Interpretable Configurational Regression (ICRegress)

Christopher Swader

2023-10-11

Introduction

This vignette has two goals:

- To introduce the use cases and value of the "Interpretable Configurational Regression" analytical method.
- To explain how the method can be implemented with the ICRegress package.

Interpretable Configurational Regression (ICR) uses a Machine Learning (genetic optimization) algorithm in order to separate the overall population into different model-specific subgroups. It thus allows for a more case-sensitive assignment of cases to models within the regression paradigm.

In many fields, the underlying reality of cases/populations is more complex than the typical one-model-fits-all regression approach assumes (see Ragin 2006). For modeling a particular outcome (dependent variable), ICR identifies the most relevant subgroups and their separate models.

For policy and diagnostic decisions, for instance in the social sciences, marketing, or epidemiology, a standard regression approach risks applying the wrong model to underlying subgroups where that model may be completely wrong. In other words, important groups of cases may systematically hide beneath the 'average-net-effects' reported in the standard one-population unitary regression model.

In contrast, ICR starts with the assumption that the population may be composed of different subgroups (represented by different configurations of variables) which are differentiated by the fact that they share a different underlying model of explanation for a given dependent variable.

Given the accessibility of modern computational power and algorithms, there is no longer a good reason to start with the assumption that 'one model fits all' in many fields. However, previous methods for modeling data with underlying model-differentiated subgroups in mind have been inadequate for fields where the substantive interpretation of the results is crucial. Both the use of complex interaction terms and many machine learning approaches do not produce results that are readily interpretable (see Molnar et al. 2020). ICR also addresses this problem.

The value of Interpretable Configurational Regression (ICR)

Why is ICR needed?

The 'standard regression approach' (defined here as applying a single regression model to one overall population) systematically produces incorrect results when there are meaningful underlying subgroups in respect to predicting an given dependent variable if those subgroups are not properly taken into account.

Figure 1. The Problem with Standard Regression
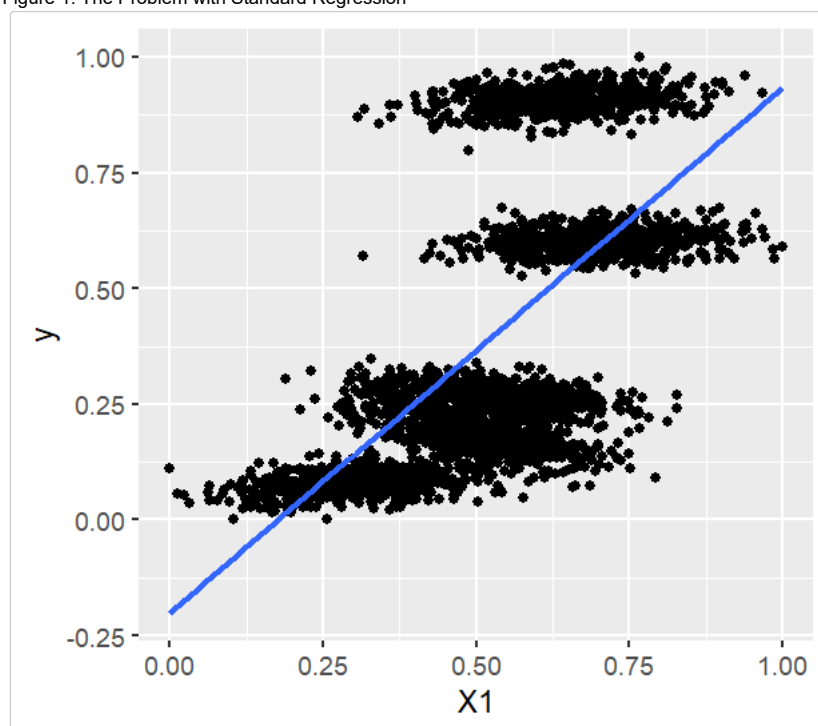
The above Figure 1 illustrates this problem. There are clearly at least 4 (if not 5) separate subgroups, which are 'hiding'

beneath the standard regression model. Notice that this model is completely wrong for all of the subgroups, as it provides the incorrect direction and strength of the effect for each of them.

The problem is easy to see in the above example because (a) there is only one independent variable, and we can see it on a scatterplot (b) these are fake data, so we knew the pattern was there in advance (c) these data are not 'messy' (the errors are relevatively small) like real world data in many fields (such as the social sciences). Messy data would make the problem more apparent.

In using single regression models to describe entire data sets, even when we have interaction terms included, we are missing this information, and replacing it with a inaccuracies or even myths.

Of course, we can attempt to avoid this problem through severak approaches, such as using interaction terms. There are several problems with this approach: - Finding complex, multi-dimensional interaction terms (the same thing is complex, multidimensional subgroups) becomes more difficult (computationally) as the number of interacted variables increases. - Even if we find the relevant interactions, interpreting them (in fields where this is important, such as the social sciences and others) is challenging as interactions produce also many irrelevant combinations of variables that also need to be deciphered and explained.

As such, the goal of ICR is to produce an output of only model-relevant subgroups and their attached models in a way that underlying subgroups are sometimes assymmetrical. This assymmetry greatly increases the interpretability of the results, as we do not need to waste time interpreting model-irrelevant variables.

Figure 2. The Outcome of ICR

Figure 2 illustrates the intended outcome of ICR as applied to the standard regression example in Figure 1. Five subgroups are found, each with their own fully independent regression model. Further, those groups are asymmetrical, because different variables may matter for different subgroups. For example, in the above theoretical example, migrant status matters for differentiating women but not men in terms of the relationship of a given set of independent variables (here, just X1) on a given dependent variable Y.

When should ICR be used?

ICR should be used in settings where there is reason to believe that independent variables do not impact the dependent variable in a merely additive way across the whole sample. ICR is relevant for every situation beyond a simple interaction between two dummy variables where the relevance of any other interaction terms is explicitly excluded,. In other words, if multiple interaction effects are supposed or theoretically possible, ICR can be used to find and explain them.

ICR works by simultaneously finding model-relevant clusters and their underlying models. These clusters/subgroups are homogeneous across the model-relevant independent variables but may differ across model-irrelevant independent variables. This approach maximizes interpretability usually at the expense of empirical explanatory power (as measured, e.g., by R-squared) compared to other machine learning methods. However, it should be noted that ICR's explained variance (R-squared) is usually equal or superior to standard regression.

What is meant by the model-relevant clustering that ICR uses? Figure 3 illustrates the theoretical difference between standard k-means clustering, which is model agnostic, and ICR. The criterium for creating k-means clusters is minimizing the distance to the cluster centroids. ICR has a different criterium for its clusters, one that essentially maximizes the difference between the separate regression models that correspond to each cluster. This criterium can result in very different clusters than k-means, represented in the figure by the different cluster spaces occupied by the three models (M1, M2, M3), because the ICR clusters only take into account those variables that are relevant which when split results in different models.

Figure 3. Standard Clustering versus ICR Clustering
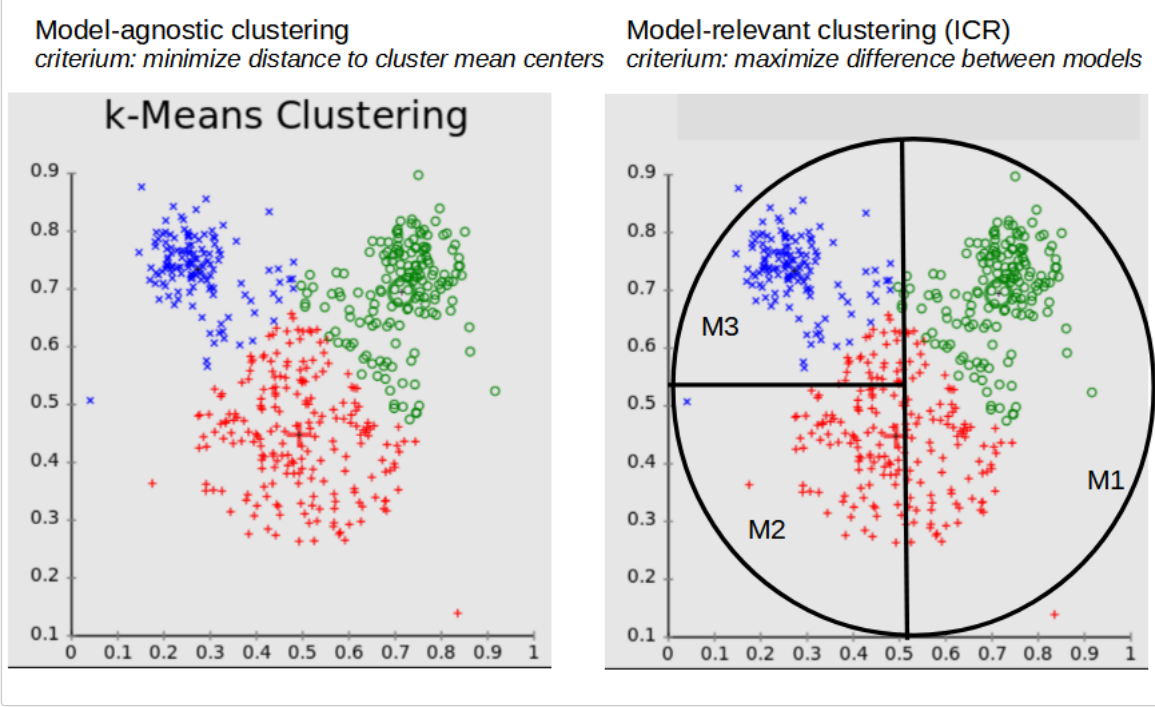
Does it work? How does ICR compare to other approaches?

ICR is methodologically and empirically contrasted with some other methods.

Methodological comparision of ICR to other approaches

The following table displays how ICR compares with standard regression plus interaction terms, clusters plus regression, latent class analysis plus regression, and simple regression.

To clarify, 'clusters plus regression' implies running a standard cluster analysis (e.g. k-means) and then computing separate regression models for each of the identified clusters for a given dependent variable. Similarly, 'latent class plus regression' implies finding latent classes in the existing data, and applying separate regression models for those latent classes for a given dependent variable. This method is also called a regression 'mixture model' (not to be confused with mixed-effects models). Another relevant method is MAIHDA (Merlo 2018), whereby a hierarchical model is used to apply interaction classes at the second level of analysis. That model is not tested here, but I assume it functions similarly to the standard regression plus interactions approach.

Table 1. ICR Methodological Comparison to Other Approaches

	ICR	**Mixture Regression**	**Cluster analysis + models**	**St**
Subgroup homogeneity	Partially homogenous subgroups (homogenous only on model-relevant variables)	Heterogenous subgroups (like LCA)	Fully homogenous subgroups (also on model-irrelevant combinations)	Fully ho mod
Model relevance of subgroups	Subgroups are model-relevant	Subgroups are model-agnostic	Subgroups are model-agnostic	Subgrou (not a
Degree of model differentiation	Fully separate models for subgroups	Fully separate models for subgroups	Fully separate models for subgroups	
Explanatory power Weighted R2	mid to high	high	mid	
Interpretability	high	low	mid	

In summary the unique defining characterstics of ICR are that its subgroups are partially homogenous (only across model-relevant variables), its subgroup models are completely independent from one another, and it provides a degree of explanatory power at least as good as standard regression while providing a degree of case-sensitivity and interpretability that exceed other approaches.

Empirical comparision of ICR to other approaches

Given the above claims, how does ICR compare with other approaches within simulations, where a ground truth is actually known?

Benchmarks are used to evaluate four approaches: ICR, standard (k-means) clusters plus regression, latent classes plus regression (mixture models), and standard regression (one model fits all, no interaction terms).

Three relevant benchmarks are used to compare performance of these approaches: -1. R-squared. For multiple regression models, a weighted R-squared is used, where the weights correspond to the number of cases belonging to each model-specific subgroup. -2. case-assignment score This is defined as the proportion of cases found together in the true subgroups that remain together in the subgroups identified by ICR, with a penalty added for the extent to which the number of discovered models differs from the number of ground truth models (otherwise, a one model solution would always have a perfect case assignment, since all original cases would end up in the same final model). Better performance is indicated by a high score. -3. model similarity error. This is defined as the summed squared differences of the ICR-identified models, weighted by their case proportions, compared to the ground truth models. Better performance is indicated by a low model similarity error.

These benchmarks were tested across these approaches using the parameters in the following code segment (one can replicate this using the benchmarking() function in the ICRegress package).

Note that the compared methods (with the exception of standard regression of course) were always forced to use the same number of models as ICR, so that the results are directly comparable. For example, if ICR identifies 3 models, clusters plus regression and mixture regression were each run with 3 clusters/latent classes as well.

NOTE ABOUT TUNING: See the later vignette section on tuning. It is necessary to tune a given data and icr configuration in order to get sensible results, also for benchmarking.

```r
library(ICRegress)

b_results <- benchmarking(random_seeds=as.integer( c(100001:100050)),
                          generations=c(1, 50, 100, 200, 400, 800, 1600),
                          solution_thresh = c( .01 ),
                          n_strands = c(50),
                          real_models = c( 20),
                          error_sd = c(.05),
                          n_IVs=c(30),
                          n_cases= 5000, #n rows in dataset
                          test_set_model_closeness= c("residuals"),
                          parallelize=F,
                          detailed_plots_on=F,
                          boxplots=T,
                          starting_population=c(500),
                          hold_back_cores=4,
                          mutation_rate = .1,
                          n_children = 2,
                          nelitism=3,
                          death_by_ageing=40)
```
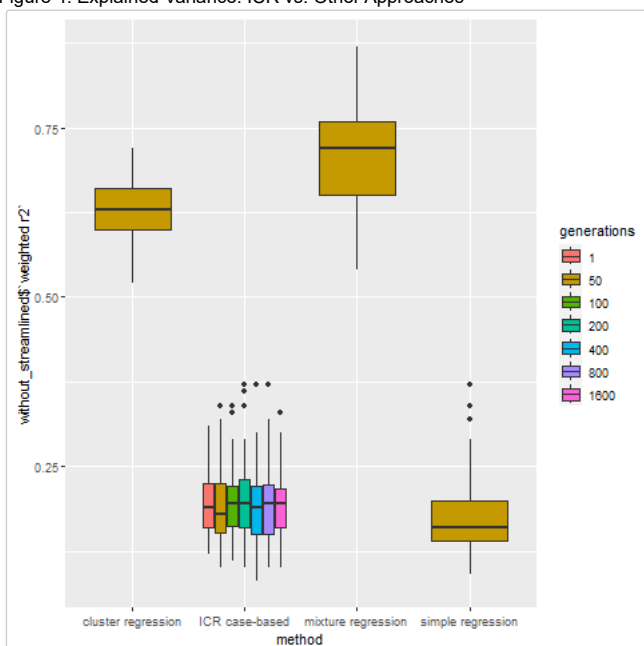
The saved benchmarking results can be found here within the ICRegress package:

```r
data("benchmark_results")
```

Explained Variance

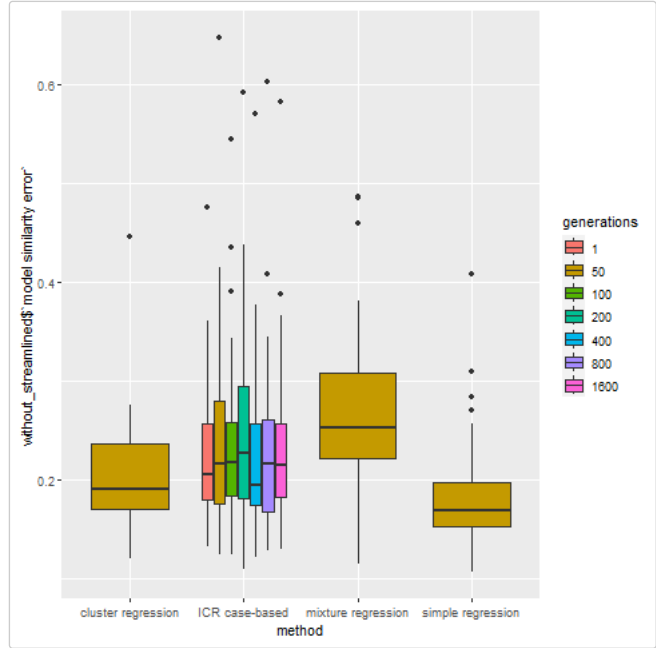Figure 4. Explained Variance: ICR vs. Other Approaches

Note: The number of 'generations' annotated in these charts is only relevant for the ICR method.

In terms of explained variance, ICR outperforms standard regression, but is superceded by both clusters and latent classes with regression.

Model Error

Figure 5. Model Error: ICR vs. Other Approaches

In terms of model error, where a lower score represents better performance, ICR performs the same as clusters plus regression, somewhat better than mixure regression, and somewhat worse than simple regression. This represents the accuracy of the underlying models across comparable techniques.

Case Accuracy

Figure 6. Case Accuracy: ICR vs. Other Approaches
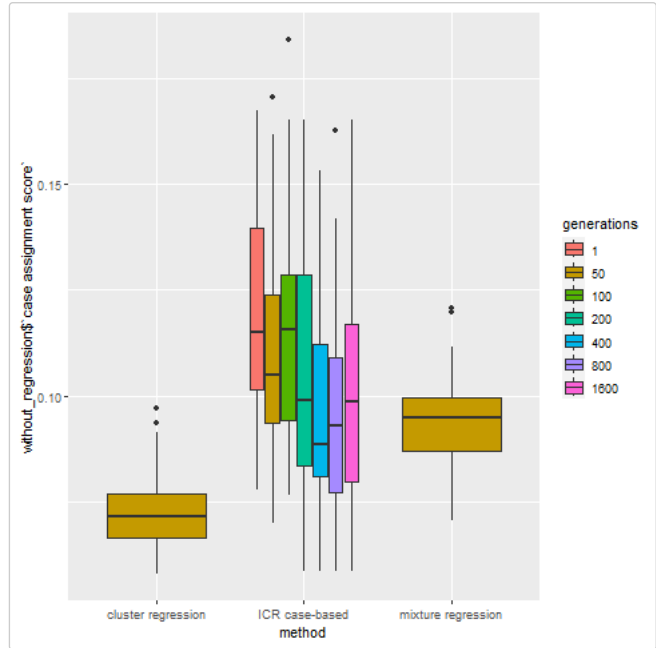
In terms of case, assignment accuracy, ICR outperforms mixture regression, which in turn outperforms clusters plus regression.

Speed

In terms of the time it takes for ICR to run, it has significantly higher time complexity than the other methods, because of the genetic algorithm it uses. Therefore a tradeoff for using this method is the time required to train the genetic algorithm. However, our analysis suggests that ICR already outperforms other methods substantively with a training period of 800 to 1000 generations (around 10 minutes), and 3200 generations (40-45 minutes) is enough to reach a plateau, putting ICR results substantively above the other quicker methods. While slower than other techniques, 45 minutes (or 2 hours with more survivors each generation) is hardly a long time compared to the considerable amounts of time that go into data cleaning, wrangling, and other forms of preparation, if the payoff, greater case sensitivity, is worthwhile.

Summary of Performance

Overall findings are that ICR beats other methods when it comes to its design intention: a low case assignment error. In other words, ICR is much less likely to assign cases to an incorrect model and maintains a high case sensitivity, which is important for conditions where predictions are needed for specific cases. Further ICR does as well as simple regression on the underlying correctness of models and is better for explained variance. However, when high explained variance is desired instead of case sensitivity, mixture regression (or alternative

machine learning approaches) is preferable. Further, ICR is slower than the alternative techniques, because the genetic algorithm takes time to find ideal subgroups and their attached models. However, for most use cases spending one to two hours to run a full analysis with ICR constitutes only a tiny proportion of the overall analysis time.

How does ICR work?

Summarized steps of the algorithm:

-1. Prepare data: User chooses the outcome variable and only the theoretically relevant independent variables in order to allow for interpretability of final results. ICRegress rescales all variables from 0 to 1. Training and test sets are separated with a 60-40 ratio.

-2. Create population for genetic algorithm: ICRegress creates a random starting population (coefficients) of a given size.

-3. Natural selection: each member of the population (set of coefficients) is multiplied against the corresponding data values for each training case. The summed absolute difference between the actual outcome and this prediction is recorded.

-4. Evolve: The algorithm counts how many cases are discretely 'solved' (e.g. within a target threshold) by each set of coefficients. Crucial is that those solutions are heavily weighted by their uniqueness, to encourage the model population to solve different niches of the data. We thus get a rating of each member of the population that is based on the unique proportions it has solved.

-5. 'Death': Those sets of coefficients that did not solve above a particular threshold die. Those that survived have a chance to 'mate'.

-6. Mating and mutation: Each surviving population member is paired with another. They share a random half of their coefficients for each 'child' they create (a new set of coefficients). Mutations may change a portion of that child's coefficients. After mating, some members of the population may die from old age.

-7. Generations: The same process repeats until a goal is reached (# of generations or % of cases solved). Currently this is tied to an explicit number of generations.

-8. Condensation: Many of the final surviving models (sets of coefficients) are related. This is fine for prediction purposes. But we need a parsimonious solution. We therefore condense into an interpretable number (currently between 2 and 6) of fewer models (using different clustering techniques).

-9. Streamlining: The models' coefficients still have some randomness in them due to the genetic algorithm that generated them. We streamline them by cleanly separating the subgroups and applying true linear regressions to them.

-10. Testset: The condensed and streamlined models are applied to the test set. Cross-validation is not needed, as the the procedure removes overfitting tendencies through condensation and streamlining.

-11. Case-model tree: A simplified decision tree is implemented for predicting which cases get which models. However, this agglomerative tree is built from the bottom up (from the leaves to the trunk), ensuring that all models have one and only one leaf. This agglomerative tree is initially population based, reflecting its heterogeneous subgroups. The population-based tree is then converted into a case-based tree to create homogeneous subgroups to aid in case assignment and interpretability. In other words, differences between subgroup means of a given variable in the initial population-based tree are used to create a case-based rule that assigns all cases homogeneously to the group either above or below the threshold for the case-based tree.

-12. Interpretation: ICR results are interpreted by first describing the homogeneous subgroup (e.g. "migrant men") and then the simplified model (only substantively significant coefficients for that subgroup's model). The process is iterated for each subgroup, highlighting key substantive differences and similarities in the models for each subgroup and also the different baselines for the dependent variable.

How to use ICR with the ICRegress package?

Example using real data

The use of the ICR technique will also be demonstrated using the ICRegress package.

The data used for this example are from the 2016 wave of the European Social Survey:

- ESS Round 8: European Social Survey Round 8 Data (2016). Data file edition 2.2. Sikt - Norwegian Agency for Shared Services in Education and Research, Norway – Data Archive and distributor of ESS data for ESS ERIC. doi:10.21338/NSD-ESS8-2016.

Loading data

Load ess2016 data included in the ICRegress package. These are individual-level data nested in countries. The country variable used in ICRegress is a wider country grouping (e.g. Southern or Western Europe) to aid in interptation.

```
data("ess2016")
```

Selecting Independent Variables

We wish to explain/predict happiness in Europe based on seven independent variables in this mini data-set: female, age, country, (generalized) trust, frequency of social meetings, income, and subjective health.

-Note that each of these independent variables has a known theoretical and empirical relationship to happiness! I will not repeat this theoretical and aggregated empirical knowledge here, but rest assured that experts on happiness and subjective well-being should not have too much trouble interpreting the results. Because the purpose of this method is to provide more nuanced, case-specific results that are *interpretable*, it is advised not to use the 'kitchen sink' (include all available features) method of variable selection with ICR, or you will likely not be able to make sense of the results, thus defeating the purpose.

Tuning

ICR is a machine learning algorithm that needs to be tuned for each use case to achieve meaningful results.

The tuning goal is to *start* in the first generation with a solved discrete solution percentage that is low, perhaps 30 to 50%, and allow the solution percentage to *grow* to a significantly higher percentage (ideally above 70 or 80%) through training. This means that the final results are due to learning the data patterns rather than through random chance. The target growth between the initial and final solved solution percentage should be at least 20 percent, and ideally higher.

The primary way to tune this is through adjusting the solution threshold (where a larger threshold will always yield a higher solution percentage) and the mutation rate (which will increase or decrease the variability between generations). The number of children is another way to balance the algorithm's ability to learn over time. A further way for the algorithm to map complex problems and cases where it gets 'stuck' at a local optimum is to increase n_strands in order to allow more survivors from generation to generation, thus allowing for more diversity to be preserved and more niches to be filled (this option will significantly increase run time and also greatly increase the starting solution %, which means the solution threshold should be reduced to allow for learning.). Hard cases will also benefit from having a larger starting population (which will also increase the starting discrete percentage solved, which means the solution threshold should be made smaller to compensate and allow for learning).

A good starting practice is to find a set of tuning parameters that allow you to achieve 20% growth in discrete solution percentage within the first 100 generations. Then reduce the mutation rate slightly, and train for at least 800 to 1000 generations (or until the discrete solution grows to a high percentage).

The below parameters for this problem allowed ICR to learn from 38% or so in the first generation to 99% in the final generation. (In this use case, I achieved this by increasing the n_strands, the survivors each generation, and decreasing the mutation rate, and then diminishing the solution threshold so that it resulted in a smaller starting solved percentage.)

```r
set.seed(1234)
icr_results <- icr(generations=400, #1000 as default
                   input_data=ess2016,
                   dv="happy",
                   iv=c("female", "age", "cntry", "ppltrst", "sclmeet",      "income", "health"),
                   starting_pop=10000, #1000
                   n_strands = 100,
                   mutation_rate = .02,
                   solution_thresh = .003,
                   n_children=3,
                   death_by_ageing =60,
                   nelitism=10)
```

Interpreting the agglomerative model tree and models

The output of the icr() function is fairly comprehensive. You may view the models into which the data have been split into corresponding subgroups. The first column is the intercept corresponding to the depending variable. The further columns are the coefficients for those corresponding independent variables (reference groups are the first categories of categorical variables with 3 or more categories).
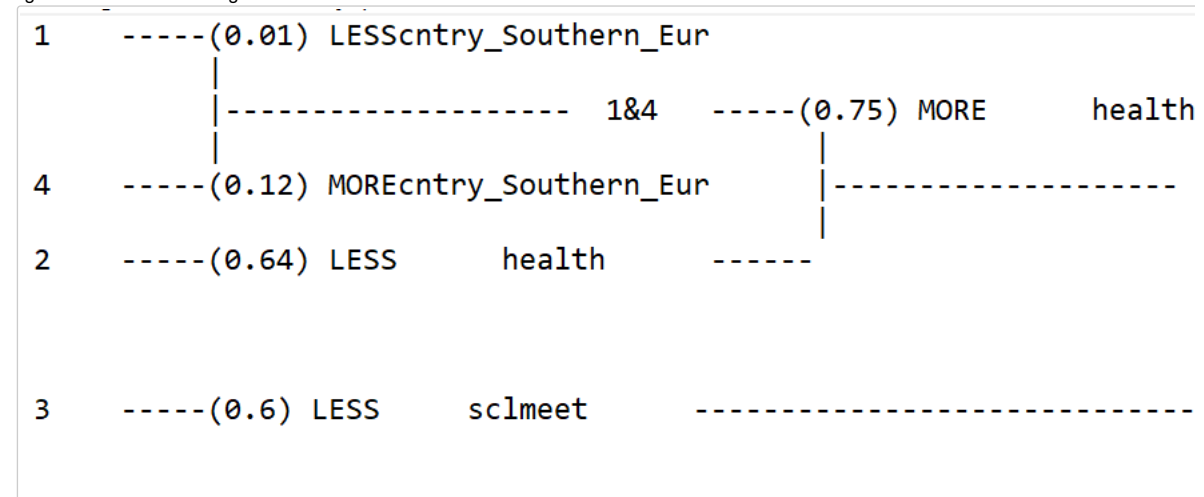
```r
icr_results$models
```

You may also view your original input data with the new models assigned in a "model" column.

```r
head(icr_results$original_data_with_new_models)
```

Next arises the important question for interpretation: how are the cases systematically assigned to their models? icr() has saved a file called agglomeration_tree.txt in your working directory. You may open it in order to visualization the agglomeration tree.

Figure 7. Case-Model Assignment Tree

```
1      -----(0.01) LESScntry_Southern_Eur
              |
              |------------------- 1&4    -----(0.75) MORE      health
              |                                |
4      -----(0.12) MOREcntry_Southern_Eur      |-------------------
                                               |
2      -----(0.64) LESS      health      ------


3      -----(0.6) LESS      sclmeet      ---------------------------
```

This tree starts with all model subgroups combined on the right in the trunk (the overall sample) and proceeds to the left by introducing model-relevant subgroup splits. These splits are *originally* made based on the initial heterogenous subgroups formed before the icr algorithm was complete. As a result, the numbers reported are the original subgroup means. Splits are introduced between them.

For example, the first split was in the frequency of social meetings, and the larger group (3 final models) had MORE social meetings, a larger mean of .65, compared to a smaller group (1 final model) with an original mean of .6. Through a similar process of splits, we reach the final leaves, with are composed of homogeneous subgroups belonging to one model.

Note that these splits are converted by icr() into case-based rules, which means that ALL cases with values above or below the split are classified cleanly into their end models. In the case of dichotomous variables, this means the split cleanly separates the groups into their two possible values.

To continue with the above example, this means that cases with more social meetings and less health are assigned to model 2. Cases with more health are assigned to model 4 if they are Southern Europeans and model 1 if from any other European region. Finally, those cases with fewer social meetings are assigned to model 3.

These case-based model assignment rules can also be viewed by looking directly at the 'cutlist' produced by icr(). The cutlist information is exactly identical to their visualization in the agglomeration tree. One can read both from the leaves toward the trunk (top to down in the cutlist) or more easily (for interpretative purposes) from the trunk to the leaves (bottom to top in the cutlist).

```
icr_results$cutlist
```

The cutlist can be read trunk-to-leaves by starting at the bottom, where all subgroups are combined. The last row shows that they are split into the a group of 3 models (leaf2) and 1 model (leaf1), and the split is based on social meetings, with a threshold between the leaf1 and leaf2 relation values.

We continue by reading the cut the second from the bottom, where see that the models 1 and 4 are split from model 2 based on health.

The final cut was made between models 1 and 4 based on being Southern European (or not).

We can also look at the model_subgroup_profiles, which are the means and standard deviations of the subgroups for each model.

```
icr_results$model_subgroup_profiles
```

These profiles can be used to distinguish subgroups according to all variables. Especially the dependent variable can be informative. In this case, the more healthy subgroups 1 and 4 have a higher average happiness than the less healthy groups 2 and 3.

Substantative Interpretation

The substantive interpretation of these results, combining information from the the case-model, the subgroup profiles, and the models themselves,is that there are four important subgroups that have different models for happiness:

-Group 3. *People with fewer social meetings* have a lower baseline of happiness and gain the most (happiness) from increased generalized trust and social meetings, and they also gain substantially from health.

-Group 2. *People with more social meetings but poorer health* have a somewhat lower baseline happiness, gain more than any group from increased health, and also gain happiness substantially from higher generalized trust and social meetings.

-Group 4. *People with more social meetings, better health, from Southern Europe* have a higher happiness baseline but gain less from generalized trust, social meetings, and health compared to other groups.

-Group 1. *People with more social meetings, better health, from another European society* also have a high happiness baseline, and they gain more than Southern Europeans from generalized trust, social meetings, and health.

Of course, more detailed descriptions of each model can be made, to include a quantification of the effects and reporting of also the smaller effect sizes, as desired. The above condensed interpretation is offered in order to

demonstrate that ICR can provide a meaningful interpretation of results that are more case-sensitive/accurate than a simple regression model.

Issues and Current Limitations

-It is yet unknown what the ideal number of generations for training is. While more generations tends to yield better results, we do not yet know at which point there is a plateau in terms of results. Of course, this is a question that also depends upon revealing ideal tuning configurations. This is also in development.

-ICRegress, in development, is initially only programmed to handle only continuous dependent variables. Full functionality for other dependent variable forms will be added.

-A built-in limitation of the method stems from its interpretability advantage. The agglomerative decision tree used to explain the differences between the subgroups of the models also means that the number of variable splits used to explain the subgroups involved is always m - 1, where m equals the number of ICR-identified distinct models (and their subgroups).

Acknowledgments

References

Merlo, J.. 2018. "Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy (MAIHDA) within an Intersectional Framework". Social Science & Medicine 203 (april): 74–80. https://doi.org/10.1016/j.socscimed.2017.12.026.

Molnar, C., C. Casalicchio, and B. Bischl. 2020. "Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges." ArXiv:2010.09337 [Cs, Stat], October. http://arxiv.org/abs/2010.09337.

Ragin, C. C. 2006. "The Limitations of Net-Effects Thinking." In Innovative Comparative Methods for Policy Analysis, edited by Benoît Rihoux and Heike Grimm, 13–41. Boston: Kluwer Academic Publishers. https://doi.org/10.1007/0-387-28829-5_2.