CERTARA | STRATEGIC CONSULTING

Comparison of Robustness and Efficiency of Four Machine Learning Algorithms for Identification of Optimal Population Pharmacokinetic Models

Mark Sale M.D. (1), Mohamed Ismail Pharm. D., M.S. (2, 4), Fenggong Wang Ph.D. (3), Kairui Feng Ph.D. (3), Meng Hu Ph.D. (3), Liang Zhao Ph.D., MBA (3), Robert Bies Ph.D. (4)
Institution: (1) Certara, (2) Enhanced Pharmacodynamics LLC, (3) FDA (4) State University of New York at Buffalo

Background & Objectives

Model selection for population PK/PD models typically starts with a simple, even trivial model. Then, "features" are added to the model and tested the consistency of the model with the observed data, the statistical properties and plausibility of the resulting model. These features traditionally start with structural features (compartments, absorption models), then the structural model is fixed, and covariates are examined. This process, is time consuming and can be rate-limiting in a drug development program. More importantly, there is reason to believe that the "adding one at a time" approach to model building is fundamentally flawed. In the field of optimization, this is known as a "local search", a "downhill search", where only candidate model very similar to the current candidate are examined. While efficient (arriving at the final answer with the fewest model evaluations), the robustness of the method (likelihood of finding the optimal or near optimal solution) is in question. Chen et. al [1] recently showed that the sequence of evaluation of features can be expected to influence the final model, and thus that the "convexity" assumption for the model space is violated. Wade and Beal noted important interactions between model features [2]. Further, Wade and Beal showed that these interactions occur across categories of features, e.g., structural features interacting with statistical effects and covariates. These results suggests that the one-at-a-time method may frequently miss better models, as combinations of features may need to be added to a model to see improvement. The one-at-a-time approach has the potential to essentially get stuck in a local minimum. Machine learning (ML) approaches, while typically less efficient, can use "global search" methods to overcome what is essentially a local minimum problem with local search [3] (Figure 1). However, all global search algorithms are based on assumptions about the structure of the "model goodness search space" that can only be overcome by a local search. We present a case where a "2-bit" local exhaustive search in combination with global search is required to find the true optimal model.

Methods

A simulation data set was constructed consisting of:

- Linear 2 compartment, first order absorption (ADVAN4), Typical Value (TV) for Clearance (CL) = 200 L/hr, TV for Central Volume (Vc) = 1000 L, Ka of 2/hr, each with log normal between subject variance of 0.2, K23 and K32 of 0.2/hr
- An absorption lag time with a TV of 0.2 hours (log normal variance of BSV = 0.2)
- True covariates included: CL~ (Weight, bilirubin, race and ALT), Vc ~Weight. Ka~age
- Three additional covariates were included that did not influence the model

Four different ML methods were used for a global search.

- Gaussian process/Bayesian Optimization (GP)
- Random Forest (RF)
- Gradient Boosted Random Tree (GBRT)
- Genetic algorithm (GA)

GP, RF and GBRT were implemented with the scikit-learn package[4]. GA was implemented with the DEAP package [5]. A command line interface for all methods was developed in Python.

The search space for the model selection consisted of 10 dimensions:

- Number of compartment (1,2,3)
- Volume as a function of Weight (yes|no)
- Volume as a function of Sex (yes|no)
- Clearance as a function Weight (yes|no)
- Clearance as a function Age (yes|no)
- Between subject variability (BSV) on Ka (yes|no)
- K23/K32 (if present) as a function of Weight (yes|no)
- Absorption model (first order|zero order|combined zero, then first order) vs Absorption lag time (yes|no)
- BSV on zero order absorption or Lag time, if present (yes|no)
- Residual error model (additive|proportional + additive)

Methods (continued)

An exhaustive search (12,960 models) was done to find the "true" optimal model. The true optimal model was two compartment, with a zero-order absorption and combined proportional + additive residual error. No covariates were included in the true optimal model. The simulation model had a higher (worse) reward than the "true" optimal model as it failed the covariance step, and thus incurred a 300 point penalty (100 each for covariance, correlation and condition number).

The four ML methods were then applied to search the model space. The search criteria included:

- Objective function value (OFV)
- Parsimony penalty (10 points for each estimated parameter, THETA, OMEGA and SIGMA)
- 100 point penalties for failing to converge, failing the covariance step, failing the correlation test, and a condition number > 1000.

These criteria can be defined by the user, if for example, a successful covariance step is of interest. In addition, user-defined R or Python code can be executed at the end of the model run to add other user-defined criteria, e.g., posterior predictive check, coded in R. No penalties specified in R or Python code were used in this example.

Results

All ML methods failed to identified the "true" optimal model. All identified a 2-compartment model, without the zero-order infusion. Examination reveals that the "model goodness" surface shows interactions consistent with those described by Wade et. al. [1994]. Because of this failure, a "1-bit local search" was added. A 1-bit local search takes the current best model and systematically changes each bit in the model representation, e.g., 2 bits are needed to specify 1|2|3 compartments. These are:

- 1 compartment ([0,0])
- 2 compartment ([0,1] and [1,0])
- 3 compartment ([1,1])

Note that redundancy will occur with 3 options as 2 bits are required. Assume the best model 2 compartment (is 0,1). Each of the 2 bits would be changed ([0,1] -> [1,1] and [0,0]) resulting in 2 models in the local search (1 and 3 compartments). 16 bits were required to represent the 10-dimensional search space. 16 models were generated in the local search at each iteration. The 1-bit local search still did not result in the true optimal model. A 2-bit search was implemented, where all combinations of 2-bit changes were examined. This resulted in 136 new models (Table 1). The 2-bit search was required to find the true optimal model. All methods found the true best model on the 1st step of the 2-bit search. GA was slightly faster than the other methods. Table 1 explains the interaction between these bits. In this table, the 1st bit change is in rows, and the 2nd bit change in columns. For example, the results of changes in bits 4 and 3 will result in a reward (penalized OFV) of 4928.74 (blue). 1-bit changes are on the diagonal, e.g., changing only bit 5 is in row 5, column 5, (green) with a reward of 4918.77. The true best model is in [15,2] with a reward of 4818.16. The reference model (best model from the ML methods) had a reward of 4882.77. Note that neither of the single bit changes that were included in this best model (bit 2 alone and bit 15 alone) resulted in a model that was better than the reference. Model [2,2] did not complete and model [15,15] had a reward of 4922.22. In a one-change-at-a-time model building method, these models are rejected, and the true best model is not found. Figure 1 depicts what is essentially a local minimum in the reward surface at the ML model, with each the 2 required single bit changes alone having a higher reward than the ML best model.

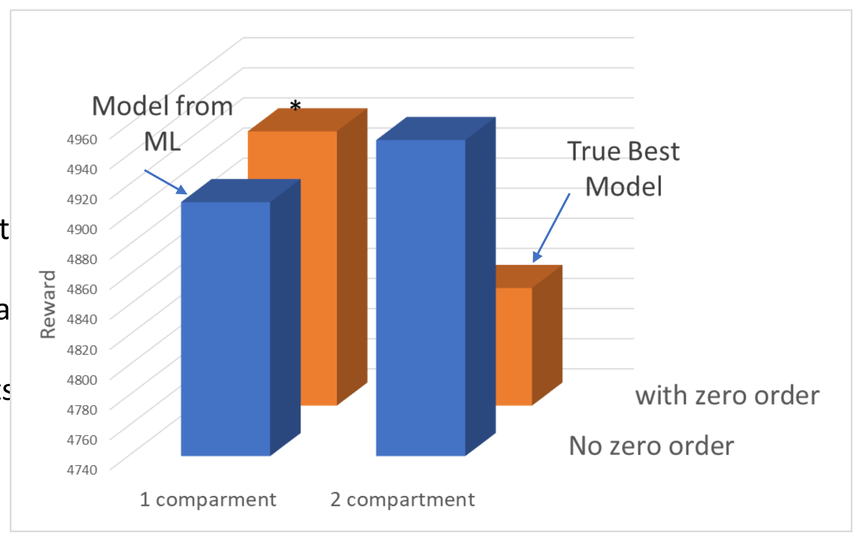
Discussion

ML has been shown to be a more robust method of model selection than traditional manual "model building" [3]. Still, the methods alone are not completely robust. This lack of robustness is related to the assumptions made about the search space. For example, GP is based on an assumption of a smooth multivariate distribution of the reward surface. This assumption is shown to be violated in Figure 1, resulting in a lack of robustness. The current application, written in Python is very cumbersome. Work is ongoing to develop an R Shiny app to facilitate the creation of the complex text files (mostly JSON) required to specify the model search space (Figure 2) and monitor the search process (Figure 3).

Table 1, 2-bit local search results

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash
2		crash	5165.69	4827.70	crash	5032.32	5417.99	crash	4924.46	4923.49	5148.49	5167.12	4928.49	24018.29	4818.16	5776.74
3			4922.70	4928.74	crash	4922.60	4922.60	4924.50	4924.66	4924.39	5158.77	5128.77	5128.77	6716.45	4918.22	5666.87
4				4932.68	crash	4932.58	4932.58	crash	4934.63	crash	5178.74	5138.74	5138.74	5144.30	4927.68	5647.75
5					4918.77	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash
6						99999.00	5032.60	4925.97	4926.18	4928.11	5152.60	5132.60	5132.60	crash	4922.13	5670.00
7							5032.64	4925.97	4926.18	4928.11	5152.60	5132.60	5132.60	crash	4922.13	5670.00
8								4926.09	crash	4930.31	5154.50	crash	5134.50	crash	4923.91	5669.31
9									4926.30	4930.41	crash	5134.66	5134.66	5310.48	4924.08	5669.31
10										4928.20	5154.40	5134.39	5134.40	5548.57	4923.87	5674.41
11											5152.70	5138.77	crash	5441.17	5156.14	6006.24
12												5132.70	crash	32929.81	5128.22	5976.87
13													5132.70	5137.88	4926.14	5776.24
14														5315.49	crash	6233.71
15															4922.22	5662.69
16																5670.06

Figure 1. Local Search Results



* 2-compartment model without zero order infusion did not complete. In the search, this was assigned a value of 999999. This value was set to 4950 for visualization.

Figure 2 – Model search space setup dialog

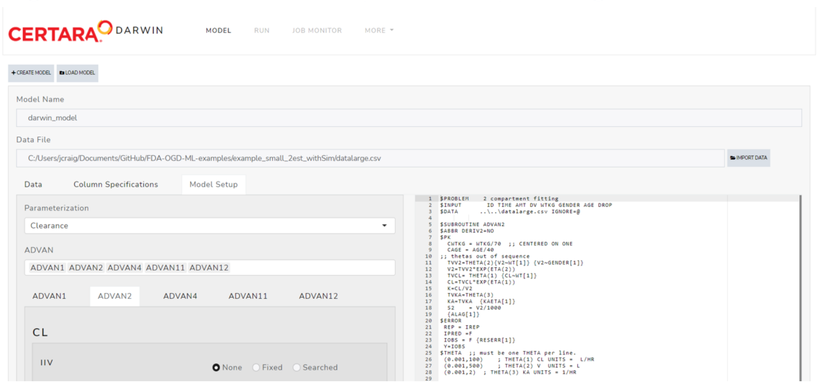
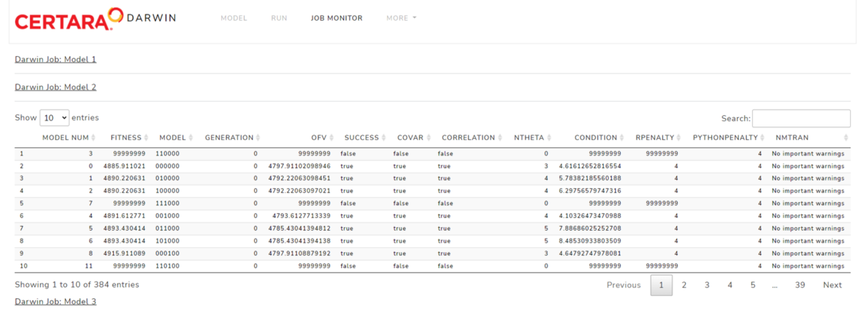


Figure 3 – Model search monitoring dialog



This work was supported by FDA/NIH grant (U01FD007355) (Development of a model selection method for population pharmacokinetics analysis by deep-learning based reinforcement learning (RFA-FD-21-027)). Views expressed in this poster do not represent FDA's views or policy.

[1] PAGE 30 (2022) Abstr 10091 [www.page-meeting.org/?abstract=10091
[2] Wade JR, Beal SL, Sambol NC. 1994 Interaction between structural, statistical, and covariate models in population pharmacokinetic analysis. J Pharmacokinet Biopharm. 22(2):165-77
[3] E. Sherer, et al., 2012 Application of a single-objective, hybrid genetic algorithm approach to pharmacokinetic model building. Journal of Pharmacokinetics and Pharmacodynamics. 39(4): 393-414
[4] https://scikit-learn.org
[5] https://github.com/deap/deap

Want to learn more?
<< Scan Here
Scan me