**Abstract**

Title: Machine Learning Approaches for Preliminary Screening of Parameter Sets in Ensemble Models

Project Members: Jacob Martin (contact: jacobmartin2022@u.northwestern.edu)

Course: EECS 349: Machine Learning, at Northwestern University

Synopsis of Work: The goal of my project is to accelerate a model of bacterial metabolism, which is the subject of my Ph.D. research. The model is implemented as a set of ODE-based chemical reaction rate equations, for which many possible sets of parameters – hereafter called "parameter sets" – are generated. These sets comprise an "ensemble" of parameter sets. Using experimental observations gathered at different initial conditions, we then integrate the set of ODE's at these initial conditions for every parameter set, each of which is either kept or removed depending on its ability to predict experimental observations (Figure 1). In addition to whether each parameter set's prediction agrees with the experimental data, this "screening" step has the binary outcome of whether the ODE reaches steady-state at several perturbed system conditions. **The task of this project is to build a machine learner whose output is a prediction of the steady-state convergence of a given parameter set, given the inputs of each of that set's individual parameters.** This is an important task because even with several acceleration strategies[1], the existing model must integrate the system of ODE's for each parameter set, which is incredibly time consuming (often a few days parallelized on ~20 nodes on Quest). If, however, we were able to train on a small number of parameter sets and then only integrate the most promising ~1% (or so) of the test set, we could improve run times significantly, hopefully without sacrificing predictive ability.

My approach was to use each of the 883 parameters in each parameter set as an attribute, and to attempt to classify examples by the binary classification of whether that parameter set, when plugged in to the system of ODEs and integrated, converged to steady state. This was done first on a test data set of 100 examples, and later on a full data set of 10,000 examples. The learners used on the test data set were 1-Nearest Neighbor, 3-Nearest Neighbor, and level-1 Decision Trees; on the full data set, I used 1-Nearest Neighbor, level-1 Decision Trees, full (J48) Decision Trees, AdaBoosted level-1 Decision Trees, and Logistic Regression. While I was able to find a succinct level-1 DT for the test data set that had 77% precision on the parameter sets which passed, this learner did not perform well on the full dataset, and was unable to find a single attribute that was predictive of parameter sets which converged to steady state. Other learners similarly did poorer on the full data set, with only J48 being able to predict which parameter sets converged at a rate better than chance. I am unsure of why the preliminary success on the test set did not carry over to the full set; however, my best guess would be that the successful split in the test set was simply noise from the small set size and did not actually carry any meaningful information.

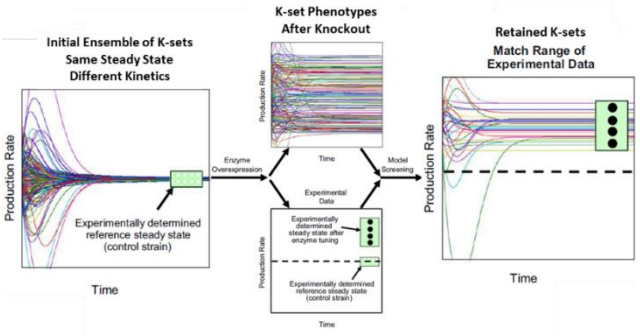


***Figure 1 (from Contador, et. al[3]):*** Ensemble Modeling screens parameter sets by comparing their predicted behavior (from ODE solutions) with experimental observations. (a) Parameter sets are generated to be consistent with a reference state. (b) Sets are then integrated at different conditions, at which experimental data is available. (c) Only those parameter sets which agree with experimental data are kept, and this process is iterated.

**Final Report**

Methods:

I used Weka to test the efficacy of several machine learning algorithms. On the preliminary/test data set, I tested 1-Nearest Neighbor, 3-Nearest Neighbor, and level-1 Decision Trees; on the full data set, I used 1-Nearest Neighbor, level-1 Decision Trees, full (J48) Decision Trees, and AdaBoosted level-1 Decision Trees. These were all used at their default settings. Logistic Regression and Multilayer Perceptrons (Neural Networks) were also attempted; however, these were unable to arrive at a solution within one hour and were abandoned.

For the remaining learners, I used 10-fold cross validation on all tests. On tests where pruning was available (all Decision-Tree based learners), I set aside 1/3 of the data for pruning.

Data Set:

Two data sets were used in this project: a test data set with 100 examples, and a final data set with 10,000 examples. Both data sets were generated by myself using the protocol outlined by Tran, *et. al*[2]. This procedure involves taking a network of reversible chemical reactions, such as A ⇄ B. First, an enzyme fraction $e_i$ is sampled for each species involved in the reaction, which determines what percentage of the enzyme E which carries out this reaction is bound (at the initial condition) to each species. Next, a reversibility $R$ is sampled, which determines the relative propensity for the reaction to go from B to A, as opposed to A to B. Lastly, using these values as well as the steady-state rate of this reaction at some experimental reference, the kinetic rate constants are calculated, which when multiplied by the concentrations of all reacting species in that reaction will give the reaction rate. In practice, this means that the enzyme fractions represent the initial conditions, the reversibilities represent the thermodynamics of the reactions, and the kinetic rate constants represent how quickly each reaction will reach equilibrium.

In my model network of 74 reversible reactions, each parameter set has 466 kinetic reaction rate constants, 184 enzyme fractions, and 233 reversibility constants. Each of these 883 parameters was thus treated as an attribute.

For each example, I attempted to classify the binary outcome of "steady-state convergence". This was the ability of a given example (parameter set) to converge to a steady solution when plugged into the system of ODEs. I was initially interested in also classifying whether the outcome predicted by a parameter set was consistent with experimental observations; however, this proved to be more challenging, as only a small fraction (about 1%) of examples were consistent with experimental data, and these did not seem to be informed by any individual attribute.

Results:

The results of the preliminary test (100 examples) and the final test (10,000 examples) are in Table 1 and Table 2 below. The preliminary test results were promising: in the test set of 100 examples (79 had classification of "no" for steady-state convergence, 21 as "yes"), the level-1 Decision Tree (Decision Stump) had an accuracy of 0.86, and a precision and recall of 0.769 and 0.476, respectively, on the classification of "yes" (I was mostly concerned with the precision of the "yes" classification, as this would tell me how many of the parameter sets I would keep would ultimately be successful). Additionally, the

3-Nearest Neighbor learner performed well, with a precision of 0.750, and the J48 learner had a precision of 0.526.

However, for the final test, these learners performed much more poorly. The level-1 DT was unable to identify a single attribute that predicted any classifications of "yes", and only the J48 learner was able to classify an outcome of "yes" with any accuracy greater than chance (i.e., the ratio of true positives to false positives is greater than the overall ratio of "yes" to "no" in the total data set).

*Table 1: Results of Preliminary Testing (100 examples)*

| | | ZeroR | | 1-NN | | 3-NN | | 1-level DT | | J48 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | classified as --> | no | yes | no | yes | no | yes | no | yes | no | yes |
| Converged? | no | **79** | **0** | **69** | **10** | **78** | **1** | **76** | **3** | **70** | **9** |
| | yes | **21** | **0** | **17** | **4** | **18** | **3** | **11** | **10** | **11** | **10** |
| | accuracy | 0.790 | | 0.730 | | 0.810 | | 0.860 | | 0.800 | |
| | Precision (on "y") | 0.000 | | 0.286 | | 0.750 | | 0.769 | | 0.526 | |
| | Recall | 0.000 | | 0.190 | | 0.143 | | 0.476 | | 0.476 | |
| | F-score | **0.000** | | **0.229** | | **0.240** | | **0.588** | | **0.500** | |

*Table 2: Results of Final Testing (10,000 examples)*

| | | ZeroR | | 1-NN | | 3-NN | | 1-level DT | | J48 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | classified as --> | no | yes | no | yes | no | yes | no | yes | no | yes |
| Converged? | no | **7271** | **0** | **5098** | **2173** | **5744** | **1527** | **7271** | **0** | **5373** | **1898** |
| | yes | **2729** | **0** | **1927** | **802** | **2168** | **561** | **2729** | **0** | **1793** | **936** |
| | accuracy | 0.727 | | 0.590 | | 0.631 | | 0.727 | | 0.631 | |
| | Precision (on "y") | 0.000 | | 0.270 | | 0.269 | | 0.000 | | 0.330 | |
| | Recall | 0.000 | | 0.294 | | 0.206 | | 0.000 | | 0.343 | |
| | F-score | **0.842** | | **0.281** | | **0.233** | | **0.000** | | **0.337** | |

These results are not only disappointing but also confusing, as aside from the final set of examples being generated in Quest, the examples were made using the same procedure. While I am unsure why the learners were successful with the preliminary data set and unsuccessful with the larger final data set, my best guess is that the attributes identified in the preliminary test were simply noise in the data. Therefore, the larger number of examples in the final data set showed that the trends seen in the smaller data sets didn't hold in general.

<u>Future Work:</u>

There are a few potential strategies which might be able to improve these outcomes. First, I would like to attempt more data preprocessing of the attributes. Specifically, I think that it is likely that individual attributes are not as important as average or outlier statistics from groups of attributes. Because each reaction in our network contributes 6 to 10 rate constants, 2 to 4 enzyme fractions, and 2 to 5 reversibilites, it may be meaningful to simply take the largest, smallest, or average value of each of these metrics from a given reaction, instead of using all of them individually.

Additionally, the outcome of parameter set convergence is not truly binary; rather, the value is set against a threshold, which in turn gives the binary output. It may be that this threshold is erroneously low, causing too many adequate parameter sets to be determined to not converge. Therefore, I will raise this threshold to a value which includes roughly ~50% of initial parameter sets (instead of the current ~20-30%), which will hopefully give the learners a better chance of accurately classifying the examples.

**References:**


1. Greene, J. L., Wäechter, A., Tyo, K. E. J. & Broadbelt, L. J. Acceleration Strategies to Enhance Metabolic Ensemble Modeling Performance. *Biophys. J.* **113,** 1150–1162 (2017).

2. Tran, L. M., Rizk, M. L. & Liao, J. C. Ensemble modeling of metabolic networks. *Biophys. J.* **95,** 5606–5617 (2008).

3. Contador, C. A., Rizk, M. L., Asenjo, J. A. & Liao, J. C. Ensemble modeling for strain development of L-lysine-producing Escherichia coli. *Metab. Eng.* **11,** 221–233 (2009).