Reproducibility report formatting instructions for ML Reproducibility Challenge 2020

Shreyansh Anand
Queen's University
Kingston, Ontario, Canada
s.anand@queensu.ca

Alan Dimitriev
Queen's University
Kingston, Ontario, Canada
alan.dimitriev@queensu.ca

Yimaj Ishak
Queen's University
Kingston, Ontario, Canada
yimaj.ishak@queensu.ca

Reproducibility Summary

Scope of Reproducibility

The target paper 'An Explainable CNN Approach for Medical Codes Prediction from Clinical Text' proposes an architectural change to James Mullenbach's CAML by increasing the number of convolutions filters used in the model from 50 to 500. The target paper claims that their new model - SWAM-CAML - yields strong improvements over previous metrics on the ICD-9 code prediction task, specifically with Macro-averaged F1 score and model precision.

Methodology

To ascertain the validity of the target paper's claims we recreated the dataset and models used in the author's original experiments by following the details and explanations written. This involved gaining access to the MIMIC-III dataset, creating a curated dataset containing discharge summaries & related ICD-9 codes, and replicating all the models used in the experiment with the help of external libraries such as tensorflow and keras. Our experiments were solely based off of information provided in the target paper, as no associated repository was linked to the authors nor the paper itself.

Results

Our results did not validate the claims made in the original paper. Both the F1 Score and Precision did not increase for the novel SWAM-CAML from either Logistic Regression or the original CAML model.

What was easy

The tokenization process of the of the discharge summaries, what tokens were to be filtered out and how the tokenized discharge summaries should be truncated was easily inferred from the target paper. As well as reproducing the embeddings of the discharge summaries. Finally, gaining approved access to the MIMIC-III dataset was also an easy process.

What was difficult

The target paper fails to clearly layout the necessary preprocessing steps needed to amalgamate the final dataset used for the experiments from the raw MIMIC-III data files. As a result many assumptions had to be made to perform the experimentation. Furthermore, the authors provide no detail on how their qualitative evaluation methods were carried out. This made it impossible to perform the 'Secondary Evaluation' described in the target paper.

Communication with original authors

Multiple attempts across different communication platforms were tried when contacting the original authors over the time span of roughly three months, however no response was ever received.

Submitted to ML Reproducibility Challenge 2020. Do not distribute.

1 Introduction

In today's world data across many fields is exponentially increasing, including healthcare. As the amount of data increases there is a growing need to interact with and extract meaningful information from it. Doing this work manually by humans would be extremely tedious, so the integration of machine learning becomes a natural next step. Machine learning is a great tool for automating the creation and comparison of models and classification tasks. These tools will also help to liberate the time of healthcare professionals so they may attend to patients personally more frequently(1). During encounters with patients, clinicians often create unstructured short-form reports that are accompanied by a set of codes from the International Classification of Diseases (ICD). These codes are a standardized method of indicating diagnoses and procedures that were completed during the patient encounter. Widespread uptake of electronic health records (EHR's) that are composed of these short form reports on patient status have generated massive datasets that can be leveraged for useful analytic projects, especially using machine learning (3). However, manually translating unstructured reports into ICD codes is time-consuming and error prone. Automating this process is an important step to propel future research in this domain. The standardized nature of ICD codes present a plethora of opportunities for integration with artificial intelligence to support interesting future projects in the field of healthcare.

Although a useful application, automatically coding patients with ICD codes using machine learning presents some challenges. In the field of healthcare transparency in diagnosis is very important, and the explainable link between the clinical notes and ICD code is lost through most applications of predictive classification networks(2). The neural network must have explainable internal mechanics to be trusted in a clinical setting. To address these challenges an explainable CNN-based method for automatic ICD coding was developed named the Shallow and Wide Attention convolutional Mechanism (SWAM). The aim of this newly developed SWAM is to provide strong ICD prediction results, while also providing satisfactory explanations for its internal mechanisms.

2 Scope of reproducibility

To evaluate the performance of the SWAM model for automatic ICD coding, two metrics were used. These metrics will also be evaluated during this reproducibility study.

- Claim 1: The SWAM model will produce a greater Macro-averaged F1 score than the baseline bag-of-words logistic regression model and the baseline CAML model.
- Claim 2: The SWAM model will produce a greater Precision score than the baseline bag-of-words logistic regression model and the baseline CAML model.

While the paper initially seemed to contain a substantial amount of detail that would make replication a straightforward process, it became apparent there was a major hurdle for true replicability: a severe lack of information regarding data preprocessing before tokenization. The original paper goes into great detail about how tokens should be filtered (each one must appear in at least three distinct discharge summaries, must contain at least one alphabetic character, and each discharge summary must be truncated to only the first 2500 tokens - which is calculated based on the max length that will contain 90% of papers), but fails to explain or provide references for how those tokens are acquired. This only becomes clear once one is approved for access to the MIMIC-III dataset and observes its structure. Take the issue of linking the output labels (the ICD-9 codes) with the input vectors (word embeddings of individual discharge summaries). Within the MIMIC-III dataset the ICD-9 codes are stored in a .CSV file with an accompanying 'subject ID' to link them to a specific patient, however, no date is given to these ICD-9 codes. This is not a problem until it is recognized that each patient designated by a 'subject ID' can have multiple discharge summaries. With no way to link a specific discharge summary to the related ICD-9 codes, it becomes impossible to guarantee the correctness of our dataset. The authors state they found their data subset by 'following previous works' but do not include a citation to these works. The citations relating to MIMIC-III in the preceding segment of the paragraph do not offer any insight as to this methodology and in fact, one of the cited papers discusses MIMIC-II and covers 'CHARTEVENTS' which do not have any relevance to discharge summaries. The author's previous papers that utilized the MIMIC-III dataset were reviewed but contained the same lack of pertinent information regarding the process of isolating target data entries. As a result, our reproduction of the original paper's research is forced to diverge and work off of our own assumptions while trying to stay as faithful as possible to the author's original methodology.

Taking into account the limitations of our ability to replicate the subset of the MIMIC-III data used, our scope of what was truly reproducible had to change. While the original paper cannot be replicated faithfully, we still intended to observe if the reported improvement in model performance could be replicated. The original paper utilized two different approaches to evaluation: quantitative precision and F1 scores and a qualitative assessment analyzing differences in low-level feature importance. In all facets the authors of the target paper claim that they achieved an increased level of precision and better F1 scores than the baseline CAML model that their architecture is based on.

3 Methodology

For this reproduction we aimed to replicate the original paper as closely as possible in all facets of implementation.

3.1 Model descriptions

CAML

The Convolutional Attention for Multi-Label classification model is a previously developed CNN-based method for automatic ICD code assignment based on discharge summaries from intensive care units. It is a per-label attention mechanism for the purpose of learning distinct document representations of each model. This model is the basis of which the researchers in the original paper began to create their unique SWAM-CAML model. The CAML model consists of one convolution layer that takes in input (consists of 50 convolutional filters and a filter size of 10) an attention layer connected to the convolution layer, that then feeds its output to a dense node for each output class where a final sigmoid function is applied. The model as whole has a learning rate of 0.0001 and a dropout rate of 0.2.

SWAM-CAML

The basis of the model architecture is incorporating a Shallow and Wide Attention Convolutional Mechanism (SWAM) to the previously created CAML model. First the model transfers the base representation (i.e. clinical notes in the word-embedding form) to the convolution representation which represents the presence of informative snippets. The per-label attention mechanism selects the most relevant pieces of information. The created per-label vectors are used to compute attention over specific locations in the text, this will find the most relevant snippets of each code by attention layer. As a result SWAM is able to find informative snippets that are relevant to any ICD code through a series of convolutional layers and filters. The informative snippets that are non-generic and only relevant to specific codes as well are kept to help with creating distinction between codes. The SWAM network is shallow and wide by design to extract both non-generic and generic snippets that can be used to correlate with the ICD codes. The attention layer assigns weights to those snippets to correlate the relevant parts of the text to certain ICD codes. The final weighted score of all code snippets is used to give the final prediction of the presence of each code in the original inputted document.

The SWAM-CAML model consists of one convolution layer that takes in input (consists of 500 convolutional filters and a filter size of 4) an attention layer connected to the convolution layer, that then feeds its output to a dense node for each output class where a final sigmoid function is applied. The model as whole has a learning rate of 0.0001 and a dropout rate of 0.2.

SWAM-textCNN

SWAM-textCNN is a different implementation of the SWAM mechanism previously seen in SWAM-CAML. This variation differs solely in how the attention layer is is implemented when compared to SWAM-CAML, and remiains a convolutional neural network for text classification. The implementation difference is that textCNN contains a fully connected attention layer that weights inputs separately for each label.

The SWAM-textCNN model consists of one convolution layer that takes in input (consists of 500 convolutional filters and a filter size of 4) and an output layer that consists of a sigmoid output for each class.

3.2 Datasets

Dataset Description

This reproducibility paper utilizes the MIMIC-III dataset. This is an open-access dataset of 53,423 patients admitted to critical care units between 2001-2012. Information contained within the records includes vital signs, medications, length of stay and more. However, we are concerned with the observations and notes created by care providers, and procedural and diagnostic codes. The dataset can be accessed after approval from: https://mimic.mit.edu/. We applied for, and were given, access to the MIMIC-III dataset to perform our study. Due to the original paper not having any linked code repositories we are forced to build our replication entirely from scratch. This became an issue relatively quickly as the paper did not have a thorough explanation of how their discharge summaries and respective ICD-9 codes were retrieved from the MIMIC-III CSV files. In their section labelled 'dataset' it is mentioned that they follow previous work but do not cite the works they are referencing.

To this end we had to make decisions on how we were going to handle data extraction. The discharge summaries and ICD-9 codes are stored in separate files (with the ICD-9 codes themselves being stored in two different files based on if they are diagnosis or procedure codes) and there lacks a clear way to connect the two pieces of information. The main issue is that one individual (designated by a 'Subject ID' can have multiple different discharge summaries) making it a confusing matter to link the ICD-9 codes to their respective discharges. The solution we implemented was to only look at individuals who had one discharge event, meaning that all the ICD-9 codes associated with their Subject ID's would be to that discharge summary. Access to the MIMIC-3 database can be applied for at: https://physionet.org/content/mimiciii/1.4/

Training Splits

The dataset after padding, embedding and encoding will be split 60% for training, 15% for validation and 25% for testing. A random state will be set as 0 for the training and testing split to ensure the model's results are reproducible. The validation split is not included in this to truly test the robustness of the model by providing it different parts of the training data to learn from.

Preprocessing

Once we filtered for individual events, the pre-processing followed the details in the original paper. Following the paper we converted the string discharge summaries into individual tokens, filtering out: tokens that contained no alphabetic characters, tokens that appeared in less than three of the training documents, and truncated discharge summaries to the long tailed distribution (truncating when any list of tokens was longer than 90% of the other token lists). We then only kept discharge summaries that had been assigned an ICD-9 code that was within the top 50 most frequent ICD codes in the whole database.

Once the preprocessing was completed we were met with stark differences between the size of our dataset and the size reported in the original paper. Our final dataset contained 31,118 discharge summaries while the original paper's final dataset only contained 11,371 discharge summaries.

The final preprocessing step was to convert the tokenized discharge summaries into a vector representation to all for them to be used as model inputs. Following the paper we used a Word2Vec 'Continuous Bag of Words' embedding approach to vectorize our tokens. While the embedding size $d_e = 100$ was provided in the paper, the embedding window size parameter was not defined. Finally, once the discharge summaries were converted to an embedding representation, the embeddings were padded on both sides with zeros.

3.3 Hyperparameters

Hyperparameter values for this reproduction study were taken from the target paper in cases when they were provided, if specific hyperparmeter values were not explicitly defined in the target paper then original values were implemented.

3.4 Experimental setup and code

For our experiments we first conducted all the pre-processing steps as outlined above until we had the dataframe filled with padded/truncated documents. This was then flattened into a 2D matrix which was then split into training and testing data 0.75/0.25. Afterwards, the training data was inputted into a OneVsRest Logistic Regression Classifier and fit against the training labels. Once the training was complete, the testing feature set was predicted upon and the model was evaluated based on how well it could predict the testing labels based on the provided features. The original unflattened dataframe which was 3D was then split into training and testing data in the same

fashion. The training feature set was then inputted into the SWAM-CAML model which was trained to predict the training labels with 20% of the dataset being used for validation. The output of the model was a binary hot-encoded vector with 1 representing the ICD code for that column being a predicted label and 0 being not. Afterwards in a similar method as the Logisitc Regression experiment, the model was used to predict labels based on the testing feature set and compared with the actual label values. This was repeated two more times with the TextCNN model and the CAML model. Classification reports were built for each of the four models outlined above to provide more detailed insights. To evaluate the experiments we observed the F1 Macro average, F1 Micro average, area under the curve (AUC) and average Precision for each of the labels. For the entire code setup and for more details, visit https://github.com/Shred13/ReproducibilityStudyExplainableCNNs

3.5 Computational requirements

We used a CPU with 80 GB of RAM.. A GPU was not used. We were using a CPU with 24 cores. The processor model was a Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz with a CPU MHz of 2294.663.

Table 1: Total run time for each model.

Model	Time (minutes)
Logistic regression	39
CAML	39
SWAM-CAML	157
SWAM-textCNN	62

4 Results

Table 2: Comparison of the four models performance in ICD-9 code prediction.

Model	F1 Macro	F1 Micro	AUC	Precision
Logistic regression	0.28	0.40	-	0.47
CAML	0.19	0.19	0.500	0.11
SWAM-CAML	0.19	0.19	0.500	0.11
SWAM-textCNN	0.02	0.08	0.517	0.25

4.1 Results reproducing original paper

4.1.1 Claim 1: Macro-averaged F1 Score

The first claim the researchers made and we worked to reproduce in our report was to create a SWAM-CAML model that obtained a greater Macro-averaged F1 score than the baseline bag-of-words logistic regression and the original CAML model. We find that our reproduced SWAM-CAML model produces an F1 Macro score of 0.19, while the logistic regression produces an F1 macro score of 0.28, and the CAML model a score of 0.19. As such we are unable to successfully reproduce the results of the original paper, which does show an improvement of F1 macro score. The other related model, SWAM-textCNN, created also produces an F1 macro score lower than the logistic regression with a result of 0.02. This model performed better than the logistic regression in the original paper. We are unsuccessful in reproducing this secondary result as well.

4.1.2 Claim 2: Model Precision

The second claim we worked to reproduce is to produce a SWAM-CAML model that has a greater precision score than the baseline bag-of-words logistic regression and the original CAML model. We find that our reproduced SWAM-CAML model produces a precision score of 0.11, the bag-of-words logistic regression a score of 0.47, and CAML a score of 0.11. We are unable to successfully reproduce the original papers results of obtaining a precision score greater than the logistic regression model and the CAML model. The secondary model, SWAM-textCNN that was

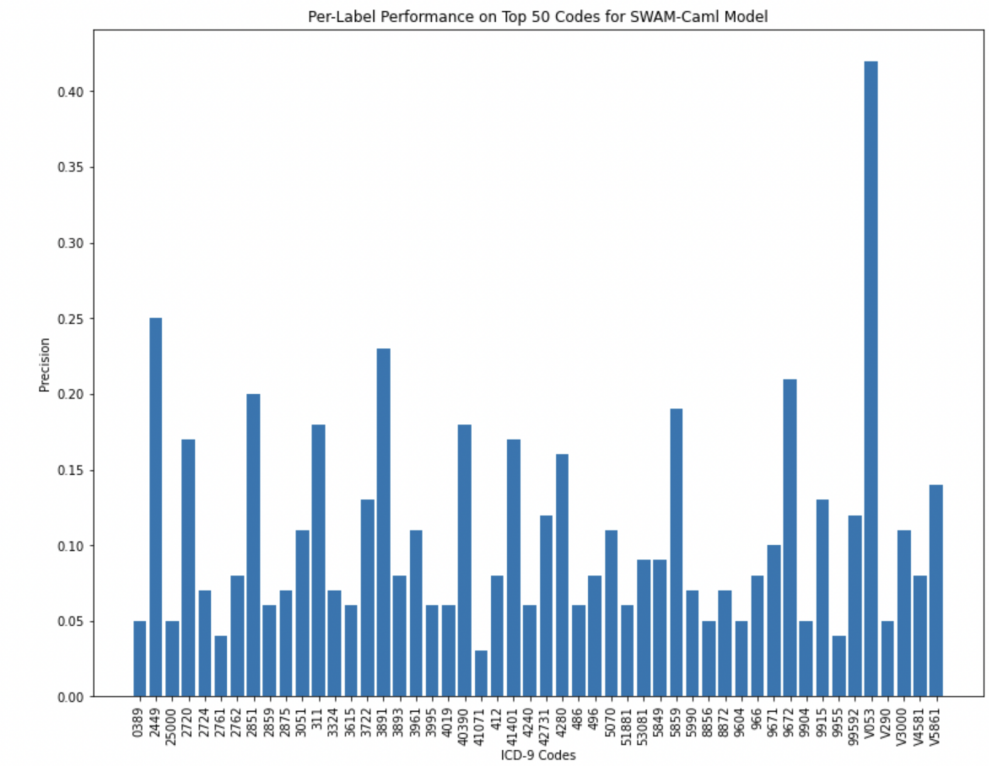

Figure 1: The per-label performances of SWAM-CAML

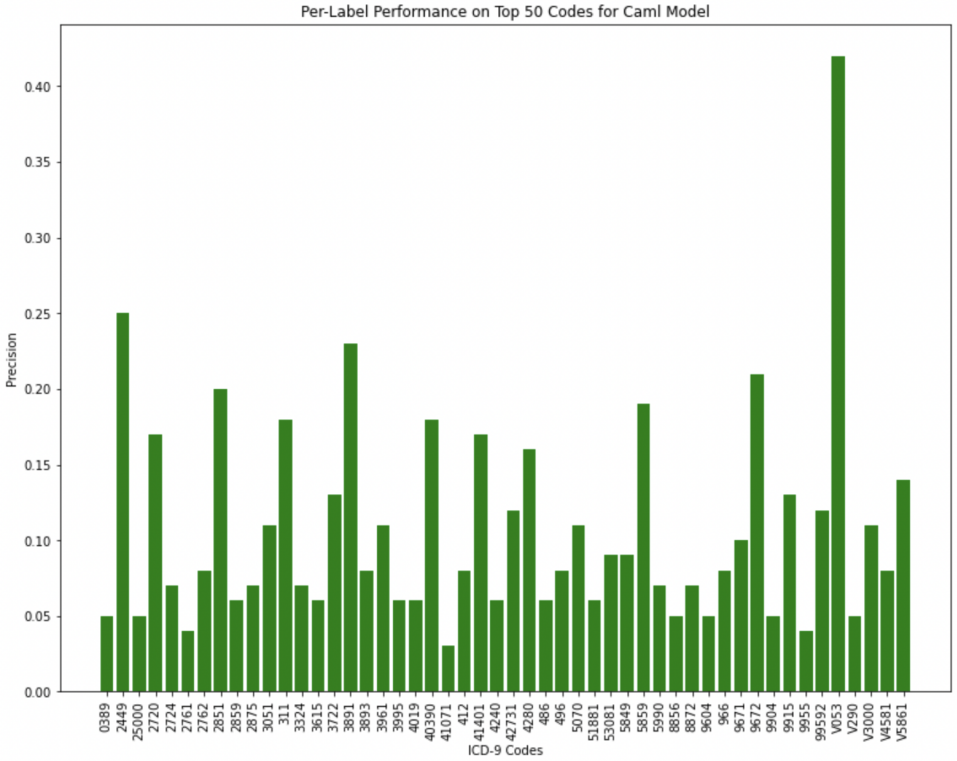

Figure 2: The per-label performances of CAML, the results were the exact same as with SWAM-CAML.

constructed produces a precision score of 0.25, which is lower than the logistic regression model but higher than either the SWAM-CAML or CAML model. This is counter to the original paper where the authors SWAM-textCNN model had greater precision than the logistic regression and CAML models but lower than SWAM-CAML. We are unsuccessful in reproducing the original papers secondary results.

4.1.3 Comparing CAML to SWAM-CAML

As can be seen in Figure 1 and Figure 2 both models produced identical precision performances on the input data. This visually highlights the lack of distinct performance increase claimed in the target paper between the CAML and SWAM-CAML architectures.

4.1.4 Secondary Evaluation

The paper described a secondary evaluation method where in they extracted informative snippets from both the SWAM-CAML and base CAML model and even displayed some of the snippets they extracted in a table. There is no mention of how this was done. The paper makes reference to the original CAML model as to where they retrieved their informative snippet extracting mechanism, and while the theoretical basis for the extraction is provided, the CAML model's associated GitHub (hosted by the paper's author) does little to shed light onto how it is actually implemented.

5 Discussion

5.1 What was easy

The sections of the target paper that were easy to reproduce were coincidentally the ones that had detailed explanations as to their implementation. Once a dataset of discharge summaries and their associated ICD-9 codes was extracted from the MIMIC-III .csv files it was rather straightforward to implement some of the preprocessing steps. The tokenization process of the of the discharge summaries for example was easy to reproduce as the target paper thoroughly described what takens were to be filtered out and how the tokenized discharge summaries should be truncated.

Reproducing the embeddings of the discharge summaries (although there was no way to properly check with the target papers results) seemed easy as the details (besides one missing parameter) were well stated.

Gaining approved access to the MIMIC-III dataset was also an easy process, although it was likely only easy in the context that we are researchers at an accredited university requesting access specifically for a paper reproduction study. The only 'Papers With Code' repository associated with the target paper was a reproduction performed by someone who could not get approved for MIMIC-III data access.

5.2 What was difficult

The target paper of this reproduction study contained many different issues that made a proper reproduction of its stated results and methods near impossible. The largest issue faced was the severe lack of clear preprocessing steps that the authors took to filter, and create, their final dataset of discharge summaries and ICD-9 codes from the base MIMIC-III dataset. Attempting to reproduce the results of a study when the initial step in the research (data preprocessing) is not thoroughly explained makes the task significantly more challenging. We had to work from our own assumptions as to what the best way to progress through extracting the required data from the MIMIC-III data files. We associate our independent extraction methods of data as being one of the reasons as to why we were unable to produce results similar to the original paper.

Our issues did not stop there however, due to the discrepancy in the size of our final dataset and the papers we ran into severe memory issues when dealing with the padded embedding tensors of the discharge summaries. Memory constraints, and how to deal with them, required a larger time investment that initially planned and was a large bottleneck to our progress in this study.

Reproducing the 'Secondary Evaluation' described in the target paper was not only difficult, but impossible from the target paper's own explanation. No explanation was provided in the target paper and while some theoretical basis for how the snippet extraction process works can be found in a separate paper by a different author that they cite, it is not

enough to instruct proper implementation. The process of extracting informative snippets from the models requires a deep understanding of attention mechanisms and interpretation techniques that cannot be infered from reading the target paper alone.

The final notable difficulty was more of an overall issue with the target paper. Many sections had awkward wording and the paper seemed to contradict itself in certain sections. An example being that the original paper devoted multiple paragraphs to explain how SWAM-CAML is a wider version of a the CAML architecture, but then at certain points in the paper it refers to a 'Narrow SWAM-CAML' model, which by itself is never defined but through broad assumptions can be assumed to be the normal CAML model.

While not a 'difficulty' per say, there is an outstanding issue with the target paper that we feel needs to be addressed. In the target paper the authors make note that their SWAM-CAML model performs better than the model that it is based on (James Mullenbach's CAML model), achieving this increase in performance by increasing the number of convolutional filters from 50 to 500. However, the range of the number of filters that was included during the parameter search for optimal hyperparameters of the original CAML model ranges up to 500, indicating that if a wider model does result in improved performance then how could these two papers possibly differ as to the ideal number of filters?

5.3 Communication with original authors

Communication with the original authors was attempted via multiple emails to the email address listed on the original paper's arXiv.org submission. We cross referenced the email with the author's official page on the Southwest Jiaotong University website to ensure it was still current, yet our attempts to reach out were unsuccessful and we did not receive any responses. Shuyuan Hu did not appear in any online searches, nor was their email listed anywhere and as such we could not contact them. After our emails were not responded to we attempted multiple times to call the number listed on Fei Teng's university page but could not get through.

References

[1] BHARDWAJ, R., NAMBIAR, A. R., AND DUTTA, D. A study of machine learning in healthcare. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)* (2017), vol. 2, IEEE, pp. 236–241.

[2] MIOTTO, R., WANG, F., WANG, S., JIANG, X., AND DUDLEY, J. T. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics 19*, 6 (2018), 1236–1246.

[3] MURDOCH, T. B., AND DETSKY, A. S. The inevitable application of big data to health care. *Jama 309*, 13 (2013), 1351–1352.