The Journal of Open Source Software

ambrosia: An R package for calculating and analyzing food demand that is responsive to changing incomes and prices

Kanishka Narayan[1], **Chris R. Vernon**[1], **Stephanie Waldhoff**[1], **James A. Edmonds**[1], **and Ryna Cui**[2]

1 Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, MD, USA **2** University of Maryland

DOI: 10.21105/joss.02890

Software

- Review
- Repository
- Archive

Editor: David Hagan

Reviewers:

- @tscheypidi
- @jhollist

Submitted: 15 October 2020
Published: 26 March 2021

Summary

The `ambrosia` R package was developed to calculate food demand for staples and non-staple commodities that is responsive to changing levels of incomes and prices. `ambrosia` implements the framework to quantify food demand as established by Edmonds et al. (2017) and allows the user to explore and estimate different variables related to the food demand system. Currently `ambrosia` provides three main functions:

(1) calculation of food demand for any given set of income levels and prices;
(2) estimation of calibration parameters within a given a dataset. Note: `ambrosia` is used to calculate the calibration parameters for the food demand model implemented in the Global Change Analysis Model (GCAM; Calvin et al., 2019);
(3) exploration and preparation of raw data before starting a calibration parameter estimation.

Statement of need

An important motivation to develop `ambrosia` is functionalizing and separating out the different components of the sophisticated food demand framework from Edmonds et al. (2017) (summarized below) into usable R functions that can be easily parameterized and customized by the user. Thus, `ambrosia` has been developed to help researchers explore questions related to trends in food demand empirically. Since the equations of the model are grounded in peer reviewed research while the code itself is written in R (which increases usability), the tool is useful to researchers interested in,

1) analyzing and exploring trends in food demand with a computational model that is responsive to changes in incomes and prices that can easily be implemented on any time series (dataset);
2) re-estimating calibration parameters of the food demand model using custom data, thus effectively allowing the user to calibrate the model to custom data;
3) incorporating a detailed food demand model in their own earth system and economic models.

`ambrosia` is part of an ecosystem of tools within the Global Change Intersectoral Modeling System (GCIMS) that help users computationally explore science and policy questions related to different dimensions of human-Earth systems (PNNL, 2020). The parameters calculated from `ambrosia` are utilized directly in GCAM (Calvin et al., 2019) to represent forecasts of food demand. `ambrosia` ensures that the parameters that are used within GCAM are scientifically and empirically sound and also ensures reproducibility of the parameters for validation to comply with the commitment of GCIMS to FAIR guiding principles for scientific data management (Wilkinson et al., 2016). The code is structured to ensure that the parameters can be updated and tested effectively with changes to the underlying data.

Thus, the tool not only enables easy use and future development, but also enables easy modularization of the code within other systems. The sections below contain a detailed discussion of the different functions and customization options available within the tool.

Summary of the Edmonds et al. framework

The Edmonds et al. (2017) model represents a food demand model for staples and non-staple commodities at different levels of prices and incomes. Demand for staples is described as increasing when income is lower, eventually peaks at under 1000$ per person per capita, and then begins to decline as higher income ranges are approached. Demand for non-staples increases with income over all income ranges; however, total (staple + non-staple) demand saturates at high income level.

The Edmonds et al. (2017) approach uses 11 calibration parameters where the parameters are fit using pooled cross-sectional-timeseries observations and a Bayesian Markov Chain Monte Carlo method (MCMC; Hastings, 1988). The framework represents demand for three categories of goods: staples (s), non-staples (n) and materials (m) where materials represent everything in the economy other than staples and non-staple food commodities. The demand for these three categories changes with changes in income (Y) and prices (P), with the response to price changes varying with income. Expenditures on these three goods are assumed to exhaust income.

Demand for these three categories can be represented mathematically as,

(1) Staple demand: $q_s = A_s(x^{h_s(x)})(w_s^{e_{ss}(x)})(w_n^{e_{sn}(x)})$

(2) Non-staple food demand: $q_n = A_n(x^{h_n(x)})(w_s^{e_{ns}(x)})(w_n^{e_{nn}(x)})$

(3) Materials demand : $q_m = x - w_s q_s - w_n q_n$

where w_i is P_i/P_m, x is Y/P_m and A_i are constants.

e_{ij} is defined in a general way,

(4) $e_{ij}(x) = g_{ij} * f_i(x)\alpha_j$

where $g_{i,j}$ are constants, $i >= j$ and $f_i(x) = (\delta ln(x^{h_i(x)}))/(\delta ln(x))$.

If h and e were constants, h would be an income elasticity as $x = Y/P_m$ and e_{ij} would be own and cross price elasticity as $w_i = P_i/P_m$.

The following functional forms are chosen for h_s and h_n,

(5) $h_s(x) = (\lambda/x)(1 + (\kappa/ln(x)))$

(6) $h_n(x) = \nu/(1 - x)$.

In addition to the above, two other scaling parameters are applied when normalizing the demand values to that of materials. These are $psscl$ for staples and $pnscl$ for non-staples.

The parameters are fit using a weighted least square log likelihood function (Caroll & Ruppert, 1988) described below.

(7) $ln(L) = \Sigma_{i=1}^{N}(w_i(y_i - \hat{y}_i)^2)/2\sigma^2$

where, y_i is the ith data value and $\hat{y}_i$ is the corresponding model output and w_i is the weight assigned to the data point. Since the parameters were fit based on regional data, the regional population was used as the weight.

By applying the 11 parameters to the equations described above, the user can generate estimates of demand for staples and non-staple commodities in thousand calories across different income levels and prices.

Main functions and customization

The 'ambrosia_vignette provides usable examples for all the major functions within the code.

The ambrosia package can be easily loaded as a standard R package after installation from GitHub. The user can calculate demand for staples and non-staples using the food.dmnd() function. The user will have to pass in a dataset with the price of staples (Ps), price of non-staples (Pn), incomes (Y) (Current income proxy used in ambrosia is GDP per capita in thousand USD). In addition to the dataset, the user must pass a vector of 11 calibration parameters. In order to functionalize the parameters, the code contains a function called vec2param() that will generate a parameter structure that can be used by the food demand function. The food demand function is implemented using equations (1), (2), (3) described above. The user can also calculate and analyze price elasticities using the function calc1eps(). These elasticities are calculated in accordance with equation (4) described above.

An interactive version of the food demand model can be launched by the user through the runapp() function to explore the impact of different parameters.

One of the benefits of using ambrosia is that a user can estimate their own calibration parameters with a custom data set using the log-likelihood maximization approach. To enable this, ambrosia is equipped with a function create.dataset.for.parameter.fit() that will help a user generate a dataset that is appropriate for parameter estimation. The user can re-create the training data used to calculate the parameters for GCAM using the Process_Demand_Data.R under the scripts directory.

Users can complete the parameter estimation on the dataset returned by create.dataset.for.parameter.fit() with a call to the calculate.ambrosia.params() function. This function builds on the Edmonds et al. (2017) approach by maximizing the log-likelihood score using the optim() function. Note that the user can also choose to use a different method (for example , a MCMC) to maximize the log-likelihood function by first setting up the function using the mc.setup() function. The code contains an example of a MCMC implementation in C++ under scripts/cpp.

Acknowledgements

This research was supported by the U.S. Department of Energy, Office of Science, as part of research in MultiSector Dynamics, Earth and Environmental System Modeling Program. The

Pacific Northwest National Laboratory is operated for DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830. The views and opinions expressed in this paper are those of the authors alone.

References

Calvin, K., Patel, P., Clarke, L., Asrar, G., Bond-Lamberty, B., Cui, R. Y., Di Vittorio, A., Dorheim, K., Edmonds, J., Hartin, C., & others. (2019). GCAM v5. 1: Representing the linkages between energy, water, land, climate, and economic systems. *Geoscientific Model Development (Online)*, *12*(PNNL-SA-137098). https://doi.org/10.5194/gmd-12-677-2019

Caroll, R. J., & Ruppert, D. (1988). Transformation and weighting in regression. *CRC Press*, *30*. https://doi.org/10.1201/9780203735268

Edmonds, J. A., Link, R., Waldhoff, S. T., & Cui, R. (2017). A global food demand model for the assessment of complex human-earth systems. *Climate Change Economics*, *8*(04), 1750012. https://doi.org/10.1142/s2010007817500129

Hastings, W. K. (1988). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, *1*. https://doi.org/10.1093/biomet/57.1.97

PNNL. (2020). In *Global Change Intersectoral Modeling System*. Pacific Northwest National Laboratory. https://gcims.pnnl.gov/

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., & others. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 1–9. https://doi.org/10.1038/sdata.2016.18