Understanding the mechanical stability of wellbores using machine learning

CS229 course project: final report
Team members enrolled in class: Reza Rastak (rastak@stanford.edu)
Other team members: Nicolò Spiezia (nicolospiezia@gmail.com)

Abstract

We try to understand the various components affecting the stability of tunneling and wellbore operations. We use a series of advanced finite element (FE) simulations to obtain a dataset which contains the stress response of the soil during drilling operations as a function of 12 input parameters. Using Multivariate Adaptive Regression Splines (MARS), we create a simplified model of the simulations (meta-model) and test its accuracy. Using the created meta-model, we perform Sobol sensitivity analysis and study the most important variables and interactions. Additionally, we explain how this approach is helpful for inverting the model to optimize the input variables.

Introduction

Horizontal drilling operations are of the most challenging engineering processes. These include wellbores for extracting oil using hydraulic fracturing, mining operations, and creating tunnels for transformation infrastructures (subways, road tunnels). Usually a drill is used to cut a cylindrical hole and move the soils and debris out of the tunnel. All the steps for a tunneling project must be examined and calculated before any operations. For example, in order for the hole to not collapse under the weight of the upper soil, an outward pressure is applied to the inside of the tunnel, called the mud pressure. Calculating the most efficient value for the mud pressure is difficult since it depends of the local geometry and material properties of the soil around the tunnel.

We have access to the proprietary FE simulation code called **tunnel-simulator**. The code simulates the drilling process and gives us accurate distribution of mechanical stress in the soil. However, performing the simulation is relatively slow. In many applications such as performing optimization on the input parameters, inverting the equation and solving for one of the inputs, and performing Monte-Carlo-based sensitivity analysis, we need to run this simulation a large number of times and it becomes very costly.

In this project, we use the code to create a training set and a dev set. We run the code for a list of random input parameters. Then train our MARS model on the training set by choosing reasonable hyper-parameters. Having the simplified model gives us incredible efficiency to predict the outputs from the model for thousands of various input parameters. We study the efficiency of the model and then show some of the benefits of having a simple model (meta-model) for the complicated simulation black-box.

Related work

Physical simulations are becoming more and more complicated and require huge computational resources to run them. New advances in the field of machine learning allowed many research groups across the world to incorporate machine learning on large datasets of simulated data. From these data, they extract simple relations, get a deeper perspective on the physical problem, and make predictions in a fraction of the time needed to run a simulation. A review of the most widely used techniques for generating meta-models are presented in [1-2].

There are some ideas for similar use of machine learning in the field of computational mechanics. For example, [3] uses neural networks to capture non-linear constitutive models in solid mechanics. Classification algorithm are used to help the construction process of tunnels [4]. Various uses of NN in geotechnical engineering are explained in [5].

Datasets and Features:

We ran simulations for 500 different input parameters. Since we need a good uniform distribution of input parameters, we make use of the Latin Hypercube sampling algorithm [6] (using pyDOE python package). Out of 500 total simulations, 10 simulations crashed. Thus, using k-fold cross-validation with 5 folds, we have (490/5=98) test (dev) data and 392 training data. All components of X and Y data values are normalized by the average of that component in the all the data. Input parameters describe the detailed properties of the tunnel and the drilling process. These inputs include:

- **Geometry (3 features):** Radius of the tunnel, the horizontal and vertical inclination angle of the tunnel.
- **In-situ conditions (4 features):** Vertical stress, two horizontal stress coefficients in two directions, and the mud-pressure used in the drilling operations.
- **Material properties of the soil (5 features):** Elastic modulus (E), Poisson's ratio (ν), cohesion (c0), friction angle (ϕ), dilatation angle (ϕ')

The ranges for these inputs are shown in Table 1.

From the output data file, we extracted 3 output variables for each simulation.

1. Y0: The maximum vertical (Z axis) stress
2. Y1: The maximum horizontal (X axis) stress
3. Y2: The maximum Mises stress

These outputs describe how the soil responds to the drilling process. For example, if the Mises stress is too high, it shows that the tunnel may collapse after drilling. Additionally, if horizontal or vertical stress exceeds the in-situ stress, it shows that the soil has fractures due to high mud pressure.

Methods

As we mentioned in the previous section, we used Latin hypercube sampling for generating the set of input parameters. Using a script, we converted these input parameters to a set of input files for the program. The FE software first creates the geometry of the system, then creates the FE mesh (see Fig.1). Later, applied the loading and boundary conditions and runs the FE simulations. It then creates detailed post-processed vtk files that show the displacement of the nodes and the stresses within each element. We wrote a separate script to compute the mises stress and to compute the maximum value of these stresses.

After creating the complete database, we use Multivariate Adaptive Regression Splines (MARS) [7] to obtain a simplified model. It creates a rich set of features containing a constant, linear terms, hinge terms. Hinge functions are in the form $\max(0, x - c)$ and $\max(0, c - x)$. The linear combination of these hinge terms gives us piecewise linear interpolations. These features are added one-by-one selectively using a greedy algorithm. A new term can be created by multiplying a new feature with an existing feature. This can create higher order polynomial terms and interactions. After reaching almost perfect training accuracy, the algorithm smartly removes features from the regression expression one-by-one and determines which features must be removed (pruned) in order to avoid overfitting. All the training process is done by the pyEarth package. As previously mentioned, we ran 5-fold cross validation to determine the best hyper parameters for our model and study the accuracy of the model. The hyper parameters include the maximum polynomial degree, and the penalty coefficient for the pruning step. The accuracy is measured by computing the average squared error for each data point (MSE).

After training the model, we performed a monte-carlo-based sensitivity analysis and computed the Sobol indices for the input parameters of the model. This analysis is performed using the SALib python library.

Results

Table 2 shows the effect of hyper parameters on the accuracy of the model. For the good models, a learning curve is plotted in Fig. 2(left). The models are fairly accurate. We then performed the Sensitivity analysis and the results are summarized in Table 3. The soil cohesion is the most important factor in the simulation followed by the mud pressure.

After successfully fitting the data, we plotted both the FE simulation result and the MARS predicted results as a function of the mud pressure, keeping all other variables constant. Note that mud pressure is the only parameter that can be adjusted during the drilling process. All other inputs are fixed for a given site. This allows us to find a good range of mud pressure values that correspond to acceptable levels of stress. The meta-model allows us to compute this curve 1000 times faster. The curves (Fig. 2 right) show that with increasing number of training samples, we can match the accurate curve more and more closely. Note that the best hyper-parameters are chosen for these plots.

Conclusion

This project shows that simplified models can be an efficient and relatively accurate way to simplify complex numerical simulations, especially in the case of tunneling operations. This simplified model, which can give us answers more than 1000 times faster than the FE simulation, enables us to run big monte-carlo simulations, analyze the sensitivity of the model, or perform optimizations. Furthermore, we show that MARS algorithm is a very effective method of generating accurate meta-models. It can handle complex models and it can easily prevent overfitting.

The parameter name	Minimum	Maximum
Radius (m)	0.05	0.15
Vertical inclination angle (deg)	0.0	10.0
Horizontal inclination angle (deg)	0.0	10.0
Mud pressure (Pa)	5.0E6	25.0E6
Vertical in-situ stress (Pa)	20.0E6	44.0E6
Horizontal stress coefficient M1	0.2	0.4
Horizontal stress coefficient M2	0.2	0.4
Elastic modulus of soil (Pa)	800.0E6	1600.0E6
Poisson's ratio	0.1	0.3
Soil cohesion (Pa)	4.5E6	12.5E6
Soil friction angle (deg)	27	57
Soil dilatation angle (deg)	27	57

Table 1: The range of input data used to create the dataset. We used SI units.

Output variable	Maximum polynomial degree	Penalty coefficient	MSE training set	MSE test (dev) set	Performance
Max. stress Z	1	3	0.014	0.020	High bias
	2	3	0.004	0.022	High variance
	2	40	0.015	0.019	Good
Max stress X	1	3	0.011	0.016	High bias
	2	3	0.003	0.011	Good
Max Mises stress	1	3	0.021	0.032	High bias
	2	3	0.008	0.137	Very high variance
	2	40	0.026	0.027	Acceptable

Table 2: The hyper parameter analysys

Paramter name	Cohesion	Mud Pressure	Friction angle(ϕ)	Radius
Sobol sensitivity index S1	0.441	0.193	0.044	0.024
Sobol total sensitivity index ST	0.507	0.357	0.120	0.092

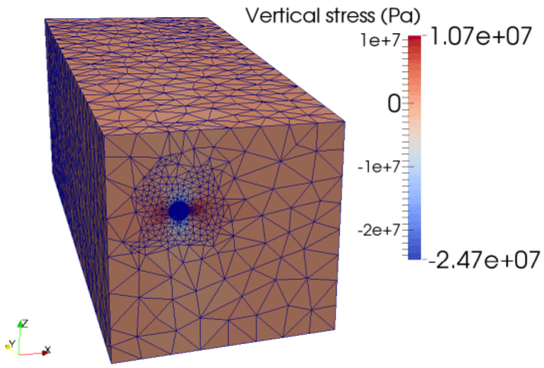
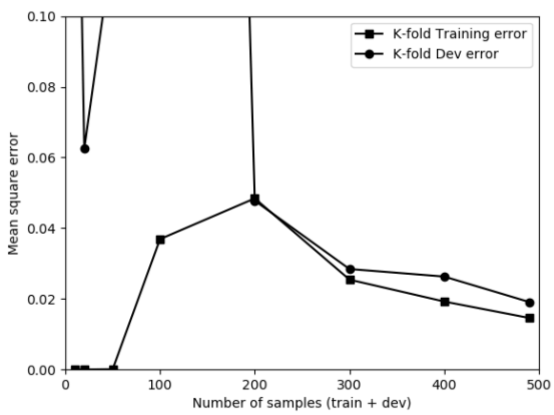
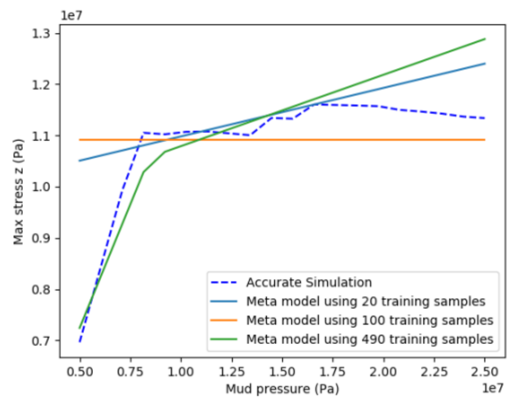
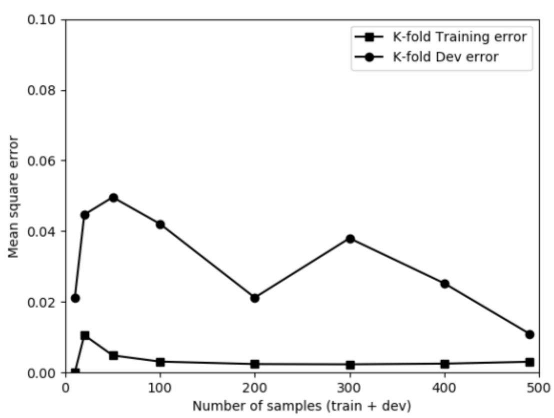
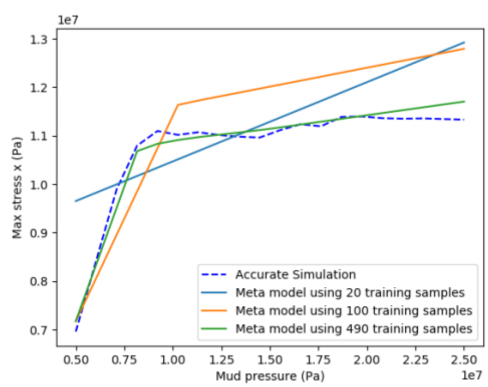


Figure 1: The contour of the vertical stress caused by drilling the tunnel shown on top of the finite element mesh









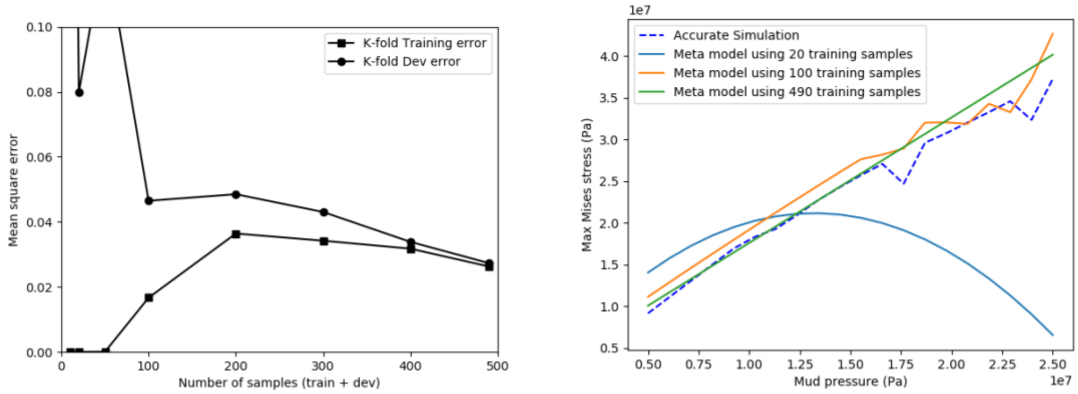


Figure 2: (left) the learning curve for each output variable {y0, y1, y2}. (right) The comparison between accurate simulation and simplified formulas for all output parameters as a function of the mud pressure.

Contributions

Reza Rastak: wrote all scripts for create the input dataset and modifying the input files of the FE software. Also wrote the code for extracting the output parameter, performing the learning algorithm, running k-fold cross-validation, Sobol parameter study and mud-pressure parameter study. Also wrote all the reports and presented the poster.

Nicolò Spiezia (not enrolled in class): provided the FE simulation code and taught Reza how to effectively use it. Gave valuable advice about the mechanical problem and the tunneling industry.

References

1. Storlie, C. B. et al. (2009), Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models, *Reliability Engineering and System Safety*, 94, 1735-1763
2. C. Zopf, M. Kaliske (2017), Numerical characterisation of uncured elastomers by a neural network based approach, *Comp Struct*, 182, 504-525
3. Hashash, Y. M. A., Jung, S. and Ghaboussi, J. (2004), Numerical implementation of a neural network based material model in finite element analysis. *Int. J. Numer. Meth. Engng.,* 59: 989–1005.
4. A.N. Jiang, S.Y. Wang, S.L. Tang (2011), Feedback analysis of tunnel construction using a hybrid arithmetic based on Support Vector Machine and Particle Swarm Optimisation, *Automation in Construction*, 20(4), 482-489
5. Sarat Kumar Das, chapter 10 (2013), Artificial Neural Networks in Geotechnical Engineering: Modeling and Application Issues, In Metaheuristics in Water, Geotechnical and Transport Engineering, edited by Xin-She Yang, Amir Hossein Gandomi, Siamak Talatahari and Amir Hossein Alavi, Elsevier, Oxford, 2013, 231-270
6. Tang, B. (1993), Orthogonal array-based Latin Hypercubes. *Journal of the American Statistical Association*. 88 (424): 1392-1397
7. Friedman, J. (1991). Multivariate adaptive regression splines. The annals of statistics, 19(1), 1–67.