Computers & Geosciences 83 (2015) 80–88

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Case study

Using Google's cloud-based platform for digital soil mapping

J. Padarian*, B. Minasny, A.B. McBratney

Faculty of Agriculture and Environment, Department of Environmental Sciences, The University of Sydney, New South Wales, Australia

ARTICLE INFO

Article history:
Received 2 January 2015
Received in revised form
29 June 2015
Accepted 30 June 2015
Available online 10 July 2015

Keywords:
Supercomputing
DSM
Cloud computing
GlobalSoilMap
Google Earth Engine

ABSTRACT

A digital soil mapping exercise over a large extent and at a high resolution is a computationally expensive procedure. It may take days or weeks to obtain the final maps and to visually evaluate the prediction models when using a desktop workstation. To increase the speed of time-consuming procedures, the use of supercomputers is a common practice. Google™ has developed a product specifically designed for mapping purposes (Earth Engine), allowing users to take advantage of its computing power and the mobility of a cloud-based solution. In this work, we explore the feasibility of using this platform for digital soil mapping by presenting two soil mapping examples over the contiguous United States. We also discuss the advantages and limitations of this platform at its current development stage, and potential improvements towards a fully functional cloud-based soil mapping platform.

1. Introduction

Current soil mapping heavily depends on the use of technology and computers, starting with the use of computer-based geographic information systems (GIS) in the late 1960s (Coppock and Rhind, 1991). This rudimentary system rapidly evolved towards current digital soil mapping (DSM) which uses sophisticated data mining techniques combined with environmental information. DSM makes the organisation and visualisation of spatial data more efficient, giving the opportunity to perform complex analysis and database management, virtually replacing the manually drawn map polygons with a soil information system.

The rapid progress in digital soil mapping is partly motivated by the increasing interest in soil information at regional or national scales and the implicit economic constraint of fieldwork and laboratory analysis. Scientists have refreshed the methods to assess soil resources using tools like the aforementioned GIS, in conjunction with global positioning system (GPS), environmental information in the form of raster maps and remote sensing technology (McBratney et al., 2003). McBratney et al. (2003) made an extensive review of soil predictive models and formalised the digital soil mapping framework, proposing an empirical model to predict, and not only explain, soil properties or soil classes based on its soil-forming factors (Scorpan factors; i.e. other soil properties, climate, organisms, topography, parent material, age and space). The so-called Scorpan factors can usually be represented in a digital form, usually as raster data. The mapping is achieved by building a spatial soil prediction function which relates observed soil variables with Scorpan factors. This technique is called Digital Soil Mapping (DSM). Digital mapping techniques homologous to DSM also are used in other earth science disciplines, such as mapping landslides (Farahmand and AghaKouchak, 2013) or mapping opal occurrences (Landgrebe et al., 2013).

For applications at a field or landscape extent, conventional desktop workstations are powerful enough to manage data, imagery, build models, and generate prediction maps. However, desktop workstations quickly become inefficient when moving to the application of high-resolution mapping to large spatial extents such as regions, continents, or the world. For example, the Landsat 7 project, which started in 1999, captures about 300 images per day, and is equivalent to more than 1 petabyte (1 petabyte $=10^6$ gigabytes) of raster data. Managing this volume of information to generate a spatio-temporal model at a continental or global extent is not a trivial task.

As a response to this challenge, Google™ is developing a platform capable of storing and analysing raster data using its computing infrastructure where algorithms are designed to run in multiple processors simultaneously. The aim of this work is to explore the feasibility of using this platform for DSM. We will demonstrate its application for mapping soil properties and classes, and discuss the advantages and limitations of this platform at its current stage of development. Finally, we propose potential improvements towards a fully functional cloud-based digital soil mapping platform.

* Corresponding author.
E-mail addresses: jose.padarian@sydney.edu.au (J. Padarian), budiman.minasny@sydney.edu.au (B. Minasny), alex.mcbratney@sydney.edu.au (A.B. McBratney).

http://dx.doi.org/10.1016/j.cageo.2015.06.023
0098-3004/

2. Google Earth Engine

Google Earth Engine (GEE), Google's geospatial data analysis platform, provides access to a large catalogue of Earth observation data, algorithms for analysing the data, and a programming interface to create and run custom algorithms. Computations in GEE are done using Google's infrastructure, and analyses are automatically parallelised so that many computer processors may be involved in any given computation. This automatic parallelisation enables global-scale analyses such as that by Hansen et al. (2013), which identified global forest change between the years 2000 and 2012 at a resolution of 30 m. The study analysed 654,178 Landsat 7 scenes, a total of 707 terabytes of data, which would have taken 1,000,000 hours if the work were done by one processing unit. However, because the analysis was run in parallel over 10,000 machines, it only took approximately 100 hours to complete.

Earth Engine's data catalogue brings together petabytes of Earth observation data, eliminating the need to download and manage data locally; typically a significant proportion of the work involved in performing large-scale geospatial analyses. Earth Engine stores the source data in its original projection, with all the original data and metadata. Information from different data sources can also be combined as Earth Engine can reproject data as needed. The data catalogue contains a nearly complete set of data from the Landsat 4, 5, 7, and 8 satellites downloaded from the USGS Earth Resources Observation and Science archive, MODIS satellite datasets, a number of digital elevation models, atmospheric data, meteorological data, and several pre-classified global land-cover datasets. Earth Engine also makes many derivative products available, such as annual mosaics and a variety of environmental indices (e.g. NDVI, slope). However, since data storage is considerably more expensive than data processing, most of these derivative products are computed upon request rather than pre-computed and stored. This "on-the-fly" analysis is a key aspect of Earth Engine which allows for rapid prototyping of complex algorithms. In the current implementation it is only possible to perform per-pixel operations, excluding more complex terrain analysis like flow direction, flow accumulation, and watershed findings. Until now, Earth Engine has been applied in just a few domains, including analysing cloud masks (Wilson et al., 2014) , urban growth (Burillo and Chester, 2013), roadless areas (Ibisch, 2013), and mangrove mapping (Giri et al., 2015).

Currently, GEE is available as a limited release, however any scientist can register for free to use the platform and participate towards its development. It is also worth noting that the work by Hansen et al. (2013) was performed in collaboration with Google, with special resources assigned to the task, which are not available for every user.

3. Conventional DSM vs. GEE

Independent of the extent, spacing, spatial support and aim, DSM usually has a standard workflow where we can identify five main steps, given the soil observations at a defined area:

(1) extract covariate data at the observation locations;
(2) compile the soil and covariate data and build a regression or classification model;
(3) apply the model to unknown locations at the whole extent;
(4) display and analyse the results; and
(5) distribute the final maps.

In this section we compare a traditional DSM routine with the alternative that GEE offers.

Table 1
List of some rasters currently available[a] in GEE.

Type	Dataset
Satellite imagery	Landsat 4, 5, 7, and 8. Datasets derived from MODIS – including surface reflectance data, vegetation continuous fields, albedo and land surface temperature Samples of high resolution data from Digital Globe and Astrium
Digital elevation model	SRTM (Jarvis et al., 2008) NED (Gesch et al., 2002) GTOPO (Gesch and Larson, 1996)
Meteorological	NLDAS (Xia et al., 2012a, b) GRIDMET (Abatzoglou, 2013)
Miscellaneous	FIRMS fire data Landcover classifications (NLCD, GlobCover, MOD12Q1) Hydrosheds (Lehner et al., 2008) Global mangrove data (Giri et al., 2011)

[a] New datasets are frequently uploaded. For a complete list check GEE data catalogue (http://earthengine.google.org/).

3.1. Extraction of covariates

In DSM, it is usually necessary to retrieve covariates at the locations of soil observations to represent soil-forming factors which could be a challenging process depending on the number of covariates that are available. In traditional DSM this usually implies

(a) downloading raster from websites or ftp servers,
(b) calculating the derivatives when necessary,
(c) adjusting resolutions, and finally
(d) extracting the information for the specific area.

With GEE the process can be done in a much simpler way, as (a) many covariates are already available in the server (Table 1), (b) calculation of derivatives is performed in bulk and in parallel, (c) resolutions are managed directly by the platform, and (d) data extraction is performed in the server. GEE manages the resolutions by storing rasters in their native resolution (e.g. 30 m) and then generates down-sampled (by mean) versions at multiples of 2 (e.g. 60 m and 120 m). When the user requires another resolution, GEE fetches the down-sampled pixels from the next level up, and resamples those to the desired resolution.

Until now, Google has made available raster datasets that have a compatible licence (e.g. CC-BY). If a raster is not already available in GEE, it is possible to upload it to an associated Google Maps Engine[1] (GME) account to access it from GEE. This is an extra step, not present in a traditional DSM exercise. If we obviate the time required to upload the file (which depends on file size and connection speed), and the time required for Google to process it (in our experience, about 10–15 minutes for a 800 MB file), there is still a significant limitation in using high resolution images because the default (free) GME account just includes 10 GB for storage.

Besides uploading rasters that are not available in GEE, it is also necessary to upload the soil observations. This can be done creating a Fusion Table (in an associated Google Drive account). The extra setup time incurred by this step depends once again on

[1] Google Maps Engine will be depracated on July 29th, 2016. Google is currently working on an alternative based on Google Drive and Google Cloud Storage (communicated via the Earth Engine Developers group).

Table 2
List of models currently available in GEE.

Model	Use	Description
FastNaiveBayes	Classification	Fast classification algorithm that uses maximum likelihood estimation to evaluate the probability of an observation belonging to a specific category, assuming that the predictor variables are independent (Rish, 2001)
GmoMaxEnt	Classification	Implementation of a maximum entropy classifier, also known as multinomial logistic regression (Böhning, 1992). Similar to Fast Naive Bayes, without the assumption of predictor variable independence
MultiClassPerceptron	Classification regression	Implementation of neural networks based on the algorithm originally proposed by Rosenblatt (1958). The algorithm creates a network of functions connected by weights, simulating the neural structure of the human brain
Winnow	Classification regression	Implementation similar to the perceptron algorithm but with a multiplicative weight-update scheme (traditional perceptron uses an additive scheme) (Littlestone, 1988)
Cart	Classification regression	Implementation of the Classification and Regression Trees algorithm (Breiman et al., 1984) where the data is partitioned maximising the difference between groups, generating a tree-like structure of nodes (classes). In the regression case, a linear model is fitted at each end-node
RifleSerialClassifier	Classification regression	Implementation of the Random Forest algorithm (Breiman, 2001). The algorithm constructs multiple trees (i.e. CART) and generates an ensemble output by estimating the mode or average for classification and regression respectively
Pegasos	Classification regression	Implementation of an iterative algorithm to improve speed of support vector machine (SVM) optimisation (Shalev-Shwartz et al., 2011)
IKPamir	Classification regression	SVM implementation optimised using intersection kernel (Maji et al., 2008) and Passive-Aggressive Algorithms (Crammer et al., 2006)
VotingSvm	Classification regression	SVM implementation where values are assigned by selecting the value of the class with highest number of appearances (mode)
MarginSvm	Classification regression	SVM implementation where the algorithm chooses the value with the maximum margin across all classifiers

the amount of data and internet connection speed, and in this work was approximately 5 minutes.

3.2. Model building

In this step of the DSM workflow it is necessary to select an appropriate model (or data-mining algorithm) to capture the relationships between the covariates and the soil properties to predict. In DSM, models such as linear regression, regression kriging, neural networks, and regression and classification trees are typically used (Adhikari et al., 2014). In contrast, GEE is limited to selected data mining models (Table 2). The models cover both prediction of class (categorical) and continuous properties. Some of the models are variants of models commonly used in DSM, such as classification and regression trees, artificial neural networks, and random forests.

In our experience, the platform is not flexible enough to build models if compared with the traditional DSM, which is usually performed using specialised software like *R*. It is worth noting that it is possible to perform this step outside of GEE and generate expressions that GEE can interpret. This gives more flexibility, specially to perform exploratory data analysis, but it is restricted to the use of models that can be written explicitly, such as logical partitions (e.g. tree-like structures) and prediction formulas (e.g. linear models).

3.3. Apply the fitted model

In the traditional DSM approach, it is necessary to extract the covariate information at the defined grid spacing over the whole mapping area and perform the prediction of the soil property for the whole extent. Depending on the final resolution (grid spacing) and the extent, this is the most computationally challenging step of the process. For high pixel density (big areas and/or fine resolutions) it is common to work with tiles (sub-areas) due to RAM constraints (Padarian et al., 2014; Hengl et al., 2014).

GEE works under the same tile logic, performing the extraction of covariates and subsequent prediction by tiles, which are handled directly by the platform, and distributing each tile to an independent process in parallel. This parallelisation can also be performed in modern workstations or computer nodes, but the implementation is not trivial and access to the latter infrastructure is not always possible.

Here is the major advantage of GEE, and we attribute most of the speed gain to this step. A more detailed comparison between the time required to complete a mapping exercise using traditional DSM and GEE is presented in Section 4.

3.4. Map display

This additional step is usually carried out in a GIS platform, where the performance is constrained by the workstation capabilities. By default, most GIS platforms would try to draw as many pixels as possible on screen, making the process of panning around the interest area extremely slow. The solution implemented in GIS platforms generally consists of pre-generating lower resolution versions of the raster (overviews) once the final map is produced, and displaying the most suitable resolution depending on the zoom level.

This overviews system is implemented by default in GEE, where each tile is always 256×256 pixels, and it is recalculated (and stored in cache) with every change in zoom level. This step is directly related with the previous one because GEE only applies the fitted model to the extent of the viewport (area displayed on screen) at the specific zoom level (each zoom level with a maximum resolution). This feature allows the user to quickly visualise a preview of the final map before exporting or downloading it.

3.5. Distribution

Most of the tasks performed in conventional DSM are local (in contrast with an online solution), so there is a need to distribute final maps. In most cases the obvious solution is to share this content online as a simple file available for the user to download, or to implement a solution using one of the many available mapping services, with the consequent hosting expense.

As we mentioned in Section 3.1, GEE is closely related with Google Maps Engine. The latter is specially developed to store and share maps online. In addition, the code and soil observations used to generate the map can be shared, which allows other researchers to replicate the mapping process. Expanding this idea, Heuvelink et al. (2010) proposed that it is possible to create a "next generation" soil information system (SIS+) where only models are stored, and the maps are generated "on-the-fly". GEE presents a potential to be this SIS+, if some of the current limitations are addressed.

4. Case study

In this section we demonstrate how GEE handles two common tasks in soil mapping: prediction of a categorical property (soil class) based on a classification algorithm, and prediction of a continuous property (soil organic carbon content) via a regression technique. We selected the contiguous United States for both cases because it has good data coverage over the whole extent. This case study is just for illustration purposes and to evaluate the usefulness of the platform in mapping large areas.

The model training and map generation process took about 2 minutes (at a 1 km resolution) in both cases, with a relatively simple coding process. If we were to perform a similar task in a regular workstation, it would take about 1.5–3.5 hours if we take into account the 0.86–1.73 ms km^{-2} at 1 km resolution for a global map (considering 149.1×10^6 km^{-2} of land surface) reported by Hengl et al. (2014), or the 125 ms km^{-2} at 100 m resolution estimated by Padarian et al. (2014).

4.1. Mapping soil classes

To test the classification features of GEE, we tried to recreate the map of soil orders in the United States based on digital mapping techniques. We used 10,000 profiles (Fig. 1) extracted from the National Cooperative Soil Survey Characterization Database (http://ncsslabdatamart.sc.egov.usda.gov) using a Rifle Serial Classifier algorithm (which is similar to Random Forest). The covariates used were elevation (SRTM, Jarvis et al., 2008); slope; minimum, maximum and mean annual air temperature; and annual precipitation (for year 2010, GRIDMET, Abatzoglou, 2013). All the covariates are currently available in GEE, and slope is calculated "on-the-fly".

The final map (Fig. 2) is a good representation of the distribution of soil orders when compared with the US soil orders map from The Natural Resources Conservation Service (available online), with a classification accuracy of 70.07% (Kappa Stat.: 0.63. See confusion matrix in Appendix B). Two soil orders are not represented: (a) Oxisols, because they are not present in the contiguous United States, and (b) Gelisols, which were not present in the database used to generate the map. In addition, there are very few observations in the state of Florida (Fig. 1) making the map highly uncertain in this area (to be discussed in Section 5.2).

4.2. Mapping continuous variables – topsoil organic carbon

For the regression example we used 29,784 soil profiles (Fig. 3) from the NASIS database (USDA–NRCS, 2014) which contains laboratory measurements of topsoil organic carbon content (0–5 cm depth), harmonised using an equal-area spline function (Bishop et al., 1999). As in the previous mapping example, we used elevation; slope; minimum, maximum, and mean annual air temperature; and annual precipitation as covariates using a Classification and Regression Tree algorithm.

The final map (Fig. 4) shows a good representation of the distribution of SOC considering the observations spatial coverage (nearly complete). Model performance (R^2: 0.20) was poor, but corresponded with results reported in similar studies at this scale (Henderson et al., 2005), and did not differ from results obtained using a desktop workstation.

5. Towards a cloud-based soil mapping platform

The GEE platform is under "beta-testing", with improvements happening daily. The advantage of being in this phase is that current users can help to improve the platform and suggest new features that might be relevant for the scientific community.

Most of the limitations we experienced during the development of this work were related with the "beta-testing" condition of the platform. For example, we experienced difficulty when

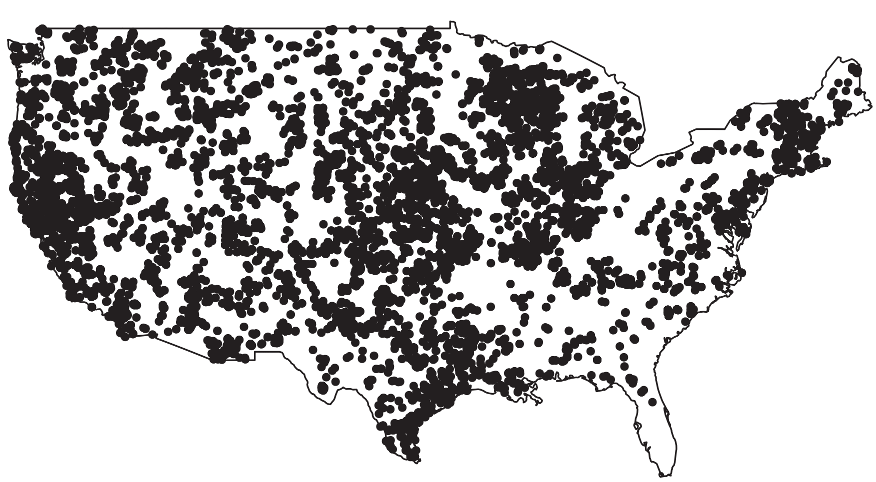

Fig. 1. Location of observations used in classification example.

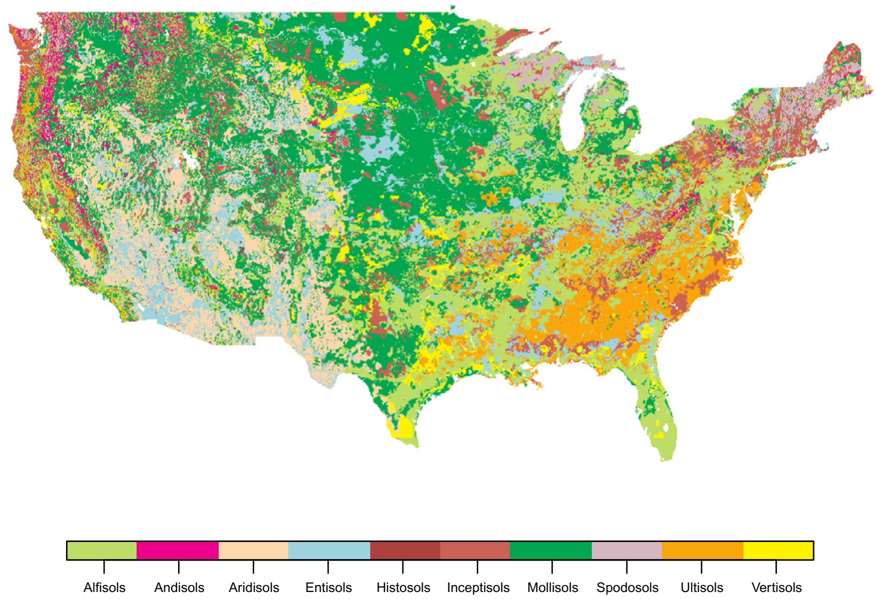

Fig. 2. Map of soil orders generated using GEE implementation of a Rifle Serial Classifier algorithm.

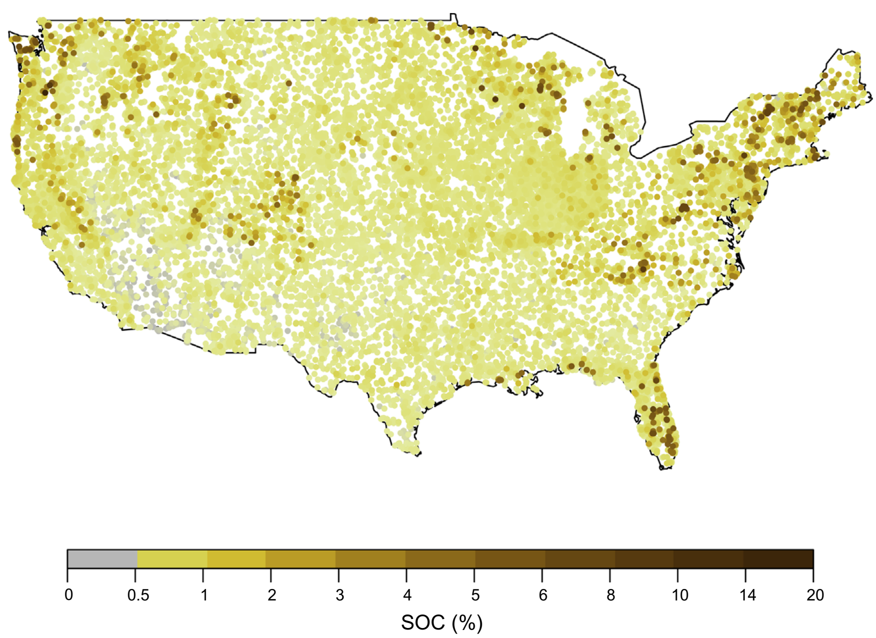

Fig. 3. Location of observations used in regression example.

working on a large set of observations (more than 20,000 observations), mainly because we exceeded the current limits of the platform. To overcome this problem, it was necessary to run the algorithm two or three times to complete the analysis.

With the examples presented in this work, we attempted to perform an entire DSM exercise, using GEE built-in algorithms to take advantage of their parallel computing capabilities. This was not possible, but we always had the option to download the intermediate results at any point of the workflow to continue the analysis from a desktop workstation. At present, we can perform data extraction, model fitting, and spatial prediction efficiently, however, there are some methodological gaps that need to be addressed before we can use GEE as a fully functional DSM tool. The gaps are mainly related to geostatistics and uncertainty analysis, on which we will elaborate in the next sections.

Another important consideration to have with this or any other

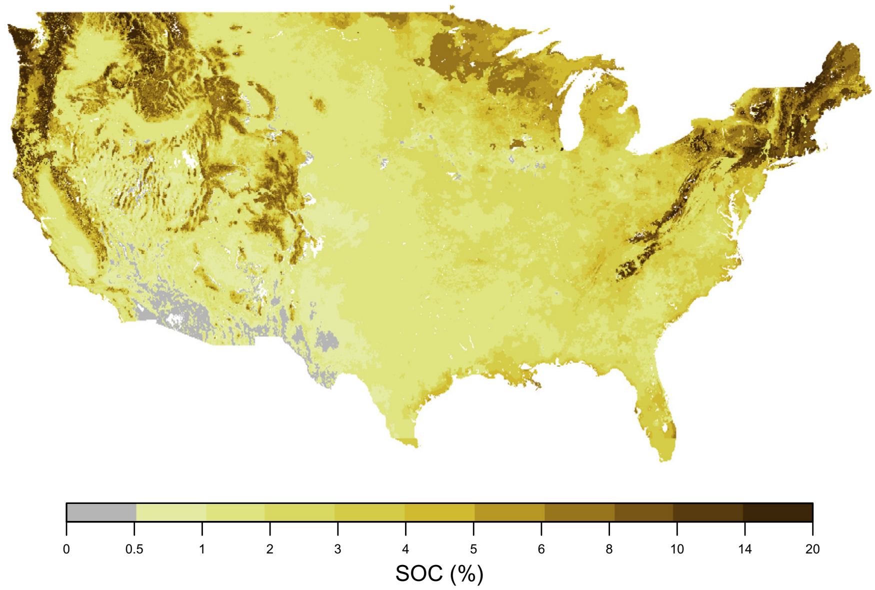

Fig. 4. Map of soil organic carbon generated using GEE implementation of a Classification and Regression Tree algorithm.

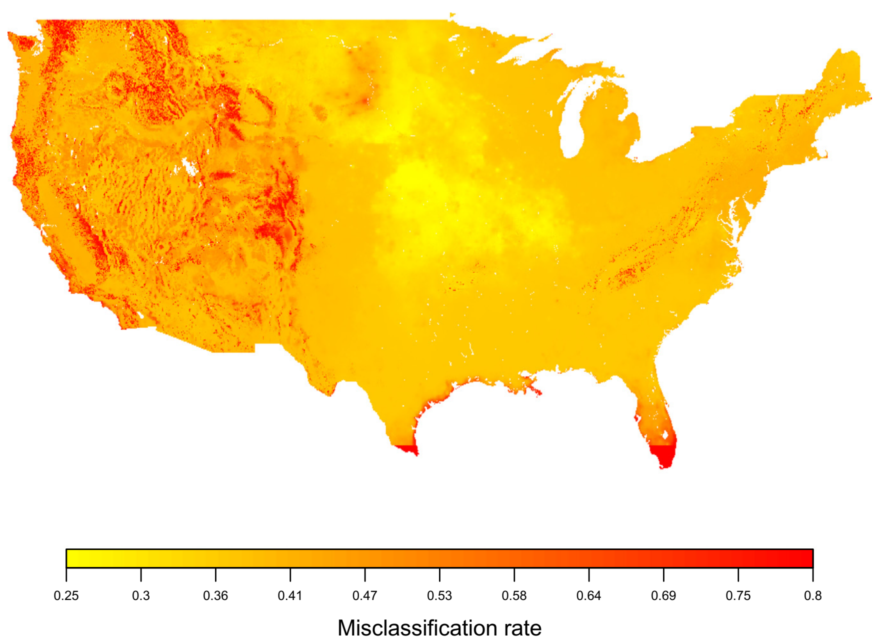

Fig. 5. Misclassification rate for classification example, obtained using the fuzzy k-means with extragrades algorithm.

type of cloud-based platform is data privacy, which is a technical aspect that concerns the scientific community (Tugrul and Polat, 2014). In another aspect, the services that can be used to store data to use from GEE, namely Fusion Tables and Maps Engine, and GEE itself, give the option to keep data and scripts as private, shared (with specific users) or public. It is important to consider that even if the data is kept as private, it is actually located in a server that belongs to a private company, which is a general issue that may emerge from this outsourcing (Pearson and Benameur, 2010), similar to sharing data via a non-institutional email or backing-up information in a cloud-based service. Methods like the one described by Tugrul and Polat (2014) are not implemented in GEE so the researcher should decide if the platform is convenient, or if it is necessary to pre-treat the data with techniques like obscuring (Clifton et al., 2002).

5.1. Geostatistics

To account for complex spatial patterns in data, techniques like Kriging (Burgess and Webster, 1980a,b; Webster and Burgess,

1980) and its derivatives are used heavily in soil and other environmental sciences. They have the capability to make predictions based on the spatial distribution of soil observations, where the final estimate at a location corresponds to a weighted average of known proximal observations. The estimation of these weights is based on the semivariogram, which represents the (semi) variance of a property as a function of the distance between any two observations. Generally, if there is a spatial structure in the data, the variance is low when the distance between two observations is minimum, and increases with the distance, until it reaches a plateau. In the regression example (Section 4.2) the CART algorithm does not fully explain the spatial structure of the data. As a result, the residuals of the model may have a spatial pattern. This is a common outcome because CART and other algorithms are not designed to deal with spatial autocorrelation of residuals. In a conventional DSM routine, if the residuals present a spatial structure, they can be modelled using kriging and added to the model prediction (Odeh et al., 1994), which results in an improved prediction (McBratney et al., 2003).

This technique is not implemented in GEE and there is no straightforward way to program it at the current development stage. Specially over large extents, a local regression kriging technique, which takes the local relationship and spatial structure (Sun et al., 2012), could be a powerful tool for GEE.

5.2. Uncertainty

A general method for uncertainty estimation is the bootstrapping algorithm, originally proposed by Efron and Tibshirani (1993). The algorithm iteratively takes a sample (with replacement) of the training data before training the model. The final prediction corresponds to the mean of the bootstrap iterations and the uncertainty is assigned to a deviation measure of these iterations. While the basic assumption of data independence in bootstrap is not true, we can still use it as an assessment of model uncertainty.

GEE offers a basic implementation of this technique, but so far it is just for classification, and the computational limit (set for security reasons) is rapidly reached when many training features are used. However, GEE provides enough functionality to implement a quasi-automatic bootstrap by randomly selecting points, training the classifiers, and combining the resulting images.

Another suggested method to estimate uncertainty, specially for large extents (Padarian et al., 2014), is the use of an empirical method that utilises the fuzzy k-means with extragrades algorithm (FKM, Tranter et al., 2010). This method classifies the covariate values at the observed points (observations used in model training) in clusters. Each cluster has a central value (centroid), and an associated error. When a new value is predicted, the distances between values of its covariates and the centroids of the clusters are estimated and a membership grade assigned (grade of “belongingness” to each cluster). The final error corresponds to the error weighted by the membership grades.

This technique is not implemented in GEE and there is no straightforward way to program it at the current development stage.

Uncertainty assessment is a critical component of DSM and its implementation in GEE is indispensable for the soil community to use the platform to perform a complete DSM exercise. To illustrate a final DSM product (prediction and associated uncertainty) we used a modification of the FKM algorithm for categorical data, where the uncertainty measure is the misclassification rate of its observations. The uncertainty map (Fig. 5), generated using a desktop workstation, shows how in the central area of the US, dominated by Mollisols, the classification is more successful when compared with more heterogeneous areas on the west coast. It also makes evident that making predictions along more complex landscapes is less satisfactory due to a poor fitting caused in part by the limited number of observations. In consequence, the unsampled areas (e.g. Florida) had a much higher error.

6. Conclusions

Google Earth Engine (GEE) is an interesting new platform which could be implemented in a routine digital soil mapping (DSM) workflow. It is specifically designed to manage large volumes of information in raster format, which are used in DSM to represent soil-forming factors. The major advantage of using GEE is that many rasters are already available, making easier the process of collecting data, and the parallel nature of its algorithms, which considerably accelerates the computation times.

We successfully demonstrated part of two common DSM exercises, namely regression to predict continuous data and classification for categorical data. Both examples had a satisfactory numerical performance, and the computation required to generate the maps (i.e. apply the model to the whole extent) was 40–100 times faster compared with conventional DSM using a desktop workstation.

Despite the reduction in computation time when applying the fitted model to the whole extent, aspects like exploratory analysis are rudimentary. Furthermore, to be able to use GEE as a complete online DSM platform it is necessary to address two important methodological gaps. The first is the need to improve the current implementation to assess the uncertainty of the predictions. The second, and more critical, is related to the use of the spatial information in the data through the implementation of geostatistical techniques.

GEE is in active development, with room for improvement, but not necessarily towards an online soil mapping platform. It is a tool for general environmental mapping that presents big opportunities for the soil and environmental sciences community, especially if we actively participate in the creation of ideas and workflows around the current implementation, to help with the development of a platform that fulfils our specific needs.

Acknowledgements

We thank the reviewers (Titia Mulder and Anonymous) for the insightful comments and suggestions. We also thank David Thau (Google) for his help in the early stages of the manuscript. Finally, we thank Clara da Costa-Reidel for her help in the final stages of the manuscript.

Appendix A. Example code of case study to predict a categorical property

```
import ee
from math import pi
# Service account
MY_SERVICE_ACCOUNT = 'xxxxxxxxxxxxxxxxxxxxx@developer.gserviceaccount.com'
# Private key file path
MY_PRIVATE_KEY_FILE = '~/privatekey.p12'
ee.Initialize(ee.ServiceAccountCredentials(MY_SERVICE_ACCOUNT, MY_PRIVATE_KEY_FILE))
# Import observations from fusion tables
points = ee.FeatureCollection('ft:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
# Import study area from fusion tables
us = ee.FeatureCollection('ft:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')
dem = ee.Image('USGS/NED')
terrain = ee.call('Terrain', ee.Image(dem))
def radians(img):
    return img.toFloat().multiply(pi).divide(180)
slope = radians(terrain.select(['slope']))
winter = ee.ImageCollection('IDAHO_EPSCOR/GRIDMET').filterDate("2009-12-01","2010-02-28")
winter = winter.min()
summer = ee.ImageCollection('IDAHO_EPSCOR/GRIDMET').filterDate("2010-06-01","2010-08-31")
summer = summer.max()
year = ee.ImageCollection('IDAHO_EPSCOR/GRIDMET').filterDate("2010-01-01","2010-12-31")
year_sum = year.sum()
year_mean = year.mean()
# Generate stack of rasters
covariates = ee.Image.cat([dem,slope,winter.select(['tmmn']),summer.select(['tmmx']),\
  year_mean.select(['tmmx','tmmn']),year_sum.select(['pr'])])
covariates = covariates.clip(us)    # Clip stack to study area
# Train classifier
classifier = covariates.trainClassifier(training_features=points,\
  training_property='tax_order',classifier_name='RifleSerialClassifier')
ans = covariates.classify(classifier)    # Apply classifier
```

Appendix B. Confusion matrix for classification exercise

See Table B1.

Table B1
Confusion matrix of categorical classification performed in GEE.

Predicted/observed	Alfisols	Andisols	Aridisols	Entisols	Histosols	Inceptisols	Mollisols	Spodosols	Ultisols	Vertisols
Alfisols	1682	14	25	83	7	120	302	37	107	33
Andisols	7	118	1	9	0	28	31	9	2	0
Aridisols	27	7	779	87	0	20	87	1	0	21
Entisols	80	3	93	611	0	75	165	8	28	18
Histosols	3	0	0	1	9	2	4	2	1	0
Inceptisols	123	31	18	188	3	775	106	38	32	8
Mollisols	254	23	87	93	0	112	2486	9	14	62
Spodosols	27	5	2	27	0	18	4	315	4	1
Ultisols	108	4	1	10	0	47	27	2	436	4
Vertisols	35	1	11	15	0	21	50	0	6	232

References

Abatzoglou, J.T., 2013. Development of gridded surface meteorological data for ecological applications and modelling. Int. J. Climatol. 33 (1), 121–131.
Adhikari, K., Minasny, B., Greve, M.B., Greve, M.H., 2014. Constructing a soil class map of Denmark based on the FAO legend using digital techniques. Geoderma 214, 101–113.
Bishop, T., McBratney, A., Laslett, G., 1999. Modelling soil attribute depth functions with equal-area quadratic smoothing splines. Geoderma 91 (1), 27–45.
Böhning, D., 1992. Multinomial logistic regression algorithm. Ann. Inst. Stat. Math.

44 (1), 197–200.
Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.
Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and Regression Trees. CRC Press, New York.
Burgess, T., Webster, R., 1980a. Optimal interpolation and isarithmic mapping of soil properties: I. The semi-variogram and punctual kriging. J. Soil Sci. 31 (2), 315–331.
Burgess, T., Webster, R., 1980b. Optimal interpolation and isarithmic mapping of soil properties: II. Block kriging. J. Soil Sci. 31 (2), 333–341.
Burillo, D., Chester, M., 2013. Advancing urban infrastructure growth geospatial sustainability assessment. In: International Symposium on Sustainable Systems and Technology.
Clifton, C., Kantarcioglu, M., Vaidya, J., 2002. Defining privacy for data mining. In: National Science Foundation Workshop on Next Generation Data Mining, vol. 1, Citeseer, p. 1.
Coppock, J.T., Rhind, D.W., 1991. The history of GIS. Geogr. Inf. Syst.: Princ. Appl. 1 (1), 21–43.
Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive–aggressive algorithms. J. Mach. Learn. Res. 7, 551–585.
Efron, B., Tibshirani, R.J., 1993. An Introduction to the Bootstrap, vol. 57, CRC Press, New York.
Farahmand, A., AghaKouchak, A., 2013. A satellite-based global landslide model. Nat. Hazards Earth Syst. Sci. 13 (5), 1259–1267.
Gesch, D., Oimoen, M., Greenlee, S., Nelson, C., Steuck, M., Tyler, D., 2002. The national elevation dataset. Photogramm. Eng. Remote Sens. 68 (1), 5–32.
Gesch, D.B., Larson, K., 1996. Techniques for development of global 1-kilometer digital elevation models. In: Proceedings of Pecora Thirteenth Symposium, Sioux Falls, South Dakota.
Giri, C., Long, J., Abbas, S., Murali, R.M., Qamer, F.M., Pengra, B., Thau, D., 2015. Distribution and dynamics of mangrove forests of South Asia. J. Environ. Manag. 148, 101–111.
Giri, C., Ochieng, E., Tieszen, L., Zhu, Z., Singh, A., Loveland, T., Masek, J., Duke, N., 2011. Status and distribution of mangrove forests of the world using earth observation satellite data. Global Ecol. Biogeogr. 20 (1), 154–159.
Hansen, M., Potapov, P., Moore, R., Hancher, M., Turubanova, S., Tyukavina, A., Thau, D., Stehman, S., Goetz, S., Loveland, T., et al., 2013. High-resolution global maps of 21st-century forest cover change. Science 342 (6160), 850–853.
Henderson, B.L., Bui, E.N., Moran, C.J., Simon, D., 2005. Australia-wide predictions of soil properties using decision trees. Geoderma 124 (3), 383–398.
Hengl, T., de Jesus, J.M., MacMillan, R.A., Batjes, N.H., Heuvelink, G.B.M., Ribeiro, E., Samuel-Rosa, A., Kempen, B., Leenaars, J.G.B., Walsh, M.G., Gonzalez, M.R., 2014. SoilGrids1km — global soil information based on automated mapping. PLoS ONE 9 (8), e105992, 08.
Heuvelink, G., Brus, D., De Vries, F., Vasat, R., Walvoort, D., Kempen, B., Knotters, M., 2010. Implications of digital soil mapping for soil information systems. In: 4th Global Workshop on Digital Soil Mapping, Rome, pp. 24–26.
Ibisch, P., 2013. Roadless areas: working towards a comprehensive global assessment. In: The Society for Conservation Biology 26th International Congress.
Jarvis, A., Reuter, H. I., Nelson, A., Guevara, E., 2008. Hole-filled SRTM for the globe, Version 4. Available from the CGIAR-CSI SRTM 90 m Database ⟨http://srtm.csi.cgiar.org⟩.
Landgrebe, T., Merdith, A., Dutkiewicz, A., Müller, R., 2013. Relationships between palaeogeography and opal occurrence in Australia: a data-mining approach. Comput. Geosci. 56, 76–82.
Lehner, B., Verdin, K., Jarvis, A., 2008. New global hydrography derived from spaceborne elevation data. EOS, Trans. Am. Geophys. Union 89 (10), 93–94.
Littlestone, N., 1988. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. Mach. Learn. 2 (4), 285–318.
Maji, S., Berg, A.C., Malik, J., 2008. Classification using intersection kernel support vector machines is efficient. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, IEEE, pp. 1–8.
McBratney, A., Mendonça Santos, M.L., Minasny, B., 2003. On digital soil mapping. Geoderma 117 (1), 3–52.
Odeh, I., McBratney, A., Chittleborough, D., 1994. Spatial prediction of soil properties from landform attributes derived from a digital elevation model. Geoderma 63 (3), 197–214.
Padarian, J., Minasny, B., McBratney, A., 2014. The evolving methodology for global soil mapping, GlobalSoilMap: Basis of the Global Spatial Soil Information System. Taylor & Francis Group, London, pp. 215–220.
Pearson, S., Benameur, A., 2010. Privacy, security and trust issues arising from cloud computing. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, pp. 693–702.
Rish, I., 2001. An empirical study of the naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, pp. 41–46.
Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65 (6), 386.
Shalev-Shwartz, S., Singer, Y., Nathan, S., Cotter, A., 2011. Pegasos: primal estimated sub-gradient solver for svm. Math. Programm. 127 (1), 3–30.
Sun, W., Minasny, B., McBratney, A., 2012. Analysis and prediction of soil properties using local regression-kriging. Geoderma 171, 16–23.
Tranter, G., Minasny, B., McBratney, A., 2010. Estimating Pedotransfer function prediction limits using fuzzy k-means with extragrades. Soil Sci. Soc. Am. J. 74 (6), 1967–1975.
Tugrul, B., Polat, H., 2014. Privacy-preserving kriging interpolation on partitioned data. Knowl.-Based Syst. 62, 38–46.
USDA–NRCS, 2014. National soil information system. Online: ⟨http://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/?cid=nrcs142p2_053552⟩, 01 August 2014.
Webster, R., Burgess, T., 1980. Optimal interpolation and isarithmic mapping of soil properties: III. Changing drift and universal kriging. J. Soil Sci. 31 (3), 505–524.
Wilson, A.M., Parmentier, B., Jetz, W., 2014. Systematic land cover bias in Collection 5 MODIS cloud mask and derived products—a global overview. Remote Sens. Environ. 141, 149–154.
Xia, Y., Mitchell, K., Ek, M., Cosgrove, B., Sheffield, J., Luo, L., Alonge, C., Wei, H., Meng, J., Livneh, B., et al., 2012a. Continental-scale water and energy flux analysis and validation for North American Land Data Assimilation System project phase 2 (NLDAS-2): 2. Validation of model-simulated streamflow. J. Geophys. Res.: Atmos. (1984–2012) 117 (D3).
Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., et al., 2012b. Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products. J. Geophys. Res.: Atmos. (1984–2012) 117 (D3).