Quick Start: ICRegress Package: DRAFT VERSION

Christopher Swader

2024-04-23

Introduction

This quick start guide will explain how the configurational regression 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 sample into different model-specific subgroups. It thus allows for a more case-sensitive assignment of cases to models within the regression paradigm.

When should ICR be used?

ICR or comparatable methods 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.

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 (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.

####Automatic Tuning It is recommended that you tune using the autotune() function and then plug that function's output (parameters) into the icr() function. It is also recommended that you use the same random seed for tuning as you use for the icr() function itself.

For best results with large datasets, it is recommended to let autotune() run overnight with restrict_sample equal to False. If time is limited, however, you may turn restrict_sample to True, which generates parameters based on a small random subset of the overall sample, and obtain decent parameter values in a much shorter period of time.

Because of the time required for tuning, it is recommended that you save your tuning results so that you can quickly re-run training using the same parameters without returning (for example by using saveRDS()).

```
autotune_results <- autotune(ess2016, dv="happy", iv=c("female", "age", "cntry", "ppltrst", "sclmeet", "income", "health"),
                             restrict_sample = F,
                             random_seed=1234)
```

Running main algorithm

Apply the tuning results as parameters in order to run then icr algorithm. Note that applying any autotune parameters to this function will fully override any manual inputs. If you want to instead manually tune, then autotune_params should be set to NULL.

Set the number of generations so that by the end you have at some point achieved significant growth from the initial solution and there is at least 80 percent ‘ecologically solved’ at some point during the training. 400 to 800 generations is usually enough for such an outcome. icr() will automatically compute its results based on the best results ever achieved in the total run of generations, so one does not need to worry if the very last generation was the best.

```
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,
                   force_subgroup_n = NA,
                   re_use_variables=F,
                   autotune_params = autotune_results) #or NULL if manually tuned
```

Interpretation

The full vignette pdf has instructions for interpreting the results in the model object returned by the icr() function.

Acknowledgments

Linus Covic contributed substantially to the project with code and ideas from September to October 2022.