PhUSE EU Connect 2018

Paper ML06

Machine Learning with SAS

Cathal Gallagher, d-Wise, Manchester, UK

Jiangtang Hu, d-Wise, Raleigh, USA

Abstract

Artificial Intelligence (AI) is intelligence learned by machines. Machine Learning is a technique within AI to give computers the ability to learn without explicit programming and handcrafted rules. Deep Learning is a special technique of machine learning which utilizes neutral networks providing the most powerful machine learning technique in real life applications.

Machine learning, especially deep learning, is very successful in fields like computer vision. In the broader medical field, a very first successful application is within medical image diagnosing, which is very close to general image classification in computer vision. But on the clinical research side, the power of machine learning has not yet been realized.

Within clinical research, we handcraft lots of rules for various tasks. Considering the large number of manual rules and associated code we develop, it's the perfect situation to think about the possibility of using machine learning techniques for the tasks without explicit programming.

Machine learning has been a popular field with languages such as Python and R. However you can use many machine learning algorithms in SAS as well. Some of the more popular ones are listed below.

- Naïve Bayes Classifier Algorithm (Supervised Learning - Classification)
- K Means Clustering Algorithm (Unsupervised Learning - Clustering)
- Support Vector Machine Algorithm (Supervised Learning - Classification)
- Linear Regression (Supervised Learning/Regression)
- Logistic Regression (Supervised learning – Classification)
- Artificial Neural Networks (Reinforcement Learning)
- Decision Trees (Supervised Learning – Classification/Regression)
- Random Forests (Supervised Learning – Classification/Regression)
- Nearest Neighbours (Supervised Learning)

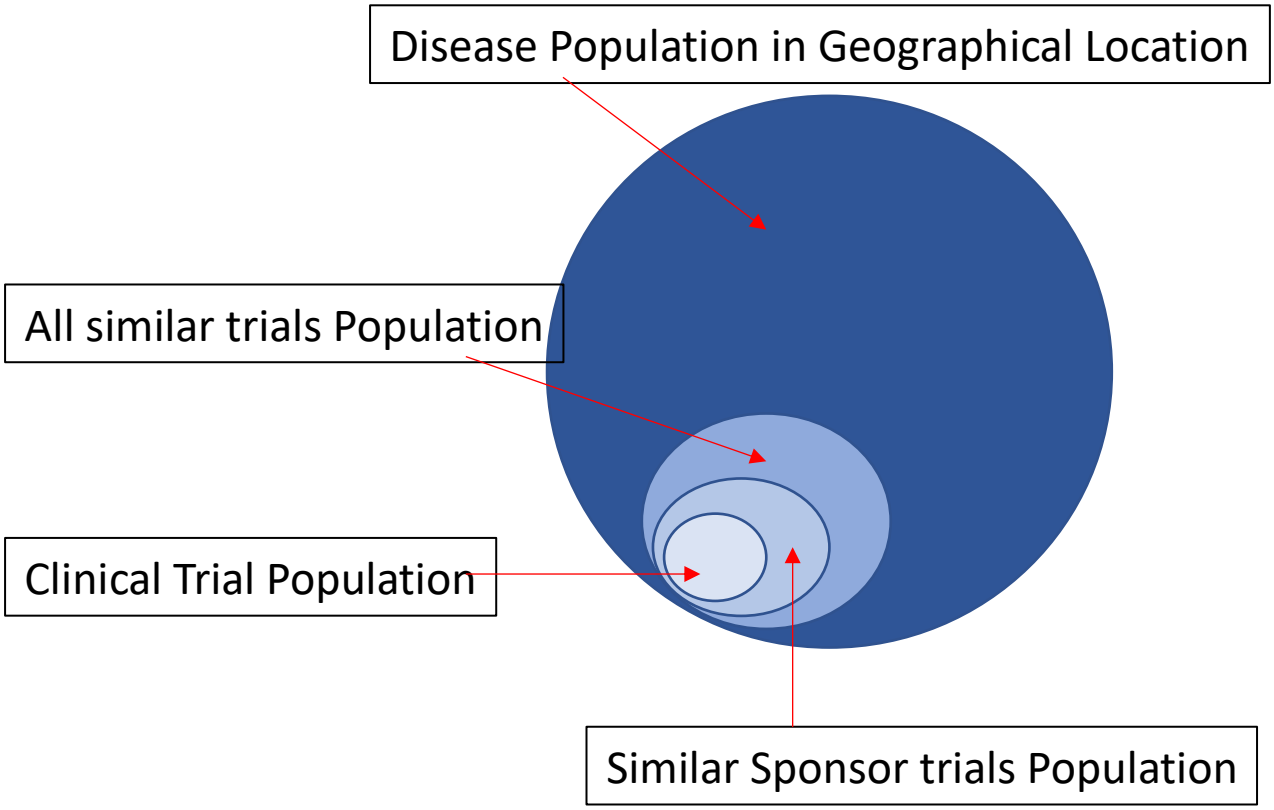
There are several areas where machine learning could be applied in clinical data. Mapping of legacy data to CDISC standards, fraud detection, candidate selection and information extraction from free text records are just some examples.

This paper will concentrate on a specific use case of generating extra population data for the use of quantitative risk assessment of anonymised data.

Back ground to using extended populations in clinical trials.

When calculating the risk of reidentification on a clinical trial, you traditionally divide the number of patients on a trial with similar characteristics into 1. For example, if you have 100 people on a clinical trial, all are the same race, all live in the same country, all are the same age etc. The only

difference is that 90 patients are male, and 10 patients are female. In this case to find the maximum risk of reidentification you divide 1 by 10 to get a reidentification risk on 0.1. Clearly when you consider many identifiers such as age, race, gender, country, height, weight, BMI etc then even large trials can have very identifiable patients. Therefore, we either anonymise some identifiers or we can calculate the risk based on a larger population. The diagram below gives examples of larger and larger populations.



The diagram above illustrates the clinical trial population that we want to share but at the same time protect the privacy of the patients. The next largest circle is the similar sponsor trials population, this data may be readily available and can help when trying to calculate the risk of reidentification. Data from the next two circles, all similar trials and disease population in a geographical location are much more difficult to get as they do not belong to the sponsor. This is where ML algorithms would be useful.

Getting hold of real population data can be time consuming and expensive. Using ML algorithms, we can rapidly generate larger populations that are reflective of participants on clinical trials. This data can then be used to reduce the probability of a successful attack on the anonymised data, and in turn, help to protect the privacy of patients.

The typical argument against using ML algorithms to generate this new population data is that the data generated is not real data, and therefore cannot accurately reflect the true aspects of either, all similar trials population or the disease population in a geographical Location.

The code below chose a random sample from some cdisc data. This was used as the training data for the K-nearest neighbour algorithm.

```
proc surveyselect data=cdisc.dm
   method=srs n=1000 out=dm_train;
run;
```

The following code then drops race from the original CDISC.DM dataset

```
data dm_in;
set cdisc.dm;
drop race;
run;
```

Below we run the K-nearest neighbour algorithm. This is looking at sex, height, weight, BMI and age, and attempting to allocate a race to race patient.

```
proc discrim data = dm_train
                test=dm_in
                testout=dm_out
            method=npar k=10
            testlist;

     class race;
     var sexn height weight bmi age;
run;
proc sort data=dm_out (rename=(_INTO_=RACE));
by usubjid;
run;

proc sort data=cdisc.dm;
by usubjid;
run;
```

Finally the compare runs, which shows us how many differences we get from our generated data and our real data.

```
proc compare base=cdisc.dm compare=dm_out;
run;
```

Since we choose a random sample for our training data, our results can change for each run. However, after 10 runs we have always had results of greater that 86% match between the generated data and the original data.

Conclusion

SAS contains a wide array of Machine Learning algorithms just like Python and R. These algorithms can be leveraged to help with things such as legacy data mapping, fake data generation, fraud detection as risk of reidentification when dealing with attacks on anonymous data.