Corresponding Author

John S Hogland
Rocky Mountain Research Station, USDA Forest Service
200 E. Broadway, Missoula, MT 59807
jshogland@fs.fed.us

Improved analyses using function datasets and statistical modeling

Hogland, John S.[1] and Anderson, Nathaniel M.[1]

[1] US Forest Service, Rocky Mountain Research Station, Missoula, MT

Abstract

Raster modeling is an integral component of spatial analysis. However, conventional raster modeling techniques can require a substantial amount of processing time and storage space and have limited statistical functionality and machine learning algorithms. To address this issue, we developed a new modeling framework using C# and ArcObjects and integrated that framework with .Net numeric libraries. Our new framework streamlines raster modeling and facilitates predictive modeling and statistical inference.

Introduction

Raster modeling is an integral component of spatial analysis and remote sensing. Combined with classical statistic and machine learning algorithms, it has been used to address wide ranging questions in a broad array of disciplines (e.g. Patenaude et al. 2005; Reynolds-Hogland et al. 2006). However, the current workflow used to integrate statistical and machine learning algorithms and process raster models within a geographic information system (GIS) limits the types of analyses that can be performed. While some analysts have successfully navigated these limitations, they typically have done so for relatively small datasets using fairly complex procedures that combine GIS procedures with statistical software outside a GIS. This process can be generally described in a series of steps: 1) build a sample dataset using a GIS, 2) import that sample dataset into statistical software such as SAS (SAS 2014) or R (R 2014), 3) define a relationship (predictive model) between response and explanatory variables that can be used within a GIS to create predictive surfaces, and 4) build a representative spatial model within a GIS that uses the outputs from the predictive model to create spatially explicit surfaces. Often the multi-software complexity of this practice warrants building tools that streamline and automate many aspects of the process, especially the export and import steps. However, a number of challenges pose significant limitations to producing final outputs in this way, including learning additional software, implementing predictive model outputs, managing large datasets, and handling the long processing time and large storage space requirements.

Out of necessity, some analysts have learned additional software and have re-sampled data to work around these problems, performed intensive statistical analyses, and created final outputs in a GIS. While learning additional software can open the doors to many different algorithms and resampling data can make data more manageable, learning statistical software comes at a significant cost in resources (time) and re-sampling can blur the relationships between response and explanatory variables. In a recent study (Hogland et. al. 2014), we were faced with these tradeoffs, examined the existing ways in which raster models are processed, and developed new ways to integrate complex statistical and machine learning algorithms directly into ESRI’s desktop application programming interface (API).

After reviewing conventional raster modeling techniques, it was clear that the way raster models process data could be improved. Specifically, raster models are composed of multiple spatial operations. Each operation reads data from a given raster dataset, transforms the data, and then creates a new raster dataset (Figure 1). While this process is intuitive, creating and reading new raster datasets at each step of a model comes at a high processing and storage price, prompting two questions: 1) does generating a final model output require the creation of intermediate outputs at each step and 2) if we remove intermediate outputs from the modeling process, what improvements in processing and storage can we expect?

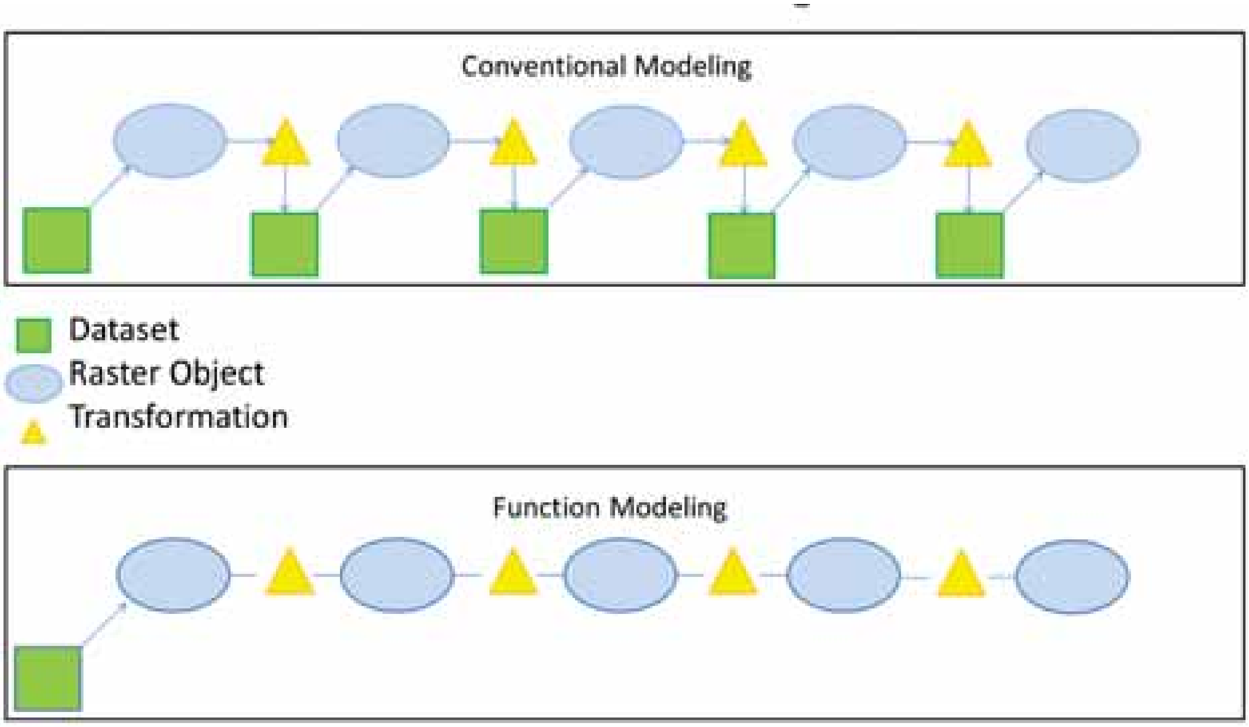


Figure 1. This figure illustrates the conceptual idea behind Function Modeling. For each raster transformation (yellow triangles) within a conventional model (CM), an intermediate raster dataset is created (green square). Intermediate raster datasets are then read into a raster object (blue ovals) and transformed to produce another intermediate raster dataset. Function Modeling (FM) does not create intermediate raster datasets and only stores the type of transformation occurring to source raster datasets. When raster data are read, this technique performs the transformations to the source datasets without creating intermediate raster datasets.

Moreover, while ESRI's libraries provide a vast set of procedures to manipulate data, compared to the programming environment used by many statisticians and mathematical modelers, these procedures are relatively limited to some of the more basic statistical and machine learning algorithms. In addition, the algorithms that are available in standard libraries are not directly accessible within ESRI classes because they are wrapped into a class requiring an input that produces a predefined output. While this kind of encapsulation does simplify coding, it limits the use of the algorithms to the designed class workflow. Take for example the principal component analysis (PCA) class within Spatial Analyst namespace (ESRIa, 2010, ESRIb 2010). The interface to this class requires three simple inputs (raster bands, path to estimated parameters, and optional number of components to calculate) and produces two outputs (a new raster surface and the estimated parameters). Within the class, the PCA algorithm combines a number of statistical steps to generate the PCA estimates (e.g., calculating variance co-variance and performing the matrix algebra to estimate Eigen values and vectors). These estimates are then used to transform the original bands into an orthogonal raster dataset. Though simple to code and implement for raster data, the algorithms used to estimate Eigen values and vectors within the class cannot be applied to other workflows that use different types of data (e.g., vector data). These types of limitations significantly restrict the types of analyses that can be performed within ArcDesktop and necessitate the development and integration of additional libraries. In this study we address the limitations of raster spatial modeling and complex statistical and machine learning algorithms within the ArcDesktop environment by extending ArcMap's functionality and creating a new modeling framework called Function Modeling (FM).

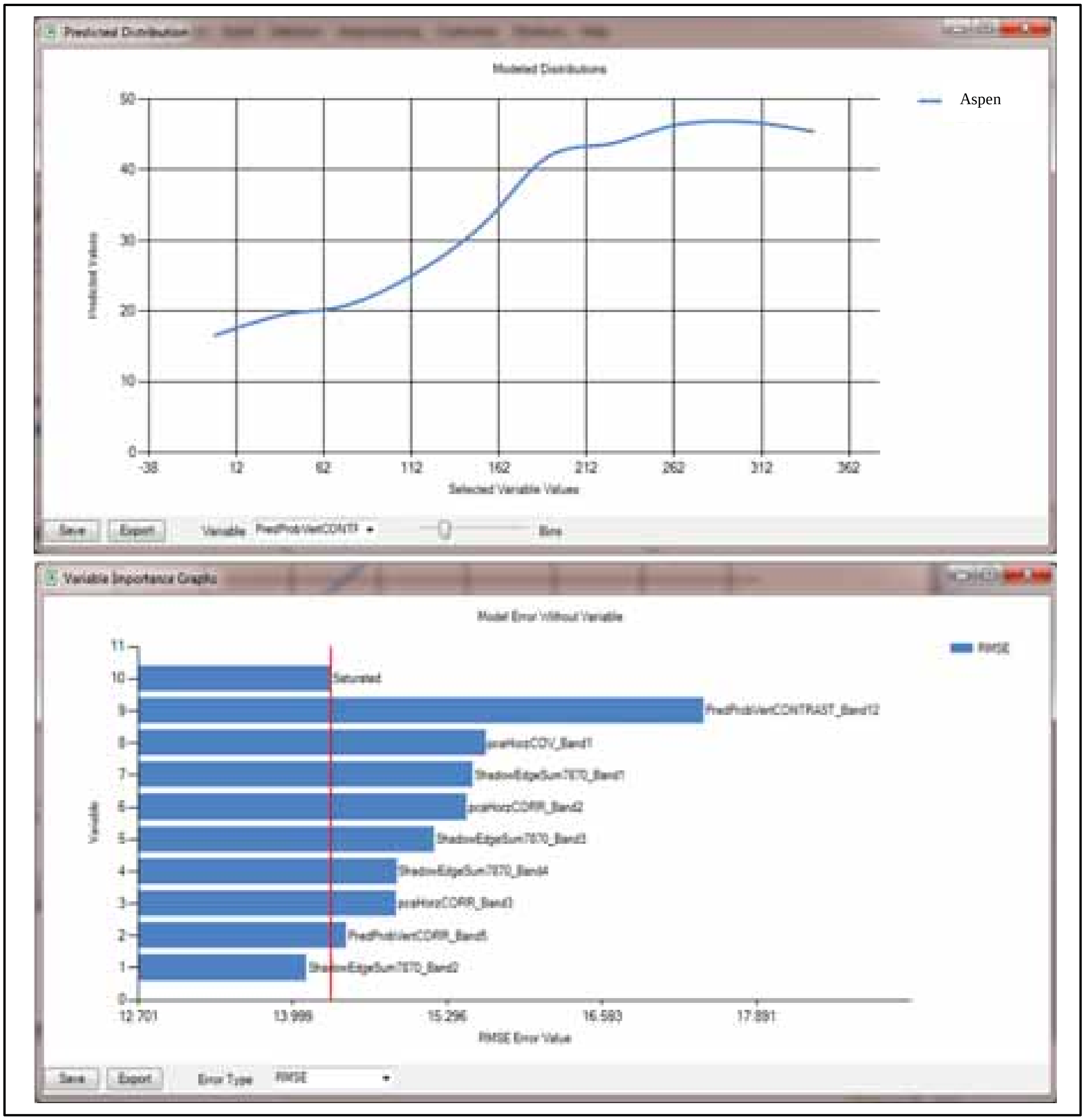
Methods

.NET Libraries

To address these challenges we developed a series of coding libraries that leverage the concepts of delayed reading using Function Raster Datasets (ESRIc 2010) and integrate numeric, statistical, and machine learning algorithms with ESRI's ArcObjects. Combined, these libraries facilitate FM which allows users to chain modeling steps and complex algorithms into one raster dataset or field calculation without writing the outputs of intermediate modeling steps to disk. Our libraries were built using an object-oriented design, .NET framework, ESRI's ArcObjects, ALGLIB (Sergey 2009), and Accord.net (Souza 2012). This work is easily accessible to coders and analysts through our subversion site (FS Software Center 2012) and ESRI toolbar add-in (RMRS 2012).

The methods and procedures of our class libraries parallel many of the functions found in ESRI's Spatial Analyst extension including focal, aggregation, resampling, convolution, remap, local, arithmetic, mathematical, logical, zonal, surface, and conditional. However, our libraries support multiband manipulations and perform these transformations without writing intermediate or final output raster datasets to disk. Our spatial modeling framework focuses on storing only the manipulations occurring to datasets and applying those transformations dynamically at the time data are read, greatly reducing processing time and data storage requirements. In addition to including common types of raster transformations, we have developed and integrated multiband manipulation procedures such as gray level co-occurrence matrix (GLCM), landscape metrics, entropy and angular second moment calculations for focal and zonal procedures, image normalization, and a wide variety of statistical and machine learning transformations directly into this modeling framework.

While the statistical and machine learning transformations can be used to build surfaces and calculate records within a field, they do not in themselves define the relationships between response and explanatory variables like a predictive model. To define these relationships, we built a suite of classes that performs a wide variety of statistical testing and predictive modeling using many of the optimization algorithms and mathematical procedures found within the ALGLIB (Sergey 2009) and Accord.net (Souza 2012) projects. Typically, these classes use samples drawn from a population of tabular records or raster cells to test hypotheses and create generalized associations (e.g., an equation or procedure) between variables of interest that are expensive to collect (i.e. response variables) and variables that are less costly and thought to be related to the response (i.e. explanatory variables; Figure 2). While the inner workings and assumptions of the algorithms used to develop these relationships are beyond the scope of this paper, the classes developed to utilize these algorithms within a spatial context provide coders and analysts with the ability to easily use these techniques to define relationships and answer a wide variety of questions. Combined with FM, the sampling, equation\procedure building, and predictive surface creation workflow can be streamlined to produce outputs that not only answer questions, but also display relationships between response and explanatory variables in a spatially explicit manner at fine resolutions across large spatial extents.



Variable PredProbVertCONTF

Error Type RMSE

Figure 2. An example of a generalized association (predictive model) between tons of above ground Aspen biomass (response) and nine explanatory variables derived from remotely sensed imagery. This relationship was developed using a random forest algorithm and the RMRS Raster Utility coding libraries. The top graphic illustrates the functional relationship between biomass and the explanatory variable called "PredProbVert CONTRAST_Band12", while holding all other explanatory variables values at 20% of their sampled range. The bottom graphic depicts the importance of each variable within the predictive model by systematically removing one variable from the model and comparing changes in root mean squared error (RMSE).

Simulations

From a theoretical standpoint, FM should reduce processing time and storage space associated with spatial modeling (i.e., less in and out reading and writing from and to disk). However, to justify the use of FM methods, it is necessary to quantify the extent to which FM actually reduces processing and storage space. To evaluate the efficiency gains associated with FM, we designed, ran, and recorded processing time and storage space associated with six simulations, varying the size of the raster datasets used in each simulation. From the results of those simulations we compared and contrasted FM with conventional modeling (CM) techniques using linear regression. All CM techniques were directly coded against ESRI's ArcObject Spatial Analyst classes to minimize CM processing time and provide a fair comparison.

Spatial modeling scenarios within each simulation ranged from one arithmetic operation to twelve operations that included arithmetic, logical, conditional, focal, and local type analyses (Table 1). Each modeling scenario created a final raster output and was run against six raster datasets ranging in size from 1,000,000 to 121,000,000 total cells, incrementally increasing in size by 2000 columns and 2000 rows at each step. Cell bit depth remained constant as floating type numbers across all scenarios and simulations.

Table 1. Spatial operations used to compare and contrast Function Modeling and conventional raster modeling. Superscript values indicate the number of times an operation was used within a given model. Model number (column 1) also indicates the total number of processes performed for a given model.

Model	Spatial Operation Types
1	Arithmetic (+)
2	Arithmetic (+) & Arithmetic (∗)
3	Arithmetic (+), Arithmetic (∗) & Logical (>=)
4	Arithmetic (+)^2, Arithmetic (∗) & Logical (>=)
5	Arithmetic (+)^2, Arithmetic (∗), Logical (>=) & Focal (Mean,7,7)
6	Arithmetic (+)^2, Arithmetic (∗)^2, Logical (>=) & Focal (Mean,7,7)
7	Arithmetic (+)^2, Arithmetic (∗)^2, Logical (>=), Focal (Mean,7,7), & Conditional
8	Arithmetic (+)^3, Arithmetic (∗)^2, Logical (>=), Focal (Mean,7,7), & Conditional
9	Arithmetic (+)^3, Arithmetic (∗)^2, Logical (>=), Focal (Mean,7,7), Conditional, & Convolution (Sum,5,5)
10	Arithmetic (+)^3, Arithmetic (∗)^3, Logical (>=), Focal (Mean,7,7), Conditional, & Convolution(Sum,5,5)
11	Arithmetic (+)^3, Arithmetic (∗)^3, Logical (>=), Focal (Mean,7,7), Conditional, Convolution(Sum,5,5), & Local(∑)
12	Arithmetic (+)^4, Arithmetic (∗)^3, Logical (>=), Focal (Mean,7,7), Conditional, Convolution(Sum,5,5), & Local(∑)

Case study

Furthermore, to illustrate the benefits of using FM to analyze data and create predictive surfaces, we evaluated the time savings associated with a recent study in the Uncompahgre National Forest (UNF; Hogland et. al., 2014). In this study we used FM, field data, and fine resolution imagery to develop a two stage classification and estimation procedure that predicts mean values of forest characteristics across 0.5 million acres at a spatial resolution of 1 m^2. The base data used to create these predictions consisted of National Agricultural Imagery Program (NAIP) color infrared (CIR) imagery (U.S. Department of Agriculture National Agriculture Imagery Program 2012), which contained a total of ten billion pixels for the extent of the study area. Within this project we created no less than 365 explanatory raster surfaces (many of which represented a combination of multiple raster functions), 52 predictive models, and 64 predictive surfaces at the extent and spatial resolution of the NAIP imagery[1].

While it would be ideal to directly compare CM with FM for the UNF case study, ESRI's libraries do not have procedures to perform many of the analyses and raster transformations used in the study, so CM was not incorporated directly into the analysis. Alternatively, to evaluate the time savings associated with using FM we used our simulated results and estimated the number of individual CM functions required to create GLCM explanatory variables and model outputs. Processing time was based

[1] While this study initially used spatial analyst procedures (PCA and ISO cluster) and statistical SAS® software to build predictive models, our libraries have since added these techniques, which in turn have been used to generate the final outputs for this study.

on the number of CM functions required to produce explanatory variables and predictive surfaces. In many instances, multiple CM functions were required to create just one explanatory variable or final output. The number of functions were then multiplied by the corresponding proportion increase in processing time associated with CM when compared to FM. Storage spaced was calculated based on the number of intermediate steps used to create explanatory variables and final outputs. In all cases the space associated with intermediate outputs from CM were calculated without data compression and summed to produce a final storage space estimate.

Results

.Net Libraries

Leveraging ESRI's API, we were able to substantially reduce development time and focus our resources on designing solutions to analytical problems. Currently, our solution contains 639 files, over 70,000 lines of code, 457 classes, and 74 forms that facilitate functionality ranging from data acquisition to raster manipulation and statistical/machine learning types of analyses. Our coding solution, RMRS Raster Utilities, contains two primary projects that consist of an ESRI Desktop class library ("esriUtil") and an ArcMap add-in ("servicesToolBar"). The project esriUtil contains most of the functionality of our solution, while the servicesToolBar project wraps that functionality into a toolbar accessible within ArcMap. In addition, two open source numeric class library solutions are used within our esriUtil project. These solutions, Accord.net (Souza 2012) and ALGLIB (Sergey 2009), provide access to numeric, statistical, and machine learning algorithms used to facilitate the statistical and machine learning procedures within our classes.

Our esriUtil project represents the bulk of our coding endeavor and the classes within this project are grouped based on functionality into General Procedures, Data Acquisition, Raster Utilities, Sampling, and Statistical Procedures. General Procedures include classes that facilitate FM, batch processing, opening and closing function models, managing models, and saving FM to conventional raster datasets (e.g., Imagine file format). Data Acquisition classes streamline the use of web services to interactively download and store vector and raster data. Raster Utilities classes streamline raster transformations and summarizations and builds the underlining foundation for FM. Sampling classes focus on creating samples of populations and extracting values from raster datasets for those samples. Statistical Procedures classes summarize datasets and analyze variables within those datasets to test hypotheses and build predictive models.

The primary classes within esriUtil include a database manipulation class (geodatabaseutily), raster manipulation class (rasterUtil), feature manipulation class (featureUtil), and multiple statistical and machine learning classes (Statistics). The geodatabaseutility class contains 2,831 lines of code that streamlines vector, tabular, and raster data access and provides procedures to create vector and tabular data along with fields within vector and tabular datasets. The rasterUtil class contains 4,258 lines of code and 124 public procedures that focus on dynamically transforming raster datasets using raster functions. Many of these procedures reference 56 newly developed raster functions that create transformations of raster datasets. Raster transformations for these newly developed raster functions range from surface type manipulations to generating predictive model outputs (Table 2) and can be added to any function raster dataset to produce a FM. The featureUtil class contains 815 lines of code and simplifies selecting features based on the statistical properties of the data and exporting features to a new output. Our Statistics namespace contain 88 different classes with over 12,000 lines of code that performs a wide variety of analyses ranging from categorical data analysis to creating predictive models such as principle component analysis (PCA), clustering, general linear modeling (GLM), Random Forest ®, and Soft-max neural nets (Table 3).

Table 2. Raster transformations that can be applied to function raster datasets using the raterUtil class of the esriUtil project.

Raster Function	Button	Description
Arithmetic Analysis	+ × − ÷	Adds, subtracts, multiplies, divides, and calculate modulus for two multiband raster datasets or a multiband raster dataset and a constant
Math Transformation		*Mathematically transform a multiband raster dataset
Logical Analysis		*Performs >, <, >=, <=, = logical comparisons between to multiband raster datasets or a multiband raster dataset and a constant
Conditional Analysis	⊕	*Performs a conditional analysis using multiband raster datasets and/or constants
Remap Raster		Remaps the cell values of a multiband raster dataset
Focal Analysis		* Performs focal analyses for a multiband raster dataset
Focal Sample		* Samples raster cells of a multiband raster dataset within a focal window, given defined offsets from the central window cell
Convolution Analysis		* Performs convolution analysis on multiband raster dataset
Local Stat Analysis	Σ Π	* Performs local type analysis on a multiband raster dataset
Linear Transform		* Performs a linear transformation of a multiband raster dataset
GLCM		* Performs grey level co-occurrence matrix transformations on multiband raster datasets
Landscape Metrics		* Performs moving window landscape metrics transformations for multiband raster datasets
Create Constant Raster	🟩	* Creates a constant value raster dataset
Null Data to value	N	* Converts no data to a new value for multiband raster dataset
Set Values to No Data	Null	* Sets specified values to no data for multiband raster dataset
Set Null Value	Set Null	* Sets the values used for no data in a raster dataset
Flip Raster		* Flips multiband raster datasets
Shift Raster		* Shifts multiband raster datasets based on a specified number of pixels
Rotate Raster		* Rotates multiband raster datasets based on a specified angle
Combine Raster		* Combines multiband raster dataset to create a single band raster dataset of unique raster band combinations
Aggregate Cells		* Resamples and perform a summarization of all cells used to increase the cell size
Slope Raster		Calculates slope
Calculate Aspect		Calculates aspect
Northing	↑↑	* Converts aspect to a northing raster dataset
Easting	→ →	* Converts aspect to a easting raster dataset
Clip Analysis		Clips a multiband raster dataset given either the extent of a raster dataset or the boundary of a feature class
Convert Pixel Type		Transforms pixel depth for multiband raster datasets
Rescale Analysis		* Rescales pixel values of a multiband raster dataset to the min and max values given the defined pixel depth
Create Composite		* Creates a multiband raster dataset given specified bands from multiple raster datasets
Extract Bands		* Creates a multiband raster dataset by extract specified bands of a multiband raster dataset
Tasseled Cap		* Performs a Tasseled Cap transformation (Landsat 7 coefficients)
NDVI		* Calculates a normalized difference vegetation index (NDVI)
Resample Raster		* Resamples the cell size of a multiband raster
Normalize		* Brings two multiband raster datasets to a common radiometric scale
Merge Raster		* Merges multiband raster datasets
Create Mosaic		Creates a mosaic raster dataset
Function Modeling	f(x)	* Combines raster function into a function model

Zonal Statistics		* Performs zonal statistics on multiband raster datasets
Region Group		* Groups regions of a multiband raster dataset
Build Raster Model		* Builds predictive raster surfaces given a predictive model and explanatory raster surfaces
Batch Processing	GO	* Will batch process RMRS Raster Utility commands

[*] Identifies either new procedures or functionality not available within ESRI's Spatial Analyst extension or Image Analysis Window.

The servicesToolBar combines the forms and procedures within esriUtil into an organized, easily deployed toolbar. While primarily a wrapper around the functionality contained within esriUtil, servicesToolBar provides analysts with easy access to a wide array of procedures that currently do not exist within ArcMap. Moreover, many of these procedures can be tied together in a step-like fashion to build multi-procedure FM and batch processes. Together, these projects facilitate a wide range of analytics procedures that can be applied to both vector and raster type data in a streamlined, efficient fashion.

Table 3. Statistical analyses and machine learning techniques accessible to coders and analysts through the RMRS Raster Utility libraries. All statistical and machine learning procedures create a RMRS Raster Utility model that can be used to display modeling results and predict new outcomes in raster, vector, and tabular formats.

Analysis	Button	Description
Variance Covariance	COV	Calculates variance covariance and correlation matrices for fields within a table or bands within a raster
Stratified Variance Covariance	SCOV	Calculates stratified variance covariance and correlation matrices for fields within a table or bands within a raster
Compare Sample to Population		Compares similar fields within two tables to determine if a sample is representative of a population
Compare Classifications		Compares the accuracy and similarity of two categorical classifications
Accuracy Assessment		Calculates the accuracy of a classification model
Adjusted Accuracy Assessment		Adjusts an existing accuracy assessment by weighting accuracy based on the proportions of each category within a user defined extent
T-Test	TT	Compares fields within a table or bands within a raster by groups or zone, respective
PCA/SVD		Performs a single vector decomposition, principal component analysis on fields within a table or bands within a raster dataset
Cluster Analysis		Performs K-means, Gaussian, and binary clustering on fields within a table or bands within a raster
Linear Regression	REG	Performs multiple linear regression using fields within a table or bands within a raster
Logistic Regression		Performs logistic regression using fields within a table or bands within a raster
Generalized Linear Modeling	GLM	Performs general linear modeling using fields within a table or bands within a raster (supported links: identity, absolute, cauchit, inverse, inverse squared, logit, log, log log, probit, sin, and threshold)
Multivariate Regression	MVR	Performs multivariate linear regression using fields within a table or bands within a raster
Multinomial Logistic Regression	PLR	Performs polytomous (multinomial) logistic regression using fields within a table or bands within a raster
Random Forest		Performs random forest classification or regression tree analysis using fields within a table or bands within a raster
Soft-max Nnet		Performs soft-max neural networks classification using fields within a table or bands within a raster
Format Zonal Data	ZF	Formats the outputs of multiband Zonal Statistics (Table 2) for use in statistical machine learning techniques
View Sample Size	SS	Estimates sample size based on a subset of the above statistical procedures using power analysis
Show Model Report		Shows the outputs of the above statistical and machine learning procedures
Predict New Data	f(x)	Creates new fields within a tabular dataset and populates the records within that field with the predicted outcomes based on the above statistical and machine learning procedures
Delete Model	☒	Deletes statistical and machine learning models create in the above procedures

Simulations

To evaluate the gains in efficacy of FM, we compared it with CM across six simulations. FM significantly reduced both processing time and storage space when compared to CM (Figures 3 and 4). To perform all six simulations FM took 26.13 minutes and a total of 13,299.09 megabytes while CM took 73.15 minutes and 85,098.27 megabytes. Theoretically, processing time and storage space associated with creating raster data for a given model, computer configuration, set of operations, and data type should be a linear function of the total number of cells within a rater dataset and the total number of operations within the model:

$$\text{Time}_i \text{ (seconds)} = \beta_i(\text{cells x operations})_i$$
$$\text{Space}_i \text{ (megabytes)} = \beta_i(\text{cells})_i$$

where i denotes the technique, either FM or CM in this case. Our empirically derived FM Time and Space models nicely fit the linear paradigm (R^2=0.96 and 0.99, respectively). However, for the CM techniques the model for Time (R^2=0.91) deviated slightly and the model for Space (R^2=0.10) deviated significantly from theoretical distributions, indicating that in addition to the modeling operations, extra operations occurred to facilitate raster processing and that some of those operations produced intermediate raster datasets, generating wider variation in time and space requirements than would be expected based on these equations.

FM and CM time equation coefficients indicate that, on average, FM reduced processing time by approximately 278%. Model fit for FM and CM space equations varied greatly between techniques and could not be compared reliably in the same manner. For the CM methodology, storage space appeared to vary depending on the types of operations within the model and often decreased well after the creation of the final dataset. This outcome is assumed to be caused by delayed removal of intermediate datasets. Overall though, FM reduced total storage space by 640% when compared to CM for the six simulations.

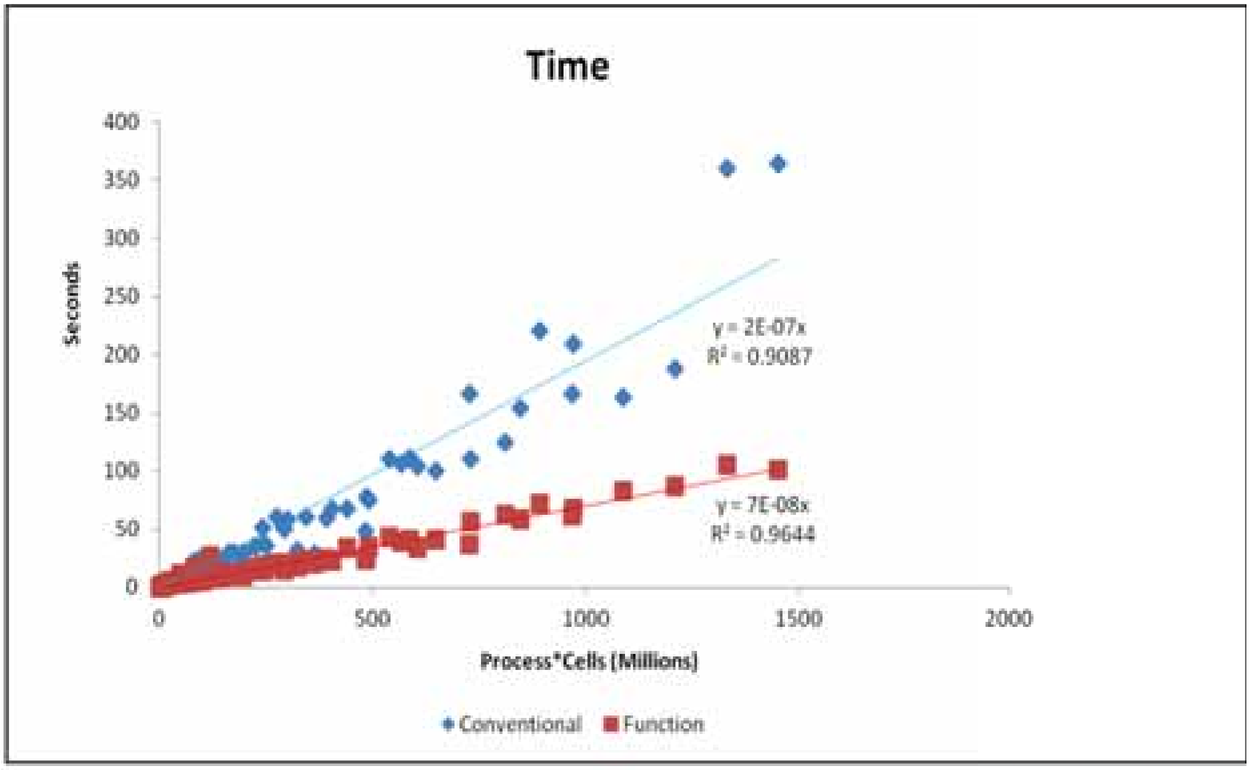


Figure 3. Function Modeling and conventional modeling regression results for processing time.

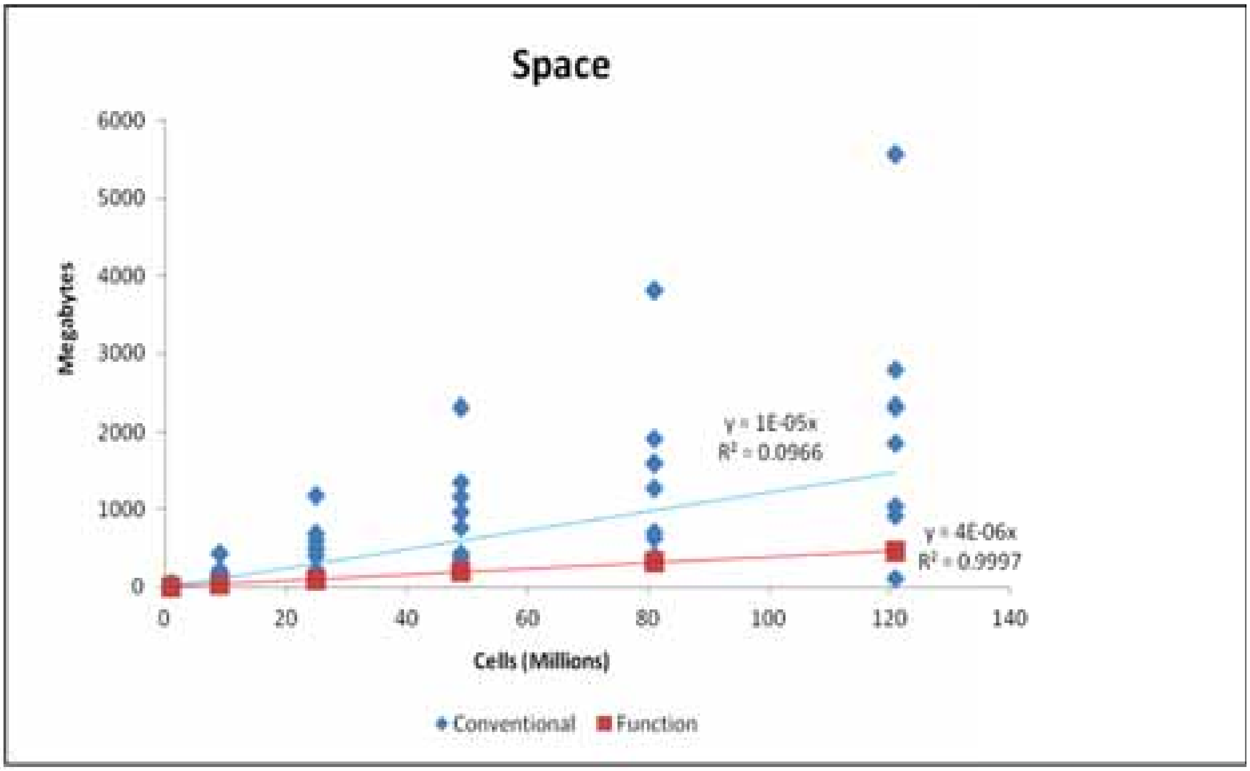


Figure 4. Function Modeling and conventional modeling regression results for storage space.

Case study

Although simulated results helped to quantify improvements related to using FM, the UNF case study illustrates these saving in a more concrete and applied manner. As described in the methods section, this project created no less than 365 explanatory raster surfaces, 52 predictive models, and 64 predictive surfaces at the extent and spatial resolution of the NAIP imagery (202,343 ha and 1 m^2, respectively). In this comparison, it was assumed that all potential explanatory surfaces were created for CM to facilitate sampling and statistical model building. In contrast, FM only sampled explanatory surfaces and calculated values for the variables selected in the final models. For the classification stage of the UNF study, we estimated that to reproduce the explanatory variables and probabilistic outputs using CM techniques, it would take approximately 283 processing hours on a Hewlett-Packard EliteBook 8740 using an I7 Intel quad core processor and would require at least 19 terabytes of storage space. Using FM these same procedures took approximately 37 processing hours and total of 140 gigabytes of storage space. Similarly, for the second stage of the classification and estimation approach using CM, we estimated a processing time of approximately 2,247 hours with an associated storage requirement of 152 terabytes. Using FM these same procedures took approximately 488 hours and required roughly 913 gigabytes of storage.

Discussion

ArcObjects provides a flexible platform that can be used to extend the functionality of ESRI software. Many have leveraged this flexibility to develop procedures, models, scripts, libraries, add-ins, extensions, and applications that automate, streamline, and create new outputs and functionality. Moreover, this flexibility makes it easy to integrate the functionality of other coding solutions. In our solution, ESRI's API provided the base functionality needed to build new modeling techniques that significantly improve raster processing and efficiently add complex statistical and machine learning algorithms directly into ArcGIS. Our solution, RMRS Raster Utility, not only adds numerous statistical and machine learning procedures directly to Desktop, but also facilitates quick and efficient manipulation and modeling of large datasets. For example, the NAIP imagery used in the case study included 10 billion pixels covering 0.5 million acres at a spatial resolution of 1 m^2. These procedures can be used to perform basic transformations of raster datasets that can be combined together through FM to perform new analyses that are simply too costly in terms of time and space to perform using CM.

The underlying concept behind FM is delayed reading. Using this concept with raster data removes the need to store intermediate raster datasets to disk in the spatial modeling process and allows models to process raster data only for the spatial areas needed at run time. As spatial models become more complex (i.e., more modeling steps), the number of intermediate outputs increases. From a CM standpoint these intermediate outputs significantly increase the amount of processing time and storage space required to run a given model. In contrast to CM, FM does not store intermediate datasets, thereby significantly reducing model processing time and storage space. In this study, we confirmed this effect by comparing FM to CM methodologies using six simulations that created final raster outputs from multiple, incrementally complex models. Furthermore, FM does not need to store outputs to disk in a conventional format (e.g., Grid or Imagine format). Instead, all functions occurring to source raster datasets can be stored and used in a dynamic fashion to display, visualize, symbolize, retrieve, and manipulate function datasets at a small fraction of the time and space it takes to create a final raster output. For example, if we were to perform the same six simulations presented here but replace the requirement of storing the final output as a conventional raster format with creating a FM, FM would take less than 1 second and would require less than 50 kilobytes of storage to finish the same simulations. This is a dramatic change in the cost associated with these types of analyses.

Similar to the simulations, FM significantly reduced processing time and storage space associated with creating the final outputs of the UNF case study. In this example, FM facilitated an analysis that simply would have been too costly to perform using CM. Moreover, FM made it possible to explore a wide range of potential relationships in a streamlined fashion. By using the concept of delayed reading and only processing and extracting data for a sample of the study area rather than its full extent, FM made it possible to explore many more relationships than could have been evaluated using CM. For example, to populate the same values within a CM context, analysts would either develop and apply spatial models for each of the potential explanatory surfaces (365) at the spatial extent of the entire study area or would need to extract the extent of each sample within the baseline raster dataset, perform transformations for each of those extractions, and then populate sample from each of those transformed extractions. Here again, populating the sample using CM techniques would simply be too costly in terms of processing and storage. However, populating the samples used in the UNF study using FM required only a few minutes and did not require any addition storage beyond that required for an additional field and its populated values. This example illustrates that FM not only improves raster spatial modeling but opens the door to projects that are currently out of reach for many analysts using only CM.

To facilitate the use of FM and complex statistical and machine learning algorithms within ESRI’s GIS, we have created a user-friendly ESRI toolbar (Figure 4) that provides quick access to a wide array of spatial analyses, while allowing users to easily store and retrieve FM models. FM models can be loaded into ArcMap as a function raster dataset and be used interchangeably with all ESRI raster spatial operations. While FM is extremely efficient and uses almost no storage space, there may be circumstances that warrant saving a FM to disk as a conventional raster format. For example, if one FM model (FM1) is read multiple times to create a new FM (FM2) and the associated time it takes to read multiple instances of FM1 is greater than the combined read and write time of saving FM1 to a conventional raster format (i.e., many transformations within the first FM), then it would be advantageous to save the FM1 to a conventional raster format and use the conventional raster dataset in FM2. Nonetheless, results from our simulations and case study demonstrate that raster modeling does not require storing intermediate datasets to disk and FM substantially reduces processing time and storage space.

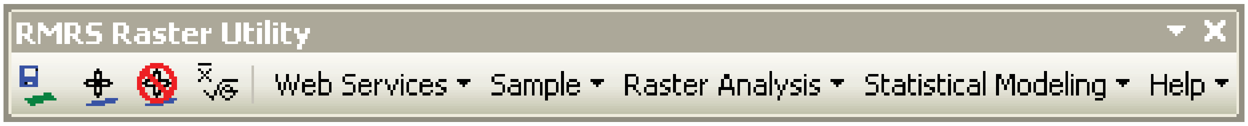


Figure 4. RMRS Raster Utility toolbar is a free, public source, object oriented library packaged as an ESRI add-in that simplifies data acquisition, raster sampling, and statistical and spatial modeling, while reducing the processing time and storage space associated with raster analysis.

Moving forward, it is a significant challenge to maintain the compatibility of these coding projects with advances in software. For example, while ESRI's libraries allowed us to streamline our coding solution and focus our efforts on new processing and analytical techniques, they also require a substantial investment in code maintenance. Specifically, as ESRI releases new versions of their software, coding projects that rely on fundamental components of ESRI libraries (like the projects described here) must test and potentially modify classes to account for those changes. For the RMRS Raster Utility solution, migrating between Desktop 10.0 and 10.1 required a fundamental change in how raster objects were being used within our classes. Although this change was relatively minor in terms of concept, it impacted almost every class within our solution, in turn requiring substantial time and resources to make FM usable within Desktop 10.1.

Conclusion

FM streamlines raster modeling by removing the need to store intermediate raster datasets to disk. Combined with complex statistical and machine learning algorithms, FM can be used to manipulate raster information in ways that were once too costly to entertain. Moreover, by integrating numeric libraries directly into our solution, users can answer questions, build predictive models, and display predictive outputs in a spatially explicit manner all within the familiar ArcGIS environment.

Acknowledgments

This project was supported by the Rocky Mountain Research Station and Agriculture and Food Research Initiative, Biomass Research and Development Initiative, Competitive Grant no. 2010-05325 from the USDA National Institute of Food and Agriculture.

References

ESRI. (2010a). ArcGIS Desktop Help 10.0 – Rasters with functions. Available online at http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Rasters_with_functions/009t0000000m000000; last accessed 6/3/2014.

ESRI. (2010b). ArcObjects SDK 10 Microsoft .Net Framework – ArcObjects Library Reference (Spatial Analyst). Available online at http://help.arcgis.com/en/sdk/10.0/arcobjects_net/componenthelp/index.html#/PrincipalComponents_Method/00400000010q000000/; Last accessed 6/3/2014.

ESRI. (2010c). ArcGIS Desktop Help 10.0 – How principal components works. Available online at http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/How_Principal_Components_works/009z000000qm000000; last accessed 6/3/2014.

FS Software Center. (2012). RMRS Raster Utility. Available online at https://collab.firelab.org/software/projects/rmrsraster; last accessed 6/3/2014.

Hogland J, Anderson N, Wells L, and Chung W. (2014). Estimating forest characteristics using NAIP imagery and ArcObjects, Proceeding of the 2014 ESRI International Users Conference, San Diego, CA.

Patenaude G, Milne R, and Dawson T. (2005). Synthesis of remote sensing approaches for forest carbon estimation Reporting to the Kyoto Protocol. *Environmental Science and Policy* 8(2): 161–178.

R (2014). The R project for statistical computing, Accessed online: http://www.r-project.org/, last accessed 5/14/2014.

Reynolds-Hogland M, Mitchell M, and Powell R, (2006). Spatio-temporal availability of soft mast in clearcuts in the Southern Appalachians. *Forest Ecology and Management* 237(1–3): 103-114.

RMRS. (2012). RMRS Raster Utility. Available online at http://www.fs.fed.us/rm/raster-utility; last accessed 6/3/2014.

SAS (2014). SAS, The Power to Know, Accessed online: http://www.sas.com/en_us/home.html?gclid=CI7Jpfja8r4CFQpefgodfiIAWw, last accessed 5/14/2014.

Sergey B. (2009). ALGLIB, Available online: http://www.alglib.net/, last accessed 6/3/2014.

Souza C. (2012). Accord.Net Framework, Available online: http://accord-framework.net/, last accessed 6/3/2014.

U.S. Department of Agriculture National Agriculture Imagery Program (2012). National Agriculture Imagery Program (NAIP) Information sheet. Accessed online: http://www.fsa.usda.gov/Internet/FSA_File/naip_info_sheet_2013.pdf, last accessed 5/14/14.