Metaheuristics for Subgroup Identification in Clinical Trials

Will Gertsch

December 22, 2022

Abstract

Heterogeneous treatment effects pose a major challenge in the analysis of randomized clinical trials. When not properly addressed, differences in treatment effects for subgroups of the population will lead to a biased estimate of the treatment effect. We use a mixture model approach to identify a group with a higher treatment effect. Because standard optimization methods may prematurely converge to local optima, we use metaheuristic algorithms to estimate the model parameters. We conduct a simulation study to determine which algorithms have the best best performance. Finally, we apply our method to a trial of a behavioral intervention for HIV to find a subgroup with an improved treatment effect.

1 Introduction

Randomized controlled trials (RCT) are the gold standard for determining the efficacy of a treatment or intervention. However, most analyses of clinical trials assume that the treatment effect is accurately depicted by a single value. For example, common practice when working with clinical trial data is to base inference and decision making around the estimation of an average treatment effect. This assumes that all the treatment effects come from a single underlying distribution. In practice, it is likely that the treatment effects will be more hetereogenous than what can be described by a single distribution. Therefore, a common post-hoc activity is to perform a subgroup analysis to investigate sources of heterogeneity in the treatment effect.

There are a wide variety of different methods for performing subgroup analyses. If there are known subgroups of interest, the analysis used for the entire population can be applied to the subgroups of interest. Another approach is to augment the analysis of the whole data with additional interaction effects to differentiate treatment effects for different subgroups of the trial population. These analyses should be performed with care to avoid statistical errors and data-dredging. A review of statistical issues in subgroup analysis for clinical trials can be found in Rothwell (2005).

Another approach to subgroup analysis is to use an algorithm to automatically search the data for interesting subgroups. This approach is popular because interesting subgroups are not usually known a priori. A review of data-driven methods for subgroup identification can be found in Lipkovich et al. (2017). Most methods use some variation of classification trees to identify latent subgroups. Another approach is change-point regression models found in Wang et al. (2019).

An alternate approach using mixture models is proposed by Shen, He (2015) where the latent subgroup membership is modeled using a logistic regression. This model is commonly referred to as a "mixture of experts" in computer science and extensions to survival and longitudinal data can be found in Wu et al. (2016) and Shen, Qu (2020). These models have the advantage of being easier to interpret than the other approaches and can provide other statistical features such as a test for latent subgroup existence. These models are also easy to extend to other types of data that are commonly found in clinical trials.

Finding latent subgroups usually involves solving one or more optimization problems. For instance, the mixture model approach requires the maximization of the log-likelihood of the model. This is usually accomplished using standard methods such as Newton-Raphson or the expectation maximization (EM) algorithm. However, these methods are sensitive to starting values and may converge to local optima if the initial values are not properly chosen. These methods may also be complex to implement. Newton-based algorithms often require lengthy derivations of the gradient and/or Hessian functions and may be infeasible or impractical to implement for complex models. The EM algorithm can also be complex to implement, requiring sub-optimization problems to be solved during the M-step, such as the fitting of a logistic regression.

Metaheuristic optimization algorithms are one option for avoiding the issues encountered when trying to maximize the likelihood for complex mixture models. These algorithms make very few assumptions about the problem and only require the likelihood function be specified. Stochastic elements are also incorporated so that the algorithms avoid getting stuck in local optima. Examples of metaheurstic optimization algorithms include simulated annealing (SA) (Bertsimas, Tsitsiklis, 1993), particle swarm optimization (PSO) (Kennedy, Eberhart, 1995), genetic algorithms (GA) Holland (1992), and the whale optimization algorithm (WOA) (Mirjalili, Lewis, 2016).

Metaheurstics have been used for fitting a variety of statistical models. Mohanty (2012) used PSO to fit spline regression models for astronomy data. Khamees et al. (2022) use a metaheuristic to estimate the parameters of a mixture model for modeling wind speeds. Shin et al. (2014) showed in an example that a genetic algorithm can outperform the EM algorithm at smaller sample sizes when estimating a mixture of Gaussians.

The structure of the paper is as follows. In section 2, we introduce metaheuristic algorithms in more detail and give an example of such an algorithm. In section 3, we review the theory of mixture models and define a mixture model for subgroup analysis. In section 4, we evaluate the performance of different metaheuristics using a simulation study. In section 5, we apply a metaheuristic-based approach to perform a subgroup analysis of a real clinical trial. Finally, section 6 provides conclusions, limitations of the current work, and directions for future work.

2 Metaheuristics

2.1 Introduction

These difficulties in maximizing the log-likelihood may be alleviated by using a metaheuristic optimization algorithm. Metaheuristics are a class of algorithm inspired by natural processes and are designed to be able to solve almost any optimization problem. These algorithms make very few assumptions about the problem, but are often able to find a quality solution in a reasonable time. Metaheuristics also incorporate stochastic elements that allow the algorithm to avoid premature convergence to local optima.

In the following subsection, we introduce a common variant of metaheuristic that has good performance on many problems. Interested readers may refer to the references section for more details on the other algorithms used in this paper.

2.2 Differential Evolution

Differential evolution is an evolutionary algorithm introduced by Storn (1996). The algorithm mimics the process of natural selection by generating successive generations of solutions and selecting the best solutions according to the objective function. It has been widely used to solve problems in engineering design and financial research. For example, in optimal design, Xu et al. (2019) used differential evolution to find optimal designs for a high-dimensional logistic regression.

Differential evolution acts iteratively on a population of solution vectors $x_{i,G}$ for $i = 1, \ldots, NP$ and has 3 main steps: mutation, crossover, and selection. After generating an initial population of solutions, the algorithm repeats these 3 steps to evolve the solutions towards the optimum. First, the mutation operation generates a mutated vector for each solution in the population by

$$v_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G})$$

where r_1, r_2, r_3 are random, mutually different, indices from $\{1, \dots, NP\}$ and $F > 0$. Next, the algorithm performs the crossover operation by generating a trial vector

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \dots, u_{Di,G+1})$$

where

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & uniform(0,1) \leq \text{CR or} j = rnbr(i) \\ x_{ji,G} & uniform(0,1) > \text{CR and} j \neq rnbr(i) \end{cases}$$

for $j = 1, \dots, D$ where $CR \in [0, 1]$ and $rnbr(i)$ is a randomly chosen index from $1, \dots, D$ to ensure at least one parameter will be adopted from the mutation step. Finally, in the selection step, the new vector produced by the mutation and crossover steps is compared to the original vector. If the new vector has a better objective value, then the new solution is selected to become part of the new population. Otherwise, the old value is retained. This process repeats for G generations and the best solution vector in the final population is chosen as the final solution.

3 Mixture models for subgroup identification

A mixture model assumes that observations are sampled from a population made up of g subpopulations. We may write the general mixture probability density as

$$f(y|w, \theta) = \sum_{j=1}^{g} w_j f_j(y|\theta_j) \tag{1}$$

where the f_j may either be from the same distributional family or from different families. If we have n observations, $y_1, \ldots, y_n$, the we may write the log-likelihood function as

$$\ell(w, \theta | y) = \sum_{i=1}^{n} \log \sum_{j=1}^{g} w_j f_j(y_i | \theta_j) \tag{2}$$

Note that the log function does not distribute over the innermost summation term. This makes it difficult to derive maximum likelihood estimator (MLE) analytically in most cases. Therefore, the log-likelihood is usually maximized numerically using the EM algorithm.

3.1 Logistic Normal mixture model

Shen, He (2015) propose the logistic-normal mixture model (LNM) for the analysis of subgroup data. Let Y_i for $i = 1, \ldots, n$ be a sample of n observations from a trial with two treatments and an unobserved latent factor with two levels. We define the logistic-normal model as

$$\begin{gathered} Y_i = Z_i'(\beta_1 + \beta_2 \delta_i) + \epsilon_i \\ P(\delta_i = 1 | X_i) = \pi(X_i'\gamma) = \exp(X_i'\gamma)/(1 + \exp(X_i'\gamma)) \end{gathered} \tag{3}$$

where $\epsilon_i \sim N(0, \sigma^2)$. The model includes two covariate vectors Z_i and X_i for the subgroup mean and predicting latent subgroup membership. The latent variable δ_i is an indicator of subgroup membership, equal to 1 if participant i is in the latent subgroup and 0 otherwise. The parameter vectors β_1 and β_2 are the treatment effect vector and subgroup interaction effect vector. It is possible to let X_i contain as many parameters as needed, but in our current application to randomized controlled trials, we will only include an intercept and treatment effect term. Without loss of generality, we can identify the model specifying the interaction effect direction, e.g. by assuming that $\beta_{22} > 0$.

The intuition behind this model is that there are two latent sub-populations within the clinical trial population. The interaction between the treatment and latent subgroup allows for the possibility of a subgroup that experiences a treatment effect while the other participants do not. By maximizing the likelihood, the model may be thought of as performing a search for those participants with a different treatment effect compared to the rest of the population.

Because we do not assume that information matrix is available, it is difficult to perform statistical inference for the parameters in this model, especially when using metaheuristics. Therefore to obtain confidence intervals for the model, we apply a bootstrap procedure (Efron, 1992). The data is resampled with replacement and the model is fit to the resampled data. Parameter estimates from this model are then saved and the process is repeated. Many bootstrap samples should be used in order to ensure accurate inference. Confidence intervals may then be constructed for each model parameter by taking the the desired quantiles of the saved parameter estimates.

4 Simulation study

In this section, we will evaluate the performance of different metaheuristic algorithms for estimating the model parameters. The first part of the study will evaluate the performance of different algorithms at maximizing the likelihood and estimating the parameters. The second part of the study will evaluate the performance of the bootstrap procedure for different algorithms. The simulation was done using Julia (Bezanson et al., 2017), with metaheuristic algorithms from the Metaheuristics.jl package (Dios Mejía-de, Mezura-Montes, 2022). Both parts, default tuning parameters were used for all algorithms. An attempt to automatically choose better tuning parameters using DE failed to produce any meaningful benefit for any of the algorithms.

For both studies, simulated data was generated from a clinical trial where there was overall treatment effect, but only some participants actually benefited from the treatment.

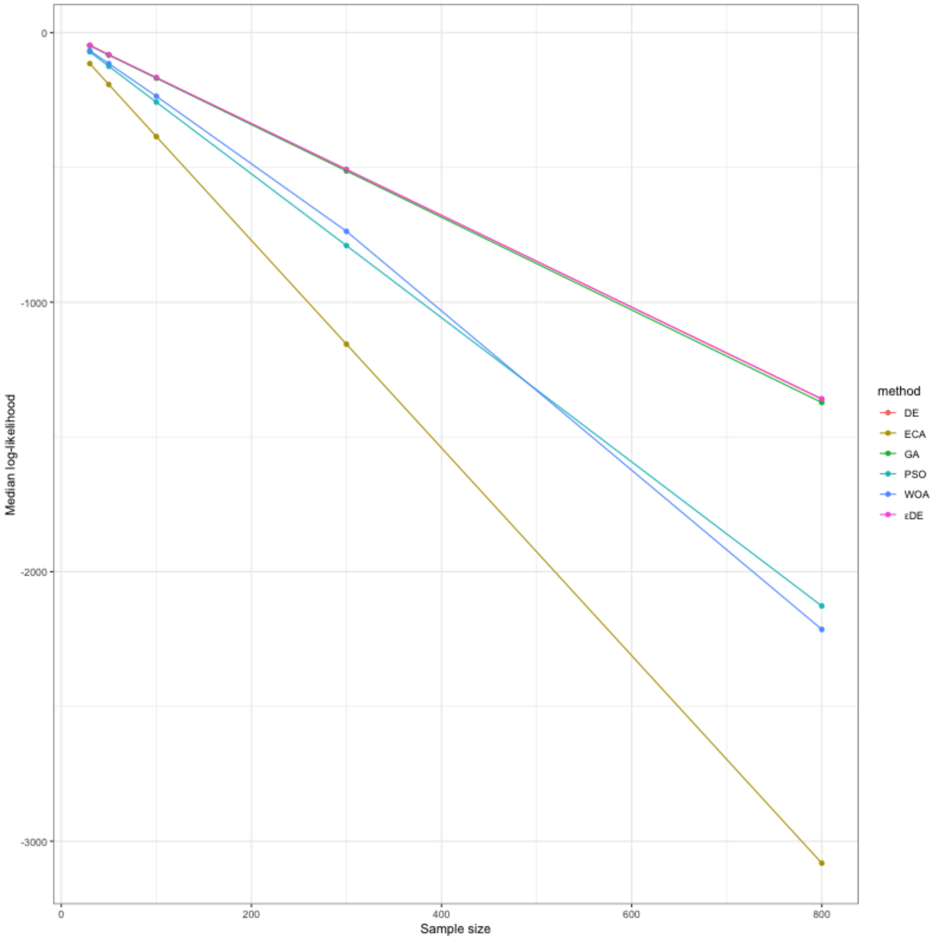
A single binary covariate (sex) was included to predict group membership. The clinical trial was simulated assuming that the ratio of male to female was $3 : 2$ and the logistic regression parameters were chosen such that the probability of being in the latent subgroup was 0.6 if the participant was female and 0.2 if the participant was male. The resulting parameter values for the model were set as $\beta_1 = (80, 0)'$, $\beta_2 = (0, 30)'$, $\gamma = (-1.39, 1.79)$, and $\sigma = 1$. Parameter bounds were chosen based on the maximum and minimum values of the data and the variance of the data.

4.1 Simulation study 1: parameter estimation

The first simulation study evaluated the performance of different algorithms at different sample sizes. Each simulation run, a simulated data set was generated according the specifications described earlier and the model was fit to the simulated data using all 7 algorithms. The log-likelihood and parameter estimates were saved for each run. The sample sizes used were 30, 50, 100, 300, 800 and the simulation was run 1000 times for each of these sample sizes.

Figure 1 and Figure 2 show the results of the simulation study. Figure 1 shows the median log-likelihood value attained by each algorithm at different sample sizes. The evolutionary algorithms (GA, DE, ε-DE) performed the best while the swarm-based algorithms (PSO, WOA) had the next best performance. ECA performed poorly on this problem as well as SA, which was excluded from the plot for better visibility. This plot suggests that the evolutionary algorithms should be used over the swarm-based methods for estimating the parameters. Figure 2 tells a similar story. The plot shows that root mean squared error (RMSE) for the treatment-latent subgroup interaction effect decreased quickly as the sample size increased when the model was estimated using one of the evolutionary methods. This trend was not seen when using the other algorithms, suggesting sub-optimal performance. The RMSE values were also much lower for the evolutionary algorithms. Both plots demonstrate that evolutionary

Figure 1: Median log-likelihoods by sample size by algorithm.



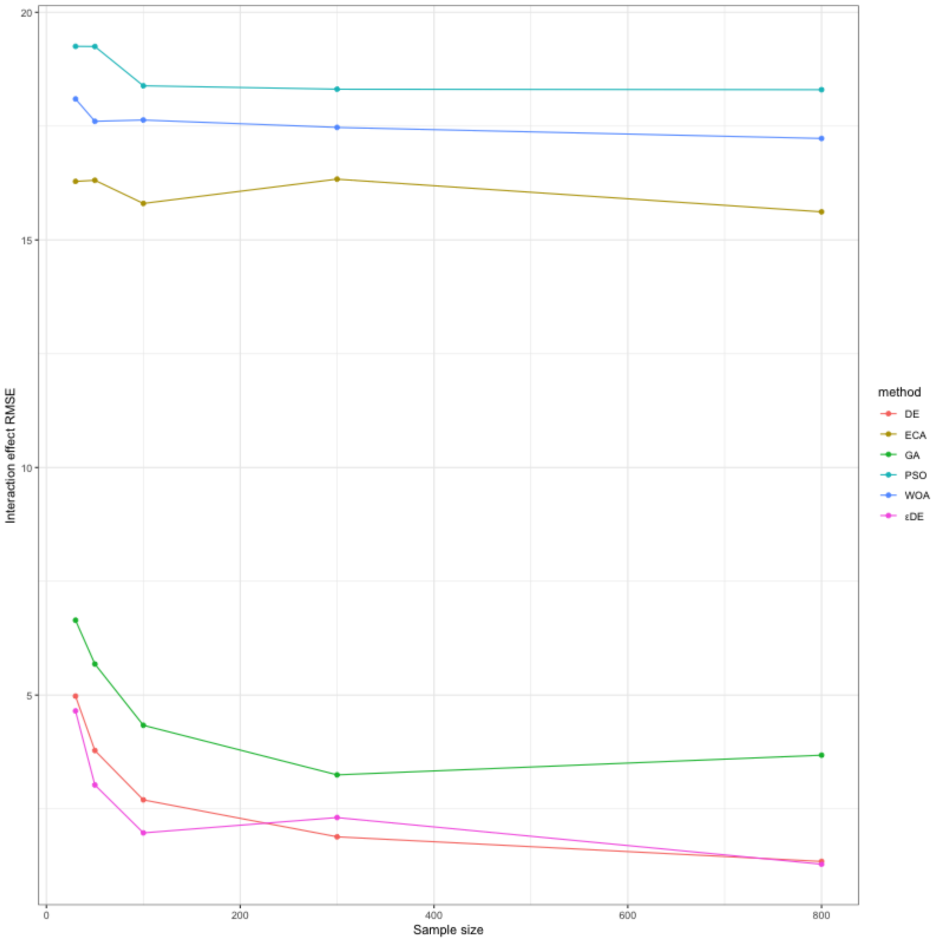
algorithms can have better performance than swarm-based methods when the search space is relatively small. If the bounds for the parameter values were increased, methods such as PSO may show improved performance relative to the evolutionary algorithms.

4.2 Simulation study 2: bootstrap confidence intervals

In the second simulation study, we evaluated the coverage rates of the bootstrap confidence intervals. Because computing the bootstrap intervals is computationally intensive, we only computed the coverage rates for the best performing algorithms in the previous study. These algorithms were DE and ε-DE.

Similar to the previous study, the simulated clinical trial data was generated 1000 times and the algorithms were used to compute 95% bootstrap confidence intervals for each simu-

Figure 2: Root mean squared errors by sample size by algorithm.



lated data set. The number of bootstrap samples for each run was 1000 and the bootstrap sample size was 80. The sample size of the generated data was fixed at 100. Intervals were constructed for every parameter in the model and checked against the parameter true values to see if the interval covered the true value.

Method	β_{11}	β_{12}	β_{21}	β_{22}	γ_0	γ_1	σ
DE	0.988	0.989	0.989	0.993	0.991	0.989	0.804
ε-DE	0.988	0.989	0.989	0.993	0.991	0.989	0.804

Table 1: Coverage rates for 95% bootstrap confidence intervals.

Table 1 shows the coverage rates for each algorithm by model parameter. Coverage was good for both algorithms with DE performing slightly better. The algorithm was able to construct confidence intervals with high coverage for all of the model parameters. Because the coverage was greater than 95% for some parameters, this algorithm for constructing interval estimates may be overly conservative. The confidence intervals for the variance parameter seems to be more difficult to construct using this method, resulting a coverage much below what was expected. Therefore, when conducting inference using this method, we should expect interval estimates to be slightly too wide for the regression parameters and too narrow for the variance parameter.

5 Application

In this section we apply the metaheuristics estimation approach to analyze a real data set. The ATN CARES study was a multi-arm randomized trial designed to evaluate the efficacy of an mHealth intervention to improve HIV-related outcomes for high-risk populations in Los Angeles and New Orleans. The intervention was a mobile phone based service that connected participants to coaching and peer support services to encourage healthy practices such as taking antiretroviral medication needed to control their HIV infection. For this analysis, we focus on the sub-population of HIV+ participants to see if the intervention reduced viral load at 24 months. We performed a complete case analysis using participants that had baseline and 24 month viral load information and demographic information. The total sample size was 75 participants.

A linear model was fit with 24 month viral load as the outcome and the intervention as the predictor. We also controlled for baseline viral loads. Since viral load data tends to be highly skewed, we applied a log10 transformation to both viral load variables. The results of the linear model showed that there was no significant intervention effect for the 24 month viral loads.

Because there was no intervention effect, we were interested to see if there was a subgroup of participants that experienced a positive intervention effect. We chose to identify this latent subgroup using baseline viral loads, sex of the participant, and whether or not the subject was black. We used this data to estimate the logistic normal model where the Z matrix contained intervention and baseline viral load variables and the X matrix contained all the baseline variables described previously. We used fit the model using all available algorithms using default tuning parameters.

Method	Log-likelihood	β_{10}	β_{12}	β_{13}	β_{20}	β_{21}	β_{22}	γ_0	γ_1	γ_2	γ_3	σ
DE	-68.89	2.62	0.33	0.45	-1.21	-0.34	-0.46	3.69	-0.27	-0.94	-0.43	0.43
ε-DE	-69.47	2.58	0.33	0.43	-1.18	-0.33	-0.45	4.20	-0.70	-1.30	0.45	0.45
GA	-101.19	1.30	0.44	0.12	0.58	1.87	-3.96	0.28	2.76	-4.58	-0.89	0.89
PSO	-102.49	1.30	0.45	0.07	4.20	-0.95	-4.20	-10.00	-5.00	-5.00	-0.95	0.95
WOA	-102.73	1.30	0.46	-0.04	-4.20	-3.87	-4.19	-3.13	-5.00	-0.01	-0.95	0.95
ECA	-172.98	3.26	-0.29	3.89	-2.01	-1.30	-0.06	8.16	1.69	-3.90	-1.52	1.52
SA	-206.22	4.25	-1.98	0.73	-1.36	0.89	-3.21	-4.81	2.88	0.00	3.75	3.75

Table 2: Results of all algorithms on viral loads data.

Table 2 shows the log-likelihood values and parameter estimates obtained by each algorithm. The ranking of the algorithms was very similar to the simulation study, with the two variants of differential evolution providing the best estimates. Looking at the difference in γ estimates between DE and ε-DE results, it is clear that minor differences in the objective function may result in major differences in the actual classification of the data.

The 95% bootstrap confidence interval for the intervention effect β_{22} was (-0.95, -0.05). This suggests that there was a small but significant intervention effect difference between the latent subgroups. A similar approach can be applied to the γ parameters to conclude significance about the prediction of subgroup membership, but the intervals all included 0.

6 Discussion

In this paper, we demostrated how metaheuristics may be used for subgroup analysis. We used several different algorithms to fit a logistic-normal mixture model for identifying participants of a clinical trial that had a heterogeneous treatment effect due to a latent factor. In addition to estimating the parameters, we were able to obtain interval estimates for all parameters in the model using bootstrap confidence intervals. Simulation studies showed that differential evolution was the best performing algorithm for the task of parameter estimation for this model. Boostrap confidence intervals were mostly conservative with estimation of the variance intervals proving the most difficult. We also applied this method to a real trial of a behavioral intervention for preventing HIV and were able to identify a group of participants with a higher treatment effect compared to the rest of the participants in the study.

One limitation of the current work is that the performance of the metaheuristic algorithms are not compared against the EM algorithm, which is the standard method for parameter estimation. Previous work has shown that for simpler mixture models, metaheuristics can match or exceed the performance of the EM algorithm for parameter estimation. It is a reasonable guess that the performance of metaheuristics will also be competitive for this problem as well. An additional advantage is that metaheuristics are much easier to implement than the EM algorithm and are less sensitive to starting values.

Another limitation of the current work is that the model cannot handle longitudinal data. This is a common issue with many subgroup analysis methods. Time and time-treatment interaction effects may be included in this model, but the correlation between measurements on the same individual over time are ignored. To address this issue, we extended the model to include a random effect term. Suppose we observe Y_{ij} for individuals $i = 1, \dots, n$ and time-points $j = 1, \dots, m$. Consider the following model

$$\begin{aligned} Y_{ij} &= Z'_{ij}(\beta_1 + \beta_2\delta_i) + \xi_i + \epsilon_{ij} \\ P(\delta_i = 1|X_i) &= \pi(X'_i\gamma) = \exp(X'_i\gamma)/(1 + \exp(X'_i\gamma)) \end{aligned} \tag{4}$$

where $\xi_i \sim N(0, \tau^2)$ is a random intercept term to model individual variation at baseline and to induce correlation for repeated measures. We assume that group membership is constant over time and that the random intercept term is independent of the residual errors.This model is similar to model in Shen, Qu (2020). Unfortunately, this model is difficult to fit, possibly because the fixed effect for the treatment effect is too inflexible. Adopting a random treatment effect such as in Shen and Qu may lead to better results. Metaheuristics should also have an advantage for estimating this model because of their ease in implementation compared to deriving an EM algorithm.

This work has a number of future directions. As mentioned previously, extension to longitudinal data would help fill a gap in the subgroup analysis literature. The model in Shen and Qu is a good starting place and a metaheuristics based approach should be able to extend their model to binary outcomes and possibly other types of outcomes. Having more methods for longitudinal data would be helpful as most modern clinical trials collect data over time on the same participants.

Another area of future work is automated parameter tuning. In the current work, an automated procedure using DE failed to improve the performance of the algorithm Other tuning algorithms, such as the racing algorithm (Birattari et al., 2002), may perform better. An important problem related to this is the issue of how to train algorithms on different data sets simulated from a variety of true parameter values. The selection of true parameter values is nontrivial and should reflect the wide array of parameter values that would be possible for the model in question. Previous work in Bayesian prior elicitation may prove useful in this area.

A final area for future work is the extension of the model more than two classes of participants. The patients in a clinical trial may be organized into three or more subpopulations, each having its own treatment effect. Estimation complexity increases when assuming more than two latent groups because a multinomial regression must be used instead of logistic regression. This makes it more difficult to derive an EM algorithm for these models and makes the use of metaheuristics more attractive.

References

Bertsimas Dimitris, Tsitsiklis John. Simulated annealing // Statistical science. 1993. 8, 1. 10–15.

Bezanson Jeff, Edelman Alan, Karpinski Stefan, Shah Viral B. Julia: A fresh approach to numerical computing // SIAM Review. 2017. 59, 1. 65–98.

Birattari Mauro, Stützle Thomas, Paquete Luis, Varrentrapp Klaus, others . A Racing Algorithm for Configuring Metaheuristics. // Gecco. 2, 2002. 2002.

Dios Jesús-Adolfo Mejía-de, Mezura-Montes Efrén. Metaheuristics: A Julia package for single- and multi-objective optimization // Journal of Open Source Software. 2022. 7, 78. 4723.

Efron Bradley. Bootstrap methods: another look at the jackknife // Breakthroughs in statistics. 1992. 569–593.

Holland John H. Genetic algorithms // Scientific american. 1992. 267, 1. 66–73.

Kennedy James, Eberhart Russell. Particle swarm optimization // Proceedings of ICNN'95-international conference on neural networks. 4. 1995. 1942–1948.

Khamees Amr Khaled, Abdelaziz Almoataz Y, Ali Ziad M, Alharthi Mosleh M, Ghoneim Sherif SM, Eskaros Makram R, Attia Mahmoud A. Mixture probability distribution functions using novel metaheuristic method in wind speed modeling // Ain Shams Engineering Journal. 2022. 13, 3. 101613.

Lipkovich Ilya, Dmitrienko Alex, B D'Agostino Sr Ralph. Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials // Statistics in medicine. 2017. 36, 1. 136–196.

Mirjalili Seyedali, Lewis Andrew. The whale optimization algorithm // Advances in engineering software. 2016. 95. 51–67.

Mohanty Soumya D. Particle Swarm Optimization and regression analysis–I // Astronomical Review. 2012. 7, 2. 29–35.

Rothwell Peter M. Subgroup analysis in randomised controlled trials: importance, indications, and interpretation // The Lancet. 2005. 365, 9454. 176–186.

Shen Juan, He Xuming. Inference for subgroup analysis with a structured logistic-normal mixture model // Journal of the American Statistical Association. 2015. 110, 509. 303–312.

Shen Juan, Qu Annie. Subgroup analysis based on structured mixed-effects models for longitudinal data // Journal of Biopharmaceutical Statistics. 2020. 30, 4. 607–622.

Shin Ju-Young, Heo Jun-Haeng, Jeong Changsam, Lee Taesam. Meta-heuristic maximum likelihood parameter estimation of the mixture normal distribution for hydro-meteorological variables // Stochastic Environmental Research and Risk Assessment. 2014. 28, 2. 347–358.

Storn Rainer. On the usage of differential evolution for function optimization // Proceedings of north american fuzzy information processing. 1996. 519–523.

Wang Jingli, Li Jialiang, Li Yaguang, Wong Weng Kee. A model-based multithreshold method for subgroup identification // Statistics in Medicine. 2019. 38, 14. 2605–2631.

Wu Ruo-fan, Zheng Ming, Yu Wen. Subgroup analysis with time-to-event data under a logistic-Cox mixture model // Scandinavian journal of statistics. 2016. 43, 3. 863–878.

Xu Weinan, Wong Weng Kee, Tan Kay Chen, Xu Jian-Xin. Finding high-dimensional D-optimal designs for logistic models via differential evolution // IEEE access. 2019. 7. 7133–7146.