The Journal of Open Source Software

# hal9001: Scalable highly adaptive lasso regression in R

**Nima S. Hejazi**[1, 4], **Jeremy R. Coyle**[2], **and Mark J. van der Laan**[2, 3, 4]

**1** Graduate Group in Biostatistics, University of California, Berkeley **2** Division of Epidemiology & Biostatistics, School of Public Health, University of California, Berkeley **3** Department of Statistics, University of California, Berkeley **4** Center for Computational Biology, University of California, Berkeley

**DOI:** 10.21105/joss.02526

**Software**

- Review
- Repository
- Archive

**Editor:** Mikkel Meyer Andersen

**Reviewers:**

- @daviddewhurst
- @rrrlw

**Submitted:** 24 June 2020
**Published:** 27 July 2020



## Summary

The `hal9001` R package provides an efficient implementation of the *highly adaptive lasso* (HAL), a flexible nonparametric regression and machine learning algorithm endowed with several theoretically convenient properties. `hal9001` pairs an implementation of this estimator with an array of practical variable selection tools and sensible defaults in order to improve the scalability of the algorithm. By building on existing R packages for lasso regression and leveraging compiled code in key internal functions, the `hal9001` R package provides a family of highly adaptive lasso estimators suitable for use in both modern data analysis tasks and computationally intensive statistics and machine learning research.

## Background

The highly adaptive lasso (HAL) is a nonparametric regression function capable of estimating complex (e.g., possibly infinite-dimensional) functional parameters at a near-parametric $n^{-1/3}$ rate under only relatively mild conditions (Bibaut & van der Laan, 2019; van der Laan, 2017; van der Laan & Bibaut, 2017). HAL requires that the space of the functional parameter be a subset of the set of càdlàg (right-hand continuous with left-hand limits) functions with sectional sectional variation norm bounded by a constant. In contrast to the wealth of data adaptive regression techniques that make strong local smoothness assumptions on the true form of the target functional, HAL regression's assumption of a finite sectional variation norm constitutes only a *global* smoothness assumption, making it a powerful and versatile approach. The `hal9001` package implements a zeroth-order HAL estimator, which constructs and selects (by lasso penalization) a linear combination of indicator basis functions to minimize the loss-specific empirical risk under the constraint that the $L_1$-norm of the vector of coefficients be bounded by a finite constant. Importantly, the estimator is formulated such that this finite constant is the sectional variation norm of the target functional.

Intuitively, construction of a HAL estimator proceeds in two steps. First, a design matrix composed of basis functions is generated based on the available set of covariates. The zeroth-order HAL makes use of indicator basis functions, resulting in a large, sparse matrix with binary entries; higher-order HAL estimators, which replace the use of indicator basis functions with splines, have been formulated but remain unimplemented. This representation of the target functional $f$ in terms of indicator basis functions partitions the support of $f$ into knot points, with indicator basis functions placed over subsets of the sections of $f$. Generally, very many basis functions are created, with an appropriate set of indicator bases then selected through lasso penalization. Thus, the second step of fitting a HAL model is performing $L_1$-penalized regression on the large, sparse design matrix of indicator bases. The selected HAL regression model approximates the sectional variation norm of the target functional as the absolute sum

of the estimated coefficients of indicator basis functions. The $L_1$ penalization parameter $\lambda$ can be data adaptively chosen via a cross-validation selector (van der Laan & Dudoit, 2003; van der Vaart, Dudoit, & van der Laan, 2006); however, alternative selection criteria may be more appropriate when the estimand functional is not the target parameter but instead a nuisance function (e.g., van der Laan, Benkeser, & Cai, 2019; Ertefaie, Hejazi, & van der Laan, 2020).

## `hal9001`'s core functionality

The `hal9001` package, for the `R` language and environment for statistical computing (R Core Team, 2020), aims to provide a scalable implementation of the HAL regression function. To provide a single, unified interface, the principal user-facing function is `fit_hal()`, which, at minimum, requires a matrix of predictors `X` and an outcome `Y`. By default, invocation of `fit_hal()` will build a HAL model using indicator basis functions for up to a limited number of interactions of the variables in `X`, fitting the penalized regression model via the lasso procedure available in the extremely popular `glmnet` `R` package (Friedman, Hastie, & Tibshirani, 2009). As creation of the design matrix of indicator basis functions can be computationally expensive, several helper functions (e.g., `make_design_matrix()`, `make_basis_list()`, `make_copy_map()`) have been written in C++ and integrated into the package via the `Rcpp` framework (Eddelbuettel, 2013; Eddelbuettel et al., 2011). `hal9001` additionally supports the fitting of standard (Gaussian), logistic, and Cox proportional hazards models (argument `family`), including variations that accommodate offsets (argument `offset`) and partially penalized linear models (argument `X_unpenalized`).

Over several years of development and use, it was found that the performance of HAL regression can suffer in high-dimensional settings. To alleviate computational aspects of this issue, several screening and filtering approaches were investigated and implemented. These include screening of variables prior to creating the design matrix and filtering of indicator basis functions (argument `reduce_basis`) as well as early stopping when fitting the sequence of HAL models in $\lambda$. Future software development efforts will continue to improve upon the computational aspects and performance of the HAL regression options supported by `hal9001`. Currently, stable releases of the `hal9001` package are made available on the Comprehensive `R` Archive Network at https://CRAN.R-project.org/package=hal9001, while both stable (branch `master`) and development (branch `devel`) versions of the package are hosted at https://github.com/tlverse/hal9001.

## References

Bibaut, A. F., & van der Laan, M. J. (2019). Fast rates for empirical risk minimization over càdlàg functions with bounded sectional variation norm. *arXiv preprint arXiv:1907.09244*.

Eddelbuettel, D. (2013). *Seamless r and c++ integration with rcpp*. Springer.

Eddelbuettel, D., François, R., Allaire, J., Ushey, K., Kou, Q., Russel, N., Chambers, J., et al. (2011). Rcpp: Seamless r and c++ integration. *Journal of Statistical Software*, *40*(8), 1–18.

Ertefaie, A., Hejazi, N. S., & van der Laan, M. J. (2020). Nonparametric inverse probability weighted estimators based on the highly adaptive lasso. Retrieved from http://arxiv.org/abs/2005.11303

Friedman, J., Hastie, T., & Tibshirani, R. (2009). Glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, *1*(4).

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

van der Laan, M. J. (2017). A generally efficient targeted minimum loss based estimator based on the Highly Adaptive Lasso. *The International Journal of Biostatistics*. doi:10.1515/ijb-2015-0097

van der Laan, M. J., Benkeser, D., & Cai, W. (2019). Efficient estimation of pathwise differentiable target parameters with the undersmoothed highly adaptive lasso. *arXiv preprint arXiv:1908.05607*.

van der Laan, M. J., & Bibaut, A. F. (2017). Uniform consistency of the highly adaptive lasso estimator of infinite-dimensional parameters. *arXiv preprint arXiv:1709.06256*.

van der Laan, M. J., & Dudoit, S. (2003). *Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: finite sample oracle inequalities and examples* (No. 130). Division of Biostatistics, University of California, Berkeley.

van der Vaart, A. W., Dudoit, S., & van der Laan, M. J. (2006). Oracle inequalities for multi-fold cross validation. *Statistics & Decisions*, *24*(3), 351–371.