Lab Notebook: Collaborative Filtering

My Health Prognosticator

Creator: Jason Ko

**Introduction**

Heavily inspired by the research paper “Time to CARE: a collaborative engine for practical disease prediction”, my implementation focuses on the collaborative filtering algorithm for disease prediction. Collaborative Filtering is a recommender system designed to predict the preferences of a person based on the preferences of other similar users. Contrary to belief, this method is actually well suited to disease prediction because of the known collaborative nature of diseases.

**Methods**

There are three main methods in the CARE framework. The first, **CARE**, is the standard implementation of Collaborative Filtering. The second, **ICARE**, is an iterative version of CARE that groups patients into separate disease groups, based on the diseases they have already developed. Then, Collaborative Filtering is applied to each disease group. The third, **time-sensitive ICARE**, is ICARE with the addition of a best match subset algorithm which makes it possible to exploit the temporal ordering in which diseases occur. This improvement makes it applicable to long-term, diverse data that is available to Optum. From my experiments, I’ve noticed that ICARE is far superior to the standard CARE implementation, so users should use either ICARE or time-sensitive ICARE when making predictions.

**Filtering**

There is a filtering process before Collaborative Filtering. We obtain a “training set” of patients by constricting the database with users that have at least 2 common diseases with our target patient. This serves to remove the influence of patients with little or no similarity with the target patient. This does not result in loss of information, and serves to reduce the runtime of our algorithm, and improve efficiency. **It is important to note that both ICARE and time-sensitive ICARE do not require this filtering process, because we apply Collaborative Filtering to each individual disease group**.

**Experiments**

Our dataset consists of users with either:

* psoriasis (icd9 code 5715) or
* cirrhosis (icd9 code 6961)

and the accompanying diseases that follow with the aforementioned two diseases. After we have created our patient database, we will artificially create two patients with heavy symptoms of each disease. Our first patient, whom we’ll call “patient zero”, will have symptoms of cirrhosis. Our second patient, named “patient one”, will have symptoms of psoriasis. Note that I have done research ahead of time, and each symptom I have chosen will be mutually exclusive of the other set of symptoms. What that means is that none of the disease symptoms in psoriasis are evident in cirrhosis, and vice versa. The two patients have the following information:

Patient Zero

* Age 57
* Male
* Diseases:
  + Chronic Fatigue Syndrome (78071)
  + Loss of Weight (78321)
  + Jaundice (Yellow skin/eyes) (7824)

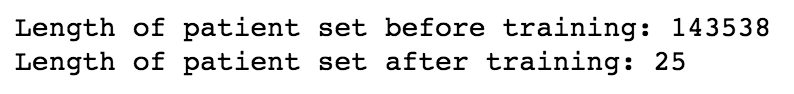
Patient One

* Age 60
* Female
* Diseases:
  + Other symptoms involving skin and integumentary tissues (7829)
  + Congenital anomalies in nails (7575)
  + Stiffness of joint involving lower leg (71956)

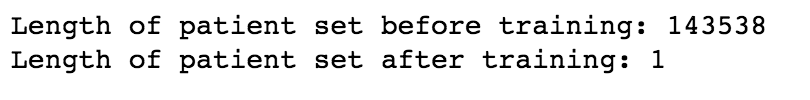
**CARE Prediction**

First, we need to filter the data for our training set. The following is a result of the data after being trained on our target patients:

Patient Zero



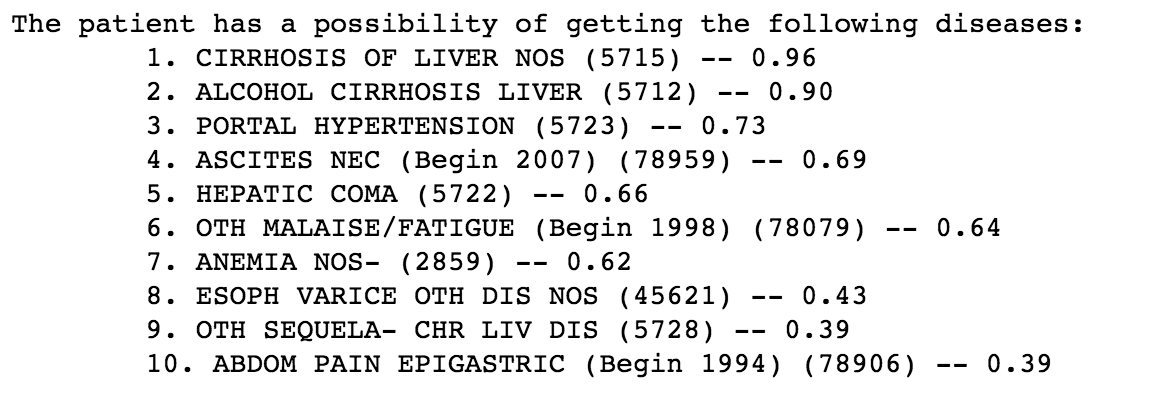
Patient One



You can notice that for patient one, the training set only returns one result. This is due to the heavily unbalanced dataset we’ve been given. Ideally, the dataset used for training will be much more rounded, and result in a larger training set. This also provides further proof for the superiority of the ICARE implementation, which is not hindered by sparse datasets.

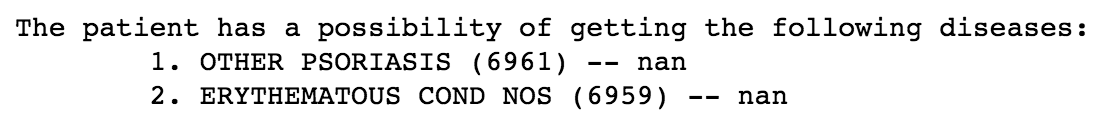
After running the data, we generate the following predictions for the standard CARE implementation:

Patient Zero (symptoms of cirrhosis)



*Here you can see that the weighted scores are a bit high, with a .96 out of 1 for cirrhosis. This is again due to the unbalanced dataset that we’ve been given.*

Patient One (symptoms of psoriasis)



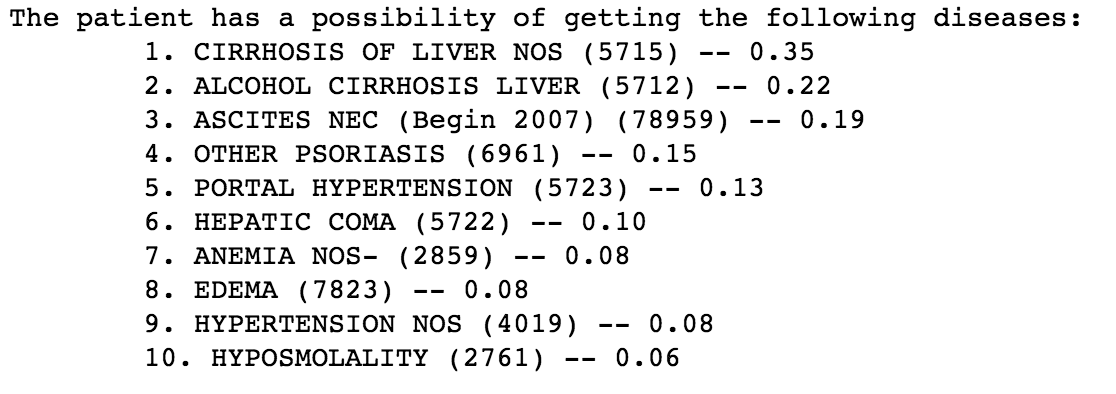
*You can see that each predicted disease has a weighted score of NaN. This is due to the fact that there has to be more than 1 patient in the training set in order to calculate a weighted score (due to the way the algorithm works). This is another drawback of using standard CARE implementation.*

*Overall, the disease we were trying to predict showed up as the highest rank in both of our results, which shows that the CARE system is successful in predicting one of the targeted diseases.*

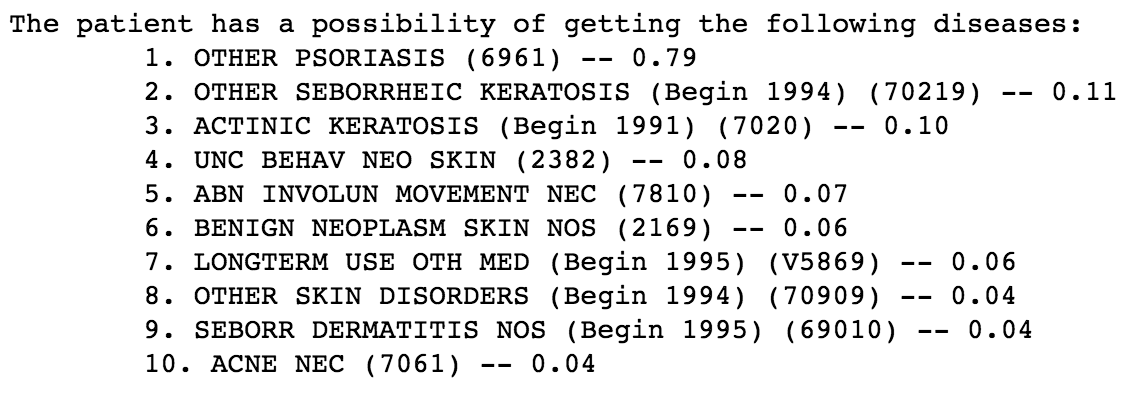
**ICARE Prediction**

Since no filtering is required, we can simply run the predictions with our full dataset. The predictions are generated with the ICARE implementation:

Patient Zero (symptoms of cirrhosis)



Patient One (symptoms of psoriasis)



*You can see that the weighted scores are more realistic, and ICARE was able to predict the correct disease for both patients.*

**Time-Sensitive ICARE**

Unfortunately, we were not able to obtain patient data from 5 - 20 years prior, as the data we were given only showed patient history from one period of time. However, once we obtain this data, we can make predictions based on the temporal pattern in which diseases occur.

**Conclusion**

Based on these results, we can see that collaborative filtering is effective in predicting targeted diseases. After running and analyzing both CARE and ICARE, I have also come to the conclusion that ICARE is superior in every way. Therefore, when making predictions on patients, either ICARE or time-sensitive ICARE should be the method to use. When looking at the big picture, Collaborative Filtering has many advantages to standard machine learning algorithms such as neural networks, and other classification/regression algorithms.

First, the process is very fast. Reading in a csv file of ~1 million rows takes roughly ~15 seconds, and the training/predicting phase is nearly instantaneous. And once we can save the data into an actual database, reading and accessing the data will be instantaneous as well. This is a huge advantage compared to a neural network, which may take hours to train a dataset this large.

Second, there were no third party libraries necessary to implement CARE. Typically, one would use a library such as python’s scikit-learn or keras to implement various machine learning algorithms. With CARE, the algorithms within Collaborative Filtering are simple, yet extremely powerful. This allows our CARE framework to be portable and easily implemented in other languages on other platforms.

Lastly, the CARE framework can take into account the temporal aspect in which diseases occur. This cannot be done with traditional machine learning algorithms, and is unique to the structure that is built with CARE. With Optum’s large, diverse dataset, we can differentiate between chronic disease and lone occurrence. With our application, doctors will be able to suggest previously unconsidered concerns, and facilitate discussion about early testing and intervention.