CS 391D

Project Proposal

Jeho Oh (jo23927)
Oleg Zilman (oz572)

1. Problem Statement

Highly Configurable Systems (HCSs) are systems with parameters so that users can configure the functionality and running environment of the system. An active research topic in HCSs is learning an accurate performance model. A performance model aims to predict the performance of a configuration based on its parameters. A performance model is valuable to understand the properties of an HCS and find configurations that have optimal performance.

Prior work utilized different techniques including multivariate and LASSO regression, decision trees, and deep neural networks [1, 2, 3, 4]. However, they were evaluated only on small HCSs (<54 parameters). The authors of a recent paper [5] tried a variety of different machine learning techniques on a Linux kernel (>15000 parameters), which yielded lower accuracy than their predecessors' [1,2,3,4] findings. Applicability of prior work [1,2,3,4] on larger HCSs is unknown, while [5] mentioned that some known machine learning approaches are not applicable for the scale of Linux.

We aim to evaluate some of the techniques presented in [1,2,3,4,5], along with other machine learning algorithms of our choice, on larger HCSs (90 ~ 988 parameters). One thing we are particularly interested in is to use smaller training sets and see if we can achieve good results for cases where training data is limited. In addition, we aim to analyze the results and gather an understanding on why some methods work better than others. Furthermore, we will try to improve the accuracy of our modeling by applying different dimensionality reduction techniques, such as Principal Component Analysis (PCA).

2. Proposed approach

Our tentative approach, as we envision it today, has three phases:

1) **Replication**: replicate prior work that used the Random Forest algorithm, multivariate regression and a feed forward deep neural network, on both small HCSs and large HCSs.
2) **Analysis**: analyze the effect of changing the number of parameters and the number of samples on the performance of the different algorithms and try to explain the results based on the mathematical concepts covered in class.
3) **Dimensionality reduction**: apply various methods to reduce the parameter space which includes: removing parameters that are fixed or have low variance, PCA, or any other methods that we will find suitable.

3. Data to Use

We have build-size measurements of 1000 configurations for each of the following HCSs:

- axTLS: a server framework with 94 parameters
- Toybox: Linux command line utility with 316 parameters
- Fiasco: real time microkernel with 234 parameters

- Busybox: executable UNIX common utilities with 998 parameters
- uClibc-ng: library for embedded Linux with 269 parameters

We can generate more samples for each of the five HCSs above and might do so if we need more training or test data. In addition, we can also use the dataset from [5], which measured the build-size of 95,854 Linux Kernel configurations. We aim to start with the smaller systems mentioned above, and if time permits, work on the Linux Kernel dataset.

4. Timeline

Our tentative development plan is presented below:

March 13th – 27th	- Setup code infrastructure and obtain all the data - Implement the algorithms we are interested in evaluating: a) random forest, b) multivariate regression c) deep neural network. If we have time left, we would add GB tree and CART.
March 27th – April 10th	- Run our algorithms on the training sets from the 5 HCSs that we have the data for, experimenting with different sample sizes (Re-using Linux kernel data from [5] is optional) - Initial analysis of the results
April 10th	First Draft submission
April 10th – 24th	- Continue Analyses of results - Review and implement possible methods for dimensionality reduction.
April 24th – Semester End	- Gather and analyze results on dimensionality reduction - Final report - Other plans TBD

5. References

[1] Siegmund, N., Grebhahn, A., Apel, S., & Kästner, C. (2015). Performance-Influence Models for Highly Configurable Systems. Proceeding of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2015), 284–294.

[2] Ha, H., & Zhang, H. (2019). Performance-Influence Model for Highly Configurable Software with Fourier Learning and Lasso Regression. 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), 470–480.

[3] Guo, J., Yang, D., Siegmund, N., Apel, S., Sarkar, A., Valov, P., … Yu, H. (2018). Data-efficient performance learning for configurable systems. Empirical Software Engineering, 23(3), 1826–1867.

[4] Ha, H., & Zhang, H. (2019). DeepPerf: Performance Prediction for Configurable Software with Deep Sparse Neural Network. Proceedings - International Conference on Software Engineering, 2019-May(July)

[5] Mathieu Acher, Hugo Martin, Juliana Pereira, Arnaud Blouin, Jean-Marc Jézéquel, Learning Very Large Configuration Spaces: What Matters for Linux Kernel Sizes. [Research Report] Inria Rennes - Bretagne Atlantique. 2019.