Introduction to ScalaTion

John A. Miller
Department of Computer Science
University of Georgia

May 12, 2018

1 Introduction

Coded in Scala, the ScalaTion framework provides support for large-scale analytics and simulation [3]. ScalaTion provides extensive support for analytics, optimization and simulation and currently consists of four major modules: `scalation_mathstat`, `scalation_database`, `scalation_modeling` and `scalation_models`. A fifth module named `scalation_automod` intended to partially automate the model development process [5, 6] is still being refined. In addition to automation, provision for incorporating theory [4] is being added to this module. Furthermore, the related Scalation Kernel project provides a ScalaTion kernel for Jupyter notebooks.

2 The `scalation_mathstat` Module

The `scalation_mathstat` module in ScalaTion provides the infrastructure for analytics, optimization and simulation. Keys parts of this module are the `linalgebra` and `tenalgebra` packages. The `linalgebra` package provides support for operations on vectors (dense, sparse and compressed [1]) and matrices (dense, sparse, compressed, symmetric tridiagonal and bidiagonal). It also supports common matrix factorization techniques, including LU, Cholesky, QR and SVD with application to regression. Eigenvalue decomposition is also supported. Many of the algorithms also have parallel versions in the `par` subpackage that utilize multithreading. Work is currently underway to provide support for SIMD vector instructions as well. Currently, this is done by using `cblis` through JNI. Once the JVM supports this, direct support will be provided.

3 The `scalation_database` Module

ScalaTion provides a high-performance columnar, main-memory analytics database that supports straightforward conversion to and from vectors, matrices and 3-level hypermatrices (similar to tensors). The analytics database has built-in temporal types and operators, to which is being added spatial types and operators as well as capabilities of working with images and videos. The temporal capabilities are being currently used to support time-series analysis, e.g., as discussed in the following paper [7]. It compares a variety of data science/machine learning techniques for forecasting vehicle traffic, including SARIMA, Exponential Smoothing, Dynamic Regression and Neural Networks, all of which are provided by ScalaTion.

4 The `scalation_modeling` Module

As discussed in [2], ScalaTion is inteded to provide a modeling continuum from predictive analytics to simulation modeling. These capabilities are centered in the `scalation_modeling` module.

4.1 Analytics

The `analytics` package provides comprehensive support for analytics including linear models ($y = \boldsymbol{\beta} \cdot \mathbf{x} + \epsilon$) and generalized linear models ($\mathbb{E}[y] = g^{-1}(\boldsymbol{\beta} \cdot \mathbf{x})$ where g is the inverse link function). Both ℓ_2 (e.g. Ridge) and ℓ_1 (Lasso) regularizations are supported. Dimensionality reduction is provided by principle component analysis. Nonlinear models are also included, primarily Nonlinear Regression and Neural Networks. Support for multi-layer Neural Networks and Recurrent Neural Networks is provided and work is underway to add Long Short-Term Memory Neural Networks.

There are several subpackages, including the following: The `classifier` subpackage handles cases where y is a categorical variable. It includes Bayesian classifiers [8], decision trees, random forests, Gaussian mixture models, linear discriminant analysis, logistic regression and support vector machines. The `clusterer` subpackage provides k-Means, k-Mean++, hierarchical, Markov and tight clustering. The `fda` or Functional Data Analysis subpackage may be used when the data come from a continuous process. It includes functional smoothing, functional regression, functional principle component analysis and functional clustering. The `forecaster` package is used for making predictions into the future. Consider for a moment a simple linear regression model $y = \beta_0 + \beta_1 x$. Many individual samples are collected $\{(x_i, y_i)\}$ and are treated as independent. For forecasting, the samples are indexed by time and treated as dependent. The model corresponding to simple linear regression would be (Auto-Regressive, Integrated, Moving Average with eXplanatory variable) ARIMAX. If the explanatory variable x is dropped, the model is ARIMA where future values of y are predicted from prior values of y. The package also includes Seasonal ARIMA (SARIMA), exponential smoothing and Kalman filters. The `recommender` package provides several ways for estimating missing values in a matrix, based on similar rows and columns (e.g., rows may be viewed as users and columns as products). Also, predictions can be made based on using the first/largest few singular values (with their corresponding singular vectors) from a Singular Value Decomposition (SVD) of the matrix.

4.2 State Models

ScalaTion support for state models can be used for both simulation and analytics and includes Markov Chains, Continuous-Time Markov Chains, Markov Decision Processes, Partially Observable Markov Decision Processes and Hidden Markov Models. Some of these modeling techniques are still under development.

4.3 Optimization

Estimation of parameter/weight vectors or matrices, which involves minimizing prediction errors (Least Squares) or maximizing the probability of a model generating the sample data (Maximum Likelihood Estimation) boils down to an optimization problem. In simple cases, a gradient may be set to zero to produce a system of equations (e.g., $X^T X \boldsymbol{\beta} = X^T \mathbf{y}$) that may be solved using matrix factorization. Generally, first order methods or gradient based optimizers are used to determine parameter/weight values, including gradient descent, stochastic gradient descent, gradient descent with momentum and conjugate gradient). An alternative that can reduce the number of iterations at the cost of more work per iteration is second order methods, typical ones the that only approximate the Hessian (notably BFGS and L-BFGS). In addition, special techniques such as iteratively reweighted least squares and alternating direction method of multipliers can also be useful. ScalaTion provides almost all of these and work is on-going to provide all of them. Some work has begun on parallel versions of the algorithms.

5 The `scalation_models` Module

The `scalation_models` module provides several example analytics and simulation models [2] that serve as exemplars for developing models using ScalaTion. Each modeling technique also includes simple models as `Test` objects (e.g., `RegressionTest`) within the same source file.

References

[1] Vishnu Gowda Harish, Vinay Kumar Bingi, and John A Miller. A big data platform integrating compressed linear algebra with columnar databases. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 2344–2352. IEEE, 2016.

[2] John A Miller, Michael E Cotterell, and Stephen J Buckley. Supporting a modeling continuum in scalation: from predictive analytics to simulation modeling. In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pages 1191–1202. IEEE Press, 2013.

[3] John A Miller, Jun Han, and Maria Hybinette. Using domain specific language for modeling and simulation: Scalation as a case study. In *Proceedings of the Winter Simulation Conference*, pages 741–752. Winter Simulation Conference, 2010.

[4] John A Miller, Hao Peng, and Michael E Cotterell. Adding support for theory in open science big data. In *Big Data (BigData Congress), 2017 IEEE International Congress on*, pages 251–255. IEEE, 2017.

[5] Mustafa V Nural, Michael E Cotterell, Hao Peng, Rui Xie, Ping Ma, and John A Miller. Automated predictive big data analytics using ontology based semantics. *International journal of big data*, 2(2):43, 2015.

[6] Mustafa V Nural, Hao Peng, and John A Miller. Using meta-learning for model type selection in predictive big data analytics. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 2027–2036. IEEE, 2017.

[7] Hao Peng, Santosh U Bobade, Michael E Cotterell, and John A Miller. Forecasting traffic flow: Short term, long term, and when it rains. In *Big Data (BigData Congress), 2018 International Congress on*, pages –. Services Society, 2018.

[8] Hao Peng, Zhe Jin, and John A Miller. Bayesian networks with structural restrictions: Parallelization, performance, and efficient cross-validation. In *Big Data (BigData Congress), 2017 IEEE International Congress on*, pages 7–14. IEEE, 2017.