breiman, Statistics Department University of California Berkeley.

Bühlmann, P. & Hothorn, T. 2007. Boosting algorithms: regularization, prediction and model fitting. *Statistical Science* 22(4): 477–505.

Forkuor, G., Hounkpatin, O.K.L., Welp, G. & Thiel, M. 2017. High resolution mapping of soil properties using remote sensing variables in south-western Burkina Faso: a comparison of machine learning and multiple linear regression models. *PLOS One* 12(1): e0170478.

Friedman, J.H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics* 29(5): 1189–1232.

Friedman, J.H. 2002. Stochastic gradient boosting. *Computational Statistics and Data Analysis* 38: 367–378.

Friedman, J., Hastie, T. & Tibshirani, R. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33(1): 1–22.

.

Hitziger, M. & Ließ, M. 2014. Comparison of three supervised learning methods for digital soil mapping: application to a complex terrain in the Ecuadorian Andes. *Applied and Environmental Soil Science* 2014: Art. ID 809495.

Quinlan, J.R. 1993. Combining instance-based and model-based learning. *Machine Learning* 76: 236–243.

Taieb, S. Ben & Hyndman, R.J. 2013. A gradient boosting approach to the Kaggle load forecasting competition. *International Journal of Forecasting* 30(2): 382–394.

Tibshirani, R. 1996. Regression selection and shrinkage via the Lasso. *Journal of the Royal Statistical Society* 58(1): 267–288.