Name:	Janos G. Hajagos
Affiliation:	Department of Biomedical Informatics Stony Brook University
Email:	janos.hajagos@stonybrook.edu
Presentation type (s):	**Poster**

Mapping Common Data Model data tables to an HDF5 file for reproducible machine learning workflows

Janos G. Hajagos, Ph.D[1]
[1]Stony Brook University, Stony Brook, New York, U.S.A

Abstract

Synthetic inpatient claims data in the OMOP Common Data Model were mapped to matrices in a HDF5 file. The content of the HDF5 were further manipulated to build a matrix for a 30-day post discharge readmission model. A random forest model was built to predict a patient's 30-day readmission risk. While the model's results were not predictive (AUC = 0.53) the modeling approach and pipeline can be applied to data in the Common Data Model (version 5.0). This opens up the possibility of rigorously comparing predictive performance of readmission models across different datasets.

Introduction

One of the first challenges of predictive modeling in health care is the domain knowledge required to construct datasets for model training and testing. The OHDSI project solves this by creating a rich environment for expressing the structure and content of health care data including administrative and EHR (electronic health records) sources.

A second challenge in fitting a machine learning model to health care data is feature engineering and selection. Due to the sparsity of most coded data in health care it is not uncommon to have hundreds to thousands of columns [1,2] in a matrix. HDF5 (Hierarchical Data Format version 5), a cross platform file container for scientific data, can organize matrices into a filesystem like structure and can efficiently store large matrices using compression. HDF5 solves the second challenge by allowing data in different domains (observation, measurement, condition, drug exposure) to be stored in groups. HDF5 datasets or arrays can be read completely or in continuous blocks into working memory.

HDF5 file containers remove dependencies such as configuring database connections and networking. To demonstrate the feasibility of the approach a database of synthetic inpatient visits were mapped into multiple matrices in a HDF5 container using a mapping script. An inpatient 30-day readmission model post discharge was trained and tested against a subset of inpatient stays. The HDF5 file is read and manipulated in Jupyter notebooks using several numerical Python libraries (H5py, scikit-learn, NumPy).

Methods

A 100,000 encounter subset of CMS's SYNPUF (Synthetic Public Use File) and OHDSI vocabulary files were loaded into a PostGreSQL (version 9.6) relational database system. Tables were first denormalized at the visit occurrence level and the results were stored in temporary database tables. A total of 66,700 inpatient visits were extracted from the relational database as a set of JSON (Javascript Object Notation) documents. The JSON documents were mapped to a single HDF5 file. The mapped file contained multiple matrices where each row

represents a separate inpatient visit and a column a feature associated with a domain. Both mapping steps are controlled by separate configuration files. The generated HDF5 file is then post processed and a 30-day readmission flag is appended to the HDF5 file as a separate dataset.

Using a Jupyter/IPython[3] notebook features are selected across multiple groups and assembled into a single matrix (66,700 rows by 5,686 columns). All conditions, observations, procedures, and measurements were included. Additionally, the gender (female), length of stay in days, age in years, and past history of readmissions were included. In total there were 6,241 30-day hospital readmissions. A random forest model was trained to predict 30-day readmission following discharge. To measure performance of the classifier the AUC (Area Under the Curve) for the ROC (Receiver Operating Curve) was calculated.

Results

The HDF5 file generated from the SYNPUF inpatient visits set is 7.59 Mb (Megabytes), the post processed file is 9.40 Mb, and the file used in the predictive model fitting is 4.85 Mb. The complete analysis of the data can be found in the Github project: https://github.com/SBU-BMI/MappingOHDSI2HDF.

Table 1. Results of mapping 66,700 inpatient visits from the CMS SYNPUF dataset which was translated into the OMOP CDM to different groups in the HDF5 file.

CDM Table	HDF5 Group	Number of columns	Non-zero element counts
person	/ohdsi/person/	9	268,370
visit_occurrence	/ohdsi/visit_occurrence/	7	533,600
condition_occurrence	/ohdsi/condition_occurrence/	3,559	410,535
procedure_occurrence	/ohdsi/procedure_occurrence/	1,888	177,437
measurement	/ohdsi/measurement/count/	41	5,551
observation	/ohdsi/observation/count/	194	32,652

A random forest model with a population of 500 trees and two feature selection steps: remove zero variance features and select K best features (ANOVA F-score with 250 features) were utilized to predict 30-day inpatient readmission. On the testing set which was not utilized for training the the total (AUC) area under the curve was 0.53. The predictive ability of the model trained on the CMS's synthetic data to predict 30-day readmission is poor.

Conclusion

The approach described here shows that it is possible to run machine learning workflow against health care data of realistic size and complexity using data in the CDM as a source. It was not expected that the model would produce meaningful results due to the synthetic nature of the underlying data. By utilizing the CDM as the data model it removes some of the uncertainty in the data modeling process. Using the approach and tools described here it would be possible to train multiple models across different CDM datasets and compare the performance of 30-day predictive models and estimate the ability of the transference of the trained models across different datasets.

References

1. Miotto, R, Li, L., Kidd, BA, & Dudley, J. T. (2016). Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. Scientific Reports, 6, 26094.
2. Choi, E., Bahadori, MT, Schuetz, A, & Stewart, WF. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks.Retrieved from https://arxiv.org/pdf/1511.05942.pdf
3. Fernando Pérez, Brian E. Granger, IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007.