Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

COMPARISON OF GAUSSIAN PROCESS MODELING SOFTWARE

Collin Erickson
Bruce E. Ankenman

Department of Industrial Engineering
and Management Sciences
Northwestern University
2145 Sheridan Road
Evanston, IL 60208, USA

Susan M. Sanchez

Operations Research Department
Naval Postgraduate School
1 University Circle
Monterey, California 93943, USA

ABSTRACT

Gaussian process fitting, or kriging, is often used to create a model from a set of data. Many available software packages do this, but we show that very different results can be obtained from different packages even when using the same data and model. Seven different fitting packages that run on four different platforms are compared using various data functions and data sets that reveal there are stark differences between the packages. In addition to comparing the prediction accuracy, the predictive variance—which is important for evaluating precision of predictions and is often used in stopping criteria—is also evaluated.

1 INTRODUCTION

Gaussian process modeling, or kriging, is commonly used to fit a model to data. Gaussian processes, or GPs, are popular because they are flexible models that interpolate well and also provide an estimate of the standard error of predictions (see, e.g., Kleijnen 2015 or Santner, Williams, and Notz 2003). The standard GP model assumes that any subset of points in the sample space follows a multivariate normal distribution whose covariance is determined by a correlation function. There are many choices for the correlation, but we only use the Gaussian correlation, which is the most popular in both statistical and simulation metamodeling settings.

In our previous experiments we found that two different software implementations for kriging gave significantly different results on the same data set. This happens because, despite using the same equations, the optimization routines vary by the implementation. Such a discrepancy would not occur for a simpler model, such as a linear model, which does not require sophisticated optimization. In this project we analyze various packages available for GP modeling, and compare the results. The most important part of modeling is that the predictions at new points are accurate. However, it is also important that the predicted standard errors for predictions are accurate, since these are often used either to determine the model quality or as a stopping criteria for sequential sampling schemes.

2 METHODOLOGY

We compare seven different software implementations for kriging: the R packages GPfit (MacDonald, Ranjan, and Chipman 2015), laGP (Gramacy 2015), and mlegp (Dancik and Dorman 2008); the Python modules GPy (The GPy authors 2015) and scikit-learn (Pedregosa et al. 2011); the MATLAB toolbox DACE (Lophaven, Nielsen, and Søndergaard 2002); and JMP (SAS 2016). We use each of these to fit data for ten test problems, including data from six different Gaussian process scenarios and four test functions.

The primary measure to test the efficacy of a software package is the accuracy of the predictions, which we evaluate with the empirical model root mean squared error (ERMSE). This is calculated by checking

the differences between the fitted model predictions of 2000 points across the surface and the actual values of the test function at each of those same points. We also evaluate the predicted model RMSE. Since the predicted RMSE is an estimate prediction error, the predicted RMSE at each of the 2000 prediction points should be approximately equal to the ERMSE described above.

3 RESULTS

First, we use data taken from an actual Gaussian Process. Since the data sets match the assumptions of the model, we expect that the models should do reasonably well and give similar results. We find that all of the packages give similar results on most of these data sets, although there are some exceptions. The fits are much better than the linear model when the design is dense enough, such as having fifty points in two dimensions. But when the design is relatively sparse, such as having 300 points in six dimensions on a rough surface, the models tend to do little better than a linear model. Thus we find that, as is generally the case, more complex models require more data in order to get improvements.

We also test the fits using data sampled from four functions, including the commonly used borehole function. On these data sets we see more differences between the various packages. JMP has numerical issues on some replicates that yield bad results, especially in the predicted RMSE. The R package mlegp also has some poor results where it gives predicted RMSE values that are far worse than others. The R package laGP is often a little worse than the other programs, but is generally good. DACE, GPy, scikit-learn, and GPfit are all generally good options, but there can still be significant differences between them even when fitting the same data. While the predicted RMSE over the whole surface is often fairly close to the ERMSE, at times they differ by a factor of two or more in either direction, showing that care must be taken when using the predicted RMSE to make decisions about the data.

Surprisingly, the fitting run times from package to package can vary as much as three orders of magnitude. In general, GPfit, mlegp, and JMP were the slowest and could take hours to fit a single model; while the others were much faster and could fit a model in under a minute in all of our tests.

4 CONCLUSIONS

Simulation researchers and practitioners may benefit from a better understanding of the relative merits of GP fitted models obtained from a variety of available packages. This study focuses only on the simple GP model, but there are many improvements on the model that should be used when more information is known about the data or the data set is large. Advanced models will have even more variability in predictions, so practitioners should take steps to make sure the model matches their expectations.

REFERENCES

Dancik, G. M., and K. S. Dorman. 2008. "mlegp: statistical analysis for computer models of biological systems using R". *Bioinformatics* 24 (17): 1966–1967.

Gramacy, R. B. 2015. *laGP: Local Approximate Gaussian Process Regression*. R package version 1.2-1.

Kleijnen, J. P. 2015. *Design and Analysis of Simulation Experiments*, Volume 230. Springer.

Lophaven, S. N., H. B. Nielsen, and J. Søndergaard. 2002. "DACE-A Matlab Kriging toolbox, version 2.0". Technical report.

MacDonald, B., P. Ranjan, and H. Chipman. 2015. "GPfit: An R Package for Fitting a Gaussian Process Model to Deterministic Simulator Outputs". *Journal of Statistical Software* 64 (12): 1–23.

Pedregosa, F. et al. 2011. "Scikit-learn: Machine learning in Python". *The Journal of Machine Learning Research* 12:2825–2830.

Santner, T. J., B. J. Williams, and W. Notz. 2003. *The Design and Analysis of Computer Experiments*. Springer Science & Business Media.

SAS 2016. "JMP: Gaussian Process".

The GPy authors 2012–2015. *GPy: A Gaussian process framework in python*.