A Predictive Linear Regression Algorithm for the Neuroimaging Data Model

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

27-07-2021 / 29-07-2021

CITATION

Kumar, Ashmita (2021): A Predictive Linear Regression Algorithm for the Neuroimaging Data Model. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.15062865.v1>

DOI

[10.36227/techrxiv.15062865.v1](https://doi.org/10.36227/techrxiv.15062865.v1)

A Predictive Linear Regression Algorithm for the Neuroimaging Data Model

Ashmita Kumar¹

Albert Crowley²

David B. Keator³

¹Troy High School. Fullerton, CA., USA.

²TCG, Inc., Washington, DC, USA

³University of California, Irvine. Psychiatry and Human Behavior, Irvine, CA., USA.

Introduction

The Neuroimaging Data Model (NIDM)(D. B. Keator et al. 2013; NIDM Working Group n.d.; Maumet et al. 2016) was started by an international team of cognitive scientists, computer scientists and statisticians to develop a data format capable of describing all aspects of the data lifecycle, from raw data through analyses and provenance. NIDM was built on top of the PROV standard(Moreau et al. 2008; “PROV-Overview” n.d.) and consists of three main interconnected specifications: Experiment, Results, and Workflow. These specifications were envisioned to capture information on all aspects of the neuroimaging data lifecycle, using semantic web techniques(“Semantic Web - W3C” n.d.). They provide a critical capability to aid in reproducibility and replication of studies, as well as data discovery in shared resources. The NIDM-Experiment component has been used to describe publicly-available human neuroimaging datasets (e.g. ABIDE(Di Martino et al. 2014), ADHD200(“ADHD200” n.d.), CoRR(Zuo et al. 2014), OpenNeuro(“OpenNeuro” n.d.) datasets) along with providing unambiguous descriptions of the clinical, neuropsychological, and imaging data collected as part of those studies resulting in approximately 4.5 million statements about aspects of these datasets.

PyNIDM(*PyNIDM* n.d.), a toolbox written in Python, supports the creation, manipulation, and query of NIDM documents. It is an open-source project hosted on GitHub and distributed under the Apache License, Version 2.0(“Apache License, Version 2.0” n.d.). PyNIDM is under active development and testing. Tools have been created to support RESTful(Ravan et al. 2020) SPARQL(“SPARQL Query Language for RDF” n.d.) queries of the NIDM documents (i.e. pynidm query) in support of users wanting to identify interesting cohorts across datasets in support of evaluating scientific hypotheses and/or replicating results found in the literature. This query functionality, together with the NIDM document semantics, provides a path for investigators to interrogate datasets, understand what data was collected in those studies, and provide sufficiently-annotated data dictionaries of the variables collected to facilitate transformation and combining of data across studies.

Beyond querying across NIDM documents, some high-level statistical analysis tools are needed to provide investigators with an opportunity to gain more insight into data they may be interested in combining for a complete scientific investigation. Here we report on one such tool providing linear modeling support for NIDM documents (i.e. pynidm linear-regression).

Statement of Need

While tools and libraries for statistics and machine learning algorithms are numerous, there are none that can be directly applied to NIDM documents. The linear regression algorithm presented here allows scientists studying the human brain to find relationships between variables across datasets. The algorithm has the ability to query for specific variables or across similar variables from different studies using concept annotations on variables. It then provides the user with the ability to construct arbitrary linear models on those data, supporting interactions between variables, contrasts of learned parameter sets, and L1 and L2 regularization(Nagpal 2017).

Software Usage and Outputs

The algorithm aggregates and parses data serialized using the standard Terse Resource Description Framework (RDF) Triple Language (TURTLE) ("RDF 1.1 Turtle" n.d.). Researchers have the ability to construct custom models based on their scientific goals. In addition, since the code is in Python, it is flexible and easy to maintain and extend.

Finally, there are existing error checks within the code to make sure the researcher has feedback on why a model cannot run, whether it is because there are not enough data points or because one or more variables could not be found in one or more of the NIDM documents. This makes the experience as simple as possible for the user which is important as our intended audience for these tools are investigators who may have no prior experience with the semantic web and/or NIDM documents.

Thus, the algorithm provides the benefit of a machine learning algorithm that can work on complex datasets written in semantic web while offering optional functionalities such as contrast and regularization. Researchers have the ability to conduct a preliminary analysis to understand if it is worth the effort to pursue combining datasets and doing the transformations necessary to make those datasets scientifically valid to combine. One can quickly determine if there are high-level relationships in the datasets, and look at the different weights to decide what variables to further study.

```
Options:
-nl, --nidm_file_list TEXT  A comma separated list of NIDM files with full
                             path [required]

-contrast, --ctr TEXT       This parameter will show differences in
                             relationship by group (e.g. -contrast
                             age*sex,group). It can be one variable,
                             interacting variables, or multiple

-model, --ml TEXT           This parameter will return the results of the
                             linear regression from all nidm files supplied
                             The way this looks in the command is python3
                             nidm_linreg.py -nl MTdemog_aseg_v2.ttl -model
                             "fs_003343 = age*sex + sex + age + group +
                             age*group + bmi" -contrast group -r L1
                             [required]

-o, --output_file TEXT      Optional output file (TXT) to store results of
                             the linear regression, contrast, and
                             regularization

-r, --regularization TEXT    This parameter will return the results of the
                             linear regression with L1 or L2 regularization
                             depending on the type specified, and the weight
                             with the maximum likelihood solution

--help                      Show this message and exit.
```

Figure 1: pynidm linear-regression parameters

The tool provides a simple command-line user interface (Figure 1) based on the click Python library (“Welcome to Click — Click Documentation (8.0.X)” n.d.) which integrates the linear regression module with existing pynidm tools (e.g. pynidm query, pynidm convert, pynidm visualize). To use the module, the user runs the command pynidm linear-regression with a variety of required and optional parameters.

The first parameter “-nl” is a comma-separated list of NIDM serialized TURTLE files, each representing a single dataset or a collection site within a multi-site research project (Figure 2). A useful set of NIDM files describing publicly-available neuroimaging data from the ABIDE, ADHD200, and CoRR studies along with datasets in the OpenNeuro database can be found on GitHub(D. Keator n.d.).

```
pynidm linear-regression -nl
/simple2_NIDM_examples/datasets.datalad.org/abide/RawDataBIDS/CMU
_a/nidm.ttl,/simple2_NIDM_examples/datasets.datalad.org/abide/Raw
DataBIDS/CMU_b/nidm.ttl -model "fs_000008 = DX_GROUP +
PIQ_tca9ck + http://uri.interlex.org/ilx_0100400" -contrast
DX_GROUP -r L1
```

Figure 2: The full command

The next parameter, “-model” provides the user with the ability to construct a linear model using standard notation found in popular statistics packages (e.g. R statistical software(Ripley 2001)). In the example shown in Figure 2, the model specified establishes the relationship between the dependent variable brain volume and independent variables of the diagnostic group, IQ, and age. In this example, fs_000008 is the unique identifier (UUID) of the supratentorial brain volume computed from the original Magnetic Resonance Imaging (MRI) structural scans of the brain and processed with the FreeSurfer software [REF] as part of the ABIDE study. DX_GROUP is the name of the variable describing the diagnostic group assigned to participants. PIQ_tca9ck is the performance IQ measure collected on study participants. Finally, ilx_0100400 is the age of the participants using a URL form to reference a concept describing the high-level measure of age which has been used to annotate the variables measuring age across studies. This example shows that one can select data elements from the NIDM files for linear regression using three specific forms: (1) using the UUID; (2) using the distinct variable name from the original dataset; (3) using a high-level concept that has been associated with specific variables described by the concept across datasets. We support these three distinct forms of selecting data elements to enable distinct usage patterns. Some investigators will use NIDM documents of their internal studies and want to be able to reference data elements using their study-specific variable names. Other investigators may want to use variables from different studies and thus the variable names are unlikely to be the same; thus, we support the use of selecting variables based on high-level concepts. Categorical and numerical variables are both supported.

```
Treatment (Dummy) Coding: Dummy coding compares each level of the categorical variable to a base reference level. The base reference level is the value of the intercept.
With contrast (treatment coding)

OLS Regression Results
=====
Dep. Variable:      fs_000008      R-squared:      0.242
Model:              OLS           Adj. R-squared: 0.110
Method:             Least Squares  F-statistic:    1.835
Date:               Thu, 08 Jul 2021  Prob (F-statistic): 0.157
Time:               11:40:17       Log-Likelihood: -94.348
No. Observations:   28            AIC:            198.7
Df Residuals:       23            BIC:            205.4
Df Model:           4
Covariance Type:    nonrobust
=====
                    coef    std err          t      P>|t|      [0.025     0.975]
-----
Intercept              8.9220      3.995      2.233    0.036     -0.657     17.187
C(DX_GROUP, Treatment)[T.1]  0.5595      3.007      0.186    0.854     -5.660      6.779
C(DX_GROUP, Treatment)[T.2]  9.0486      9.132      0.991    0.332     -9.842     27.940
ilx_0100400            0.8798      0.443      1.987    0.059     -0.036      1.796
PIQ_tca9ck            -0.1204      0.271     -0.445    0.661     -0.681      0.440
=====
Omnibus:              0.227    Durbin-Watson:      1.829
Prob(Omnibus):        0.893    Jarque-Bera (JB):    0.427
Skew:                 -0.067    Prob(JB):            0.808
Kurtosis:              2.410    Cond. No.             78.1
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Figure 3: Output of command in Figure 2 with treatment coding (contrast using diagnostic group)

The “-contrast” parameter allows one to select one or more independent variables to test the difference among the levels of those categorical variables. The contrast variable in this example is “DX_GROUP” which describes the diagnostic group of each participant in the ABIDE study. The option is provided in order to allow users to make comparisons across multiple coefficients of the linear model. Our tool supports multiple methods of coding treatment variables (e.g. treatment coding (Fig. 3), simple coding, sum coding, backward difference coding, and Helmert coding) as made available by the Patsy Python library (Brooke 1923). The user can select multiple independent variables to contrast and/or contrasts on interactions.

The “-r” parameter allows the user to select L1 (Lasso) or L2 (Ridge) regularization implemented in scikit-learn (Varoquaux et al. 2015). In either case, regularizing prevents the data from being overfit, potentially improving model generalizability and demonstrating which variables have the strongest relationships with the dependent variable.

Conclusions

In this work we have designed a linear regression tool that works on NIDM linked data in support of understanding relationships between variables collected across different research studies. This tool helps scientists evaluate relationships between data at a high level prior to fully integrating datasets for hypothesis testing which may require considerable time and resources. In our initial evaluations, this tool has shown utility for these use-cases. In future work we are creating additional machine learning tools allowing users to cluster data in a similar fashion to the linear regression tool presented here.

Acknowledgements

This work has been supported by the National Institute of Mental Health under grant RF1 MH120021 (PI:Keator), the International Neuroinformatics Coordinating Facility (INCF), and from the National Institute of Biomedical Imaging and Bioengineering P41 EB019936 (PI:Kennedy).

References

“ADHD200.” n.d. Accessed July 8, 2021. http://fcon_1000.projects.nitrc.org/indi/adhd200/.

“Apache License, Version 2.0.” n.d. Accessed July 8, 2021.
<https://www.apache.org/licenses/LICENSE-2.0.html>.

Brooke, Margaret C. 1923. “Patsy.” *The Psychological Clinic* 15 (1-2): 41–43.

Di Martino, A., C-G Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, et al. 2014. “The Autism Brain Imaging Data Exchange: Towards a Large-Scale Evaluation of the Intrinsic Brain Architecture in Autism.” *Molecular Psychiatry* 19 (6): 659–67.

Keator, David. n.d. *simple2_NIDM_examples: ReproNim simple2 NIDM Files for ABIDE/ADHD200 Available from Datalad*. Github. Accessed July 22, 2021.
https://github.com/dbkeator/simple2_NIDM_examples.

- Keator, D. B., K. Helmer, J. Steffener, J. A. Turner, T. G. M. Van Erp, S. Gadde, N. Ashish, G. A. Burns, and B. N. Nichols. 2013. "Towards Structured Sharing of Raw and Derived Neuroimaging Data across Existing Resources." *NeuroImage* 82 (November): 647–61.
- Maumet, Camille, Tibor Auer, Alexander Bowring, Gang Chen, Samir Das, Guillaume Flandin, Satrajit Ghosh, et al. 2016. "Sharing Brain Mapping Statistical Results with the Neuroimaging Data Model." *Scientific Data* 3 (December): 160102.
- Moreau, Luc, Bertram Ludäscher, Ilkay Altintas, Roger S. Barga, Shawn Bowers, Steven Callahan, George Chin, et al. 2008. "Special Issue: The First Provenance Challenge." *Concurrency and Computation: Practice & Experience* 20 (5): 409–18.
- Nagpal, Anuja. 2017. "L1 and L2 Regularization Methods." Towards Data Science. October 13, 2017. <https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c>.
- NIDM Working Group. n.d. "NIDM." Accessed July 8, 2021. <http://nidm.nidash.org/>.
- "OpenNeuro." n.d. Accessed July 8, 2021. <https://openneuro.org/>.
- "PROV-Overview." n.d. Accessed July 8, 2021. <https://www.w3.org/TR/prov-overview/>.
- PyNIDM. n.d. Github. Accessed July 8, 2021. <https://github.com/incf-nidash/PyNIDM>.
- Ravan, Jay, Person, Dave Young, Jason Packer, Chandrajeet Choudhary, Luis, et al. 2020. "What Is REST." May 31, 2020. <https://restfulapi.net/>.
- "RDF 1.1 Turtle." n.d. Accessed July 8, 2021. <https://www.w3.org/TR/turtle/>.
- Ripley, Brian D. 2001. "The R Project in Statistical Computing." *MSOR Connections* 1 (1): 23–25.
- "Semantic Web - W3C." n.d. Accessed July 8, 2021. <https://www.w3.org/standards/semanticweb/>.
- "SPARQL Query Language for RDF." n.d. Accessed July 8, 2021. <https://www.w3.org/TR/rdf-sparql-query/>.
- Varoquaux, G., L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller. 2015. "Scikit-Learn." *GetMobile Mobile Computing and Communications* 19 (1): 29–33.
- "Welcome to Click — Click Documentation (8.0.X)." n.d. Accessed July 8, 2021. <https://click.palletsprojects.com/en/8.0.x/>.
- Zuo, Xi-Nian, Jeffrey S. Anderson, Pierre Bellec, Rasmus M. Birn, Bharat B. Biswal, Janusch Blautzik, John C. S. Breitner, et al. 2014. "An Open Science Resource for Establishing Reliability and Reproducibility in Functional Connectomics." *Scientific Data* 1 (December): 140049.